BUILDING A CHATBOT FOR HEALTHCARE USING NLP

Submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

BRUCE KEVIN (Reg.No - 39110191)

BRIKESH VIKIN (Reg.No – 39110190)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAR NAGAR, RAJIV GANDHISALAI,
CHENNAI – 600119

APRIL – 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | 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 **Bruce Kevin(39110191) and Brikesh Vikin(39110190)** who carried out Project Phase-2 entitled **"BUILDING A CHATBOT FOR HEALTHCARE USING NLP"** under my supervision from Jan 2023 to April 2023.

Manju C.

Internal Guide
Mrs. Manju C Nair, M.E.,(Ph.D)

[illegible]

Head of the Department
Dr. L. Lakshmanan, M.E., Ph.D.

Submitted for Viva-voce Examination held on ____________________

[illegible]

Internal Examiner

[illegible]

External Examiner

DECLARATION

I, **Bruce Kevin(Reg.No- 39110191),** hereby declare that the Project Phase-2Report entitled "**BUILDING A CHATBOT FOR HEALTHCARE USING NLP"** done by me under the guidance of **Mrs. Manju C Nair, M.E.,(Ph.D)** is submittedin partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

Kevin

DATE:
PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and successfully completing it. 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**., Head of the Department of Computer Science and Engineering for providing me with the necessary support and details at the right timeduring the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. Manju C Nair, M.E.**,(Ph.D) for her valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my phase-1 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 waysfor the completion of the project.

ABSTRACT

A chatbot or conversational agent is a software that can communicate with a person using natural language. Designing dialogue is one of the key challenges in artificial intelligence and natural language processing. Creating effective chatbots hasbeen the most difficult problem for artificial intelligence since its beginning. While chatbots can perform a variety of tasks, their primary objective is to comprehend human language and provide appropriate responses. In the past, chatbotarchitectures were created using simple statistical methods or manually crafted templates and rules. However, since around 2015, end-to-end neural networks havereplaced these models due to their greater learning capabilities. The encoder-decoderrecurrent model, which was originally developed for the neural machine translation field and proved highly successful, now dominates conversation modeling. Several enhancements and modifications have been made to chatbots to significantly enhance their conversational abilities up to this point.

In this essay, we conducted a thorough review of recent literature. We lookedat a wide range of papers that discuss chatbots. The bot will identify the user's ailmentand deliver doctor information. It offers dietary recommendations that indicate the kindof food you should eat. People will therefore be aware of their health and be properlyprotected.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	Full Form
NLP	Natural Language Processing
AI	Artificial Intelligence
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Networks
NLG	Natural Language Generation
NLTK	Natural Language Toolkit
EHR	Electronic Health Records
FAQ	Frequently Asked Questions
EDA	Exploratory Data Analysis
TF	Term Frequency
IDF	Inverse Document Frequency
HIPAA	Health Insurance Portability and Accountability Act
HTML	Hypertext Markup Language

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Excellent healthcare is essential to modern life. These days, individuals are occupied with work receptions, work at work, and further Internet addictions. In termsof your health, you are passive. Due to a minor issue, you decide not to visit the hospital. This might have serious negative effects. As a result, think about creating anAI-powered health chatbot system that can recognize the condition and offer a rudimentary description of the ailment before you contact your doctor. Using text or text-to-speech technology, chatbots are software applications that communicate withconsumers. When chatbots were first created, they were simply designed to conversewith people for amusement.

The largest problem that India is currently facing is how to both deliver high- quality, reasonably priced healthcare to its expanding population and remain economically viable. Patients postpone treatment or look for nearby but still cost- effective medical facilities to suit their medical demands since it is difficult to obtain and transport medical facilities, especially in rural regions. It implies that. The functionof medical chatbots is to link patients to chatbots and deliver the proper antibiotics/drugs and precautions, which is an effective approach to give patients fastaccess to high-quality care, treatment, and care. Helpful for :

A. Necessity

Emergencies are nothing new to the healthcare sector. And time is a very important factor in dealing with them! Healthcare chatbots rapidly give useful information, especially when every second counts. A doctor may rapidly obtain the patient's information, such as past records, other ailments, allergies, check-ups, etc.,via a bot, for instance, if a patient runs in with an attack.

B. Objectives

A chatbot is a software application that converses with users either orally or textually. By communicating with consumers in a human-like manner, a medical chatbot streamlines a healthcare provider's duties and aids in improving performance.

Intelligent medical chatbots might be very useful in a variety of situations fordoctors, nurses, therapists, patients, and their families. They can intervene and cut down on how much time is spent on things like:

1. Providing consumers with health-related information
2. Guidance for patient
3. Dose and control of medications
4. Establishing links between individuals and groups and first responders
5. Questions of this nature (contact information, location, hours of operation,and service/treatment specifics)

It's crucial to remember that while chatbots can provide helpful data and symptoms, they are not trained to provide a diagnosis. The fundamental idea behindthese intelligent chatting or messaging algorithms is to establish communication before requiring human intervention.

The four basic types of chatbot applications are advisory, entertainment, commercial, and service bots. Advisory chatbots are designed to fix items, provide maintenance, provide advice on services, or make ideas. These kinds of chatbots have the ability to connect with individuals when necessary, as well as provide guidance and help. Chatbots for entertainment are designed to keep users interestedin their favorite movies, musicians, sports, and other events. These chatbots provideinformation about upcoming events, ticket discounts, and betting possibilities. Commercial chatbots' primary purpose is to make consumer purchases easier. A pizza delivery service, for example, may utilize a message interface to announce deals or collect delivery orders. Customer service chatbots are focused on offering amenities. For example, a logistics company can utilize instant messaging to react todelivery questions and send copies of delivery documentation in place of phone callsor emails.

CHAPTER 2
LITERATURE SURVEY

2.1 INFERENCES FROM LITERATURE SURVEY

This study is a comprehensive review of the effectiveness of synchronous chat-based online one-on-one psychological state therapies, which involve text-based exchanges between patients and therapists. The authors of the study highlight the growing popularity of this form of therapy as a web-based intervention for individualsexperiencing psychological issues. The review finds preliminary evidence to support the use of text-based synchronous communication treatments as an effective meansof psychological intervention and support. The study indicates that text-based synchronous communication treatments had comparable results to traditional treatments and outperformed waitlist circumstances overall. This suggests that chat-based therapy can be a valuable alternative to traditional forms of therapy for individuals who prefer the convenience and accessibility of online support.

However, the authors caution that future research studies must consider the feasibility of using this technology in clinical settings, taking into account its cost andother practical considerations. They also suggest that the effectiveness of this form oftherapy may depend on various factors, such as the therapist's training and experience, the type of psychological issue being treated, and the level of support provided to patients outside of therapy sessions. Overall, the study offers valuable insights into chat-based therapy's potential benefits and limitations as a viable form of psychological intervention and support. It highlights the need for further research to fully understand the effectiveness of this technology and its potential to improve mental health outcomes for individuals.

The chatbot is an innovative tool designed to function as a virtual doctor, providing patients with a way to interact with a healthcare professional. The chatbot utilizes advanced technology, including linguistic communication processing and pattern-matching algorithms, to provide patients with accurate and helpful responses.

The chatbot was developed using the Python programming language and hasundergone an evaluation to assess its accuracy and effectiveness. The results of theevaluation indicate that 80% of the responses given by the chatbot were correct, whilethe remaining 20% were unclear or incorrect. This suggests that the chatbot can be useful for training purposes and can also provide valuable assistance in providing firstaid and increasing awareness as a virtual doctor.

In addition, the chatbot could be utilized to assist in initial diagnoses or to provide guidance for non-urgent medical concerns. This has the potential to improve access to healthcare services for individuals who may not have easy access to in- person consultations or who may not feel comfortable seeking medical advice in person. Overall, the chatbot represents an exciting development in the field ofhealthcare technology, providing patients with a valuable tool for accessing healthcare services and receiving medical advice.

This text presents an examination of the effectiveness and practicality of synchronous chat-based text communication in online one-on-one psychological therapies. These therapies have become increasingly popular as web-based interventions for psychological issues. The analysis reviews the different technologiesused for synchronous chat-based therapyand assesses their effectiveness. However,the study identifies several limitations in the current systems, including delayed responses to patients, which can be disadvantageous. Patients may have to wait fora long time before receiving a response from a consultant. Furthermore, some of the techniques may require a fee for telecom or chat communication. Therefore, future research studies should consider the feasibility of these technologies for use in clinicalsettings.

2.2 OPEN PROBLEMS IN THE EXISTING SYSTEM

The statement is referring to current healthcare systems that use live text chatas a form of communication between patients and healthcare experts. While these systems do provide a convenient way for patients to reach out to healthcare professionals, there are some drawbacks to their use. One significant drawback is that patients may not receive a fast answer. This is because live text chats rely on theavailability of healthcare professionals to respond to patient inquiries. If there are toomany patients and not enough healthcare professionals available to respond, patientsmay have to wait a long time for an expert to notice them.

Additionally, some procedures may levy fees for telephone or live chat communications. This can add an additional financial burden to patients who may already be struggling with healthcare costs. As healthcare technology advances, future research investigations will need to take into account whether these technologies are economical in clinical settings. For example, if the costs associatedwith implementing and maintaining a live text chat system are too high, it may not bea viable option for many healthcare providers. Overall, while live text chats can be a helpful tool for patients to communicate with healthcare professionals, there are significant drawbacks that must be considered. Future research will need to explore the most effective and economical ways to implement these technologies in clinical settings.

The statement refers to the use of artificial intelligence (AI) techniques to classify emotions in human interactions, particularly in counseling scenarios. The research is centered around building models that can categorize emotions using a combination of recurrent neural networks (RNNs), deep learning algorithms, and convolutional neural networks (CNNs). To build these models, a large amount of labeled data is used, which allows the algorithms to learn and improve over time.

In addition to the use of AI techniques for emotion classification, the researchers are also using natural language processing (NLP) and natural language generation (NLG) to understand user interactions in counseling sessions. This involves analyzing the language used by the user, including the words they choose and the tone they use, to gain insight into their emotional state. To accomplish this, amulti-modal approach to emotion recognition is employed. This means that multiple sources of information, such as facial expressions, body language, and speech patterns, are used to classify emotions. By using multiple sources of information, theresearchers hope to improve the accuracy of their emotion recognition models.

To better understand the semantics of words and phrases, the researchers have gathered large corpora of text data. They then convert this text data into word vectors, which are numerical representations of the meaning of words. This allows the algorithms to understand the meanings of words and phrases, even when they are used in different contexts. Additionally, the researchers are utilizing lexical knowledge of synonyms to help the algorithms better understand the nuances of language.

Overall, the goal of this research is to develop more effective and accurate ways to classify emotions in human interactions, particularly in counseling scenarios. By using a combination of AI techniques, including RNNs, deep learning, and CNNs,as well as NLP and NLG, the researchers hope to better understand the emotional states of individuals and provide more effective counseling services.

Drawbacks:

1. An intricate user interface. making it harder for the user to understand the procedures
2. Due to a large amount of spilled data, the user received a delayed response.
3. Large data processing also took longer.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/ RISK ANALYSIS OF THE PROJECT

The statement describes two significant developments in the chatbot industry.The first development is the rapid growth of messaging services over the past severalyears. Messaging services have become a popular platform for ordering, paying, andbooking services, which traditionally require a distinct website or application. This eliminates the need for users to download multiple apps for different services and allows them to perform various activities directly from their chosen chat app. Popularmessaging apps such as Facebook Messenger, WhatsApp, Line, and WeChatprovide these services.

The second development is the improvement in the quality of decision-makingand comprehension by chatbots, thanks to the advances in deep learning and sophisticated AI approaches. These techniques enable chatbots to manage and analyze massive amounts of data to deliver results that surpass human performance.As a result, chatbots have become more capable of handling complex tasks and interacting with humans in a more natural way.

The article also defines four categories of chatbot applications: response- generated-based, service-based, knowledge-based, and goal-based. Response- generated-based chatbots are programmed to respond to specific keywords orphrases with pre-defined responses. Service-based chatbots provide a range of services, such as ordering food or booking a flight. Knowledge-based chatbots provide information on a specific topic, such as weather or news. Finally, goal-basedchatbots help users achieve a specific goal, such as losing weight or saving money.

In summary, the recent worry about chatbots is related to the growth of messaging services and the improved capabilities of chatbots thanks to deep learning and AI approaches. The article also defines four categories of chatbot applicationsbased on their functionality.

RESPONSE GENERATED-BASED CHATBOT

The input and output of the response models are natural language text, whichmeans that the chatbot can understand and generate responses in a language that issimilar to human language. The dialogue manager is responsible for selecting whichresponse models to use to generate the chatbot's response. The chatbot can use different response models, such as pre-defined responses, machine learning models,or a combination of both. The dialogue manager chooses the response models basedon the context of the conversation and the user's input.

The answer is produced by the dialogue manager in three phases. In the first phase, the conversation manager generates a collection of replies by using all available response models. In the second phase, the response is sent according to priority. For example, if there is a predefined response that matches the user's input,the chatbot will use that response. If there is no predefined response, the chatbot willuse a machine learning model to generate a response. In the third phase, the modelselection strategy is used to choose the answer when there isn't a priority response. This could involve selecting the most relevant response based on the context of the conversation, the user's input, or the chatbot's previous responses.

SERVICE-BASED CHATBOT

Service-based chatbots are designed to provide a range of services to users, such as ordering food, booking flights, or scheduling appointments. These chatbots can be used for professional or domestic purposes. Professional service-based chatbots are designed to assist businesses and organizations in providing their services to customers. For example, a logistics company may use a chatbot to transmit copies of delivery paperwork to customers, which saves time and reduces errors. Similarly, a bank might use a chatbot to help customers with account-related queries, such as checking account balances or transferring funds.

Domestic service-based chatbots, on the other hand, are designed to assist individuals in performing daily activities, such as ordering food or booking a ride. Forexample, a restaurant might offer a chatbot to help customers order food, pay for theirmeals, and track the delivery status of their order. A ride-hailing app might use a chatbot to help users book a ride, get an estimated fare, and track the driver's location.

KNOWLEDGE-BASED CHATBOT

Knowledge-based chatbots rely on the knowledge that they have learned froma training dataset or other primary data sources. There are two primary data sourcesfor knowledge-based chatbots: closed-domain and open-domain. Closed-domain data sources deal with a specific area of knowledge, and the chatbot can provide responses related to that area. For example, a medical chatbot might use a closed- domain dataset that includes instances from bAbI, MCTest, and Daily Mail, which provides all the necessary data to respond to medical queries.

On the other hand, open-domain data sources deal with broad subjects, and the chatbot can provide responses based on the queries asked. For example, a general knowledge chatbot might use an open-domain dataset like Allen AI Science or Quiz Bowl, which can provide responses to a wide range of queries related to science and general knowledge.

GOAL-BASED CHATBOT

Goal-based chatbots are designed to achieve a specific task and assistconsumers by providing them with important information during brief discussions. Themain characteristic of goal-based chatbots is their ability to achieve specific goals. These chatbots are designed to understand the user's intent and provide responses that help the user achieve their desired outcome. For example, a goal-based chatbotmight help a user book a flight or reserve a hotel room.

Goal-based chatbots are particularly useful for companies that want to automate their customer service operations. Customers can interact with a company'sinternet chatbot to find solutions to their issues or answers to their inquiries. For example, a customer might interact with a chatbot to inquire about a product's availability, pricing, or shipping information. The chatbot would provide the customerwith the necessary information and help them achieve their goal.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

HARDWARE CONFIGURATION

- Processor- i3/Intel Processor
- RAM- 4GB (min)
- Hard Disk- 128 GB

SOFTWARE CONFIGURATION

- Operating System: Windows 7+
- Server-side Script: Python 3.6+
- IDE: PyCharm
- Libraries Used: Pandas, Numpy, Sklearn, gensim, re.
- Framework: Flask

CHAPTER 4
DESCRIPTION OF PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

Since Chabot mostly relies on artificial intelligence, we have chosen to contribute to the healthcare industry utilizing this technology. The suggested system has a linear user interaction process that starts with symptom extraction, moves on to symptom mapping, identifies the related symptom, determines if the patient has a major or minor condition, and finally proposes the kind of disease and course of therapy. The already trained database will be used to retrieve details. The lack of patient interaction after they leave clinics or hospitals may frequently be helped by our healthcare chatbot solution. Numerous studies in this field have demonstrated how affordable healthcare may be provided through chatbots. The ai-healthcare chatbot system was developed using the Chatterbot Python Library Module.

To solve the aforementioned challenges with frequent health disorders, we have created an artificial intelligence healthcare chatbot system. No user's personal information will be saved by this chatbot for healthcare. Chatbots lack the cognitive capacity necessary to make informed judgments. In order to remove the chatbot's decision-making process and allow individuals to get specific information about their health conditions, we will include a custom Google search. In-depth information fromseveral sources will be provided by integrating custom Google searches from variousreliable websites. This program was created using chatterbot and natural language processing (a python library module). Using Chatterbot, we have trained a chatbot. Using Chatterbot, we have trained the chatbot. To be able to understand human input,we have taught the bot to recognize specific categories of terms. The backend is theninformed of this information.

4.2 ARCHITECTURE / OVERALL DESIGN OF THE PROPOSED SYSTEM

In order to identify specific sorts of phrases and understand the user's intent, we had to train a chatbot using the chatterbot library. The backend will then get this information. It is possible to train the chatbot to provide certain logical reasoning andreplies without using the backend. The system development approach that has been suggested includes medical applications. It elaborates on the system's business features. First, a chatbot that may assist users in learning about their medicalsymptoms is developed. The hospital website is then incorporated, aiding others in learning about the facility and its personnel. Users' records are stored in the system'sdatabase.

The database will process the chatbot's input and turn it into an action that willbe taken. The Admin and User entities make up the proposed system. To use the healthcare online application, the admin must first log in using their login information. Natural language processing separates the user's textual inquiries. The resolution engine aids in decision-making by using the input data that has been supplied to a custom data source. To examine the text's syntax and grammar, the output data is retrieved and sent to the NLG engine. The final message is sent back to the localhostserver and shown in the healthcare chatbot's interface. The discussion records are kept in a database that the administrator may view and change by adding or removingdata. The user must provide personal information, like name, age, date of birth, phonenumber, etc. The user will be asked to describe their symptoms by a chatbot, which will then answer with the condition and offer treatments using natural language processing. After that, it will prompt you to make a hospital appointment. Every member of the staff will have access to the reports because the information is kept inthe hospital database. Finally, the health bot will provide the user the appointment information and let them leave the site by stopping the chat.

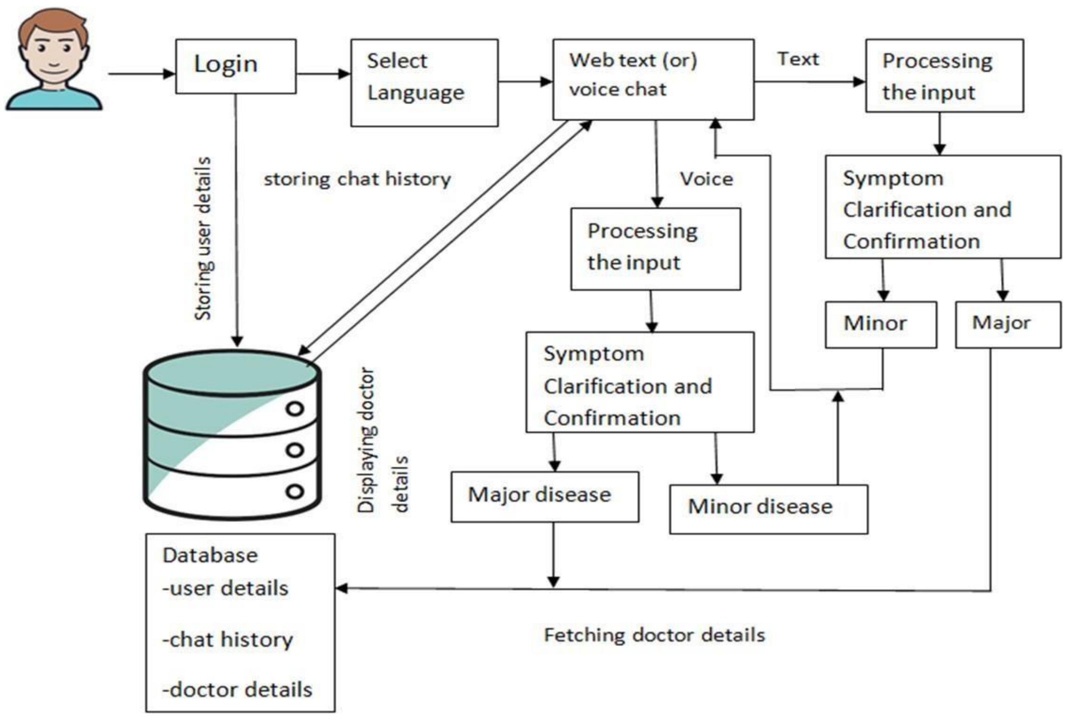


Fig 4.2 System Architecture for Health-Care ChatBot

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

You must first sign up for our chatbot's system before using it. After that, our system will save this data. The chatbot will then connect with you. You will first see abutton to launch the chatbot. The chatbot will begin by asking you questions such as,"What type of symptoms are you having," and then it will display some symptoms onthe screen. Just select "yes" or "no" as your response. If you respond "Yes, these kinds of symptoms do happen to me," the chatbot will use its database to match yoursymptoms, at which point it will recommend the original prescription and provide yourdoctor's contact information so you may call or schedule an appointment.

DESCRIPTION OF PROGRAMMING LANGUAGES

PYTHON

Python is a high-level, interpreted programming language that emphasizes code readability and simplicity. Its syntax is clear and easy to learn, making it an excellent choice for both beginners and experienced programmers. Python is versatileand can be used for a wide range of tasks, including web development, data analysis,scientific computing, and artificial intelligence. The language has a vast standard library and a thriving ecosystem of third-party packages, making it a favorite among developers. Python's popularity also stems from its portability and cross-platform compatibility, allowing developers to write code once and deploy it on multiple platforms.

HTML

HTML, or Hypertext Markup Language, is a markup language used to create and design web pages. It is the standard language for creating documents that can be displayed in web browsers, and it is an essential part of web development. HTMLuses a series of tags and attributes to structure and format content on a web page, such as text, images, and hyperlinks. With HTML, developers can create rich,interactive web pages that are accessible across a variety of devices and platforms. HTML is constantly evolving, with new features and capabilities being added with each new version**.**

DESCRIPTION OF LIBRARIES USE

NUMPY

NumPy is a Python library that is widely used for scientific computing and dataanalysis. It provides a powerful array and matrix processing functionality, along with a variety of other numerical routines. NumPy allows for fast and efficient mathematicaloperations on arrays and matrices, and supports a wide range of mathematical functions such as trigonometric, logarithmic, and exponential functions. It also includes tools for linear algebra, random number generation, and Fourier analysis. NumPy is an essential component of the scientific Python ecosystem and is used extensively in fields such as physics, engineering, finance, and machine learning.

PANDAS

Pandas is a popular Python library used for data manipulation, analysis, and visualization. It provides a wide range of tools and functions for working with tabular and time-series data, making it a powerful tool for data scientists and analysts. The library includes data structures such as Series and DataFrame, which allow for easyindexing, filtering, and grouping of data. Pandas also provides functions for data cleaning and preparation, such as handling missing values and converting data types.Additionally, it integrates well with other Python libraries such as NumPy, Matplotlib, and SciPy. Overall, pandas is a crucial tool in the data science toolkit and is widely used in industry and academia.

FLASK

Flask is a popular Python web framework that is lightweight, flexible, and easyto use. It allows developers to build web applications quickly and efficiently, with minimal boilerplate code. Flask is designed to be modular, so developers can use only the components they need, making it highly customizable. It also has a robust set of extensions available, providing additional functionality such as database integration, user authentication, and more. Flask's simplicity and flexibility make it anexcellent choice for small to medium-sized projects, or for developers who prefer to have more control over the structure and functionality of their web applications.

NLTK or Natural Language Toolkit

NLTK, or Natural Language Toolkit, is a Python library for working with human language data. It provides a suite of tools for text processing, including tokenization, stemming, lemmatization, parsing, and classification. NLTK is widely used in academic research, especially in the fields of computational linguistics and natural language processing, as well as in industry for various language-related applicationssuch as sentiment analysis, named entity recognition, and machine translation. Withits comprehensive documentation, vast community support, and easy-to-use interface, NLTK has become a popular choice for developers and researchers alike who are working with language data.

WARNINGS

The Warnings Python package provides a simple yet effective way to handle warnings in your code. It allows you to catch and suppress warning messages, as well as to modify their behavior. With Warnings, you can control the verbosity of warning messages and decide whether to ignore or treat them as errors. This packageis especially useful for large codebases, where warnings can easily get lost among other output. Warnings are a powerful tool for identifying potential issues in your code,and the Warnings package provides an easy-to-use interface for managing them. It'sa valuable addition to any Python developer's toolkit.

RANDOM

A random package is a software library or module that provides functionality for generating random numbers or selecting random items from a collection. These packages are commonly used in a wide range of applications, such as gaming, statistical analysis, and cryptography. Some popular random packages include NumPy, random, and secrets in Python, as well as the Random class in Java. Thesepackages typically provide a range of methods for generating random numbers or selecting random items, and often allow for the customization of the distribution of thegenerated values. Random packages are an essential tool for any programmer who needs to work with random data.

SKLEARN

Scikit-learn, commonly abbreviated as sklearn, is a powerful open-source machine learning library for the Python programming language. It provides a range oftools for data preprocessing, feature selection, model selection, and evaluation, making it a go-to library for building machine learning models. Sklearn includes a widevariety of popular algorithms, including linear regression, logistic regression, k-nearest neighbors, decision trees, random forests, and support vector machines, among others. It also offers support for a range of data formats, including NumPy arrays and Pandas dataframes. With its user-friendly interface and extensive documentation, sklearn is an excellent choice for both novice and experienced machine learning practitioners looking to build high-quality models quickly and efficiently.

4.4 WORKING MECHANISM

Healthcare chatbots use artificial intelligence (AI) and natural language processing (NLP) technologies to provide healthcare services and assistance to users. Here is the working mechanism of healthcare chatbots:

Understanding user input: When a user interacts with a healthcare chatbot, the chatbot uses NLP technology to understand the user's input. It can recognize the intent behind the user's question or statement, identify relevant keywords, and extractthe information needed to provide an appropriate response.

Providing information and guidance: Based on the user's input, the chatbot can provide information about symptoms, diagnosis, treatment, medications, and other healthcare-related topics. It can also offer guidance on lifestyle changes and preventive measures.

Personalization: Chatbots can be programmed to personalize the information and guidance they provide based on the user's profile, medical history, and preferences. This can improve the accuracy and relevance of the responses.

Integration with healthcare systems: Healthcare chatbots can be integrated with electronic health records (EHRs) and other healthcare systems to access relevant medical information about the user. This integration can enable the chatbot to providemore accurate and personalized responses.

Learning and improvement: As users interact with a healthcare chatbot, the chatbotcan learn from its input and improve its responses over time. This can lead to more accurate and helpful guidance for users.

4.5 PROJECT MANAGEMENT PLAN

Week 1	Web Scraping
	Data Collection
Week 2	EDA Data Processing
	Chatbot Modeling
Week 3	Data Preprocessing
	Chat Modeling
	Creating FAQs
Week 4	Initialization of Web-App
	Integrating the ChatBot
	Deploying the deliverables

CHAPTER 5
IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

NATURAL LANGUAGE PROCESSING

The NLP engine is responsible for parsing human inputs into a format that canbe understood by a computer system, allowing the chatbot to comprehend the contextof the user's inquiry. The NLP engine uses a combination of techniques to analyze and understand natural language inputs, including machine learning and statistical modeling. It employs a series of algorithms to identify the intent and entities of the user's input, which are then used to generate a response.

The NLP engine compares the user's input to the stored intents in the chatbot'sdatabase to decide whether the intentions are supported by that chatbot. Thechatbot's database includes predefined intents that the chatbot has been programmed to recognize and respond to. If the user's intent matches one of the predefined intents, the chatbot will respond accordingly. If not, the chatbot may provide a default response or ask the user for more information.

The NLP engine is critical to the success of a chatbot, as it enables the chatbotto understand and respond to the user's input in a natural and intuitive way. It is the component that makes the chatbot feel like a human-like conversation and allows users to interact with the chatbot in a way that is similar to how they would interact with another human being.

DIALOGUE MANAGEMENT UNIT

Context is crucial in the functioning of a chatbot's dialogue management agent,as it affects how the chatbot responds to user inquiries. In the example provided, thechatbot needs to understand that the user requires a heart doctor, based on their initial statement. However, if the user then changes their mind and asks for a chest doctor instead, the chatbot needs to re-evaluate the context of the inquiry and respondaccordingly.

To achieve this, the chatbot must have a robust dialogue management systemthat can identify changes in context and adjust its responses accordingly. This involves not only understanding the user's intent but also keeping track of the conversation's context and history. The chatbot must be able to recognize when a previous statement is no longer relevant, rephrase it, and provide an appropriate response.

However, it's important to note that the chatbot cannot make any modificationsto the user's request without their approval. The chatbot must seek the user'spermission before making any changes to its previous instruction. This ensures that the chatbot remains respectful of the user's wishes and doesn't make any unwanted modifications to their request.

5.2 ALGORITHMS

We are using two algorithms to implement a healthcare chatbot system.
1. TF-IDF(Term frequency-inverse data frequency)
2. N-gram Algorithm

TF-IDF(Term frequency-inverse data frequency)

Term Frequency(TF)

Text modeling is a technique used to analyze textual data and extract meaningful information from it. One of the common pre-processing steps in text modeling is to remove stop words, which are commonly occurring words that do not carry much meaning in the text. Examples of stop words include "the," "and," "of," and "to." Another technique used to determine the importance of words in text modeling is the Term Frequency-Inverse Document Frequency (TF-IDF) method. TF-IDF is a statistical measure that takes into account both the frequency of a word in a documentand the frequency of that word across the entire corpus of documents being analyzed. The TF-IDF score for a given word in a given document is calculated by dividing the frequency of the word in the document by the frequency of the word across the entirecorpus. This helps to identify the words that are most relevant to the meaning of the text.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

The TF-IDF technique is commonly used in search engines and text-mining applications to identify the most important keywords for further analysis or classification. By ranking the relevance of terms in a document or a corpus, it can helpto identify patterns and trends in the data, as well as highlight the most significant features of the text.

Inverse Document Frequency(IDF)

To further elaborate, TF-IDF is not only used to rank the importance of words in a single document but also in a corpus of documents. In this case, the "document frequency" component of TF-IDF comes into play, which measures how frequently aword appears in the entire corpus. This means that words that are common across multiple documents in the corpus, such as "the" or "and," will receive a lower TF-IDFscore since they are less unique to any specific document. On the other hand, words that appear frequently in a particular document but rarely in the rest of the corpus willreceive a higher TF-IDF score, indicating that they are more important and unique tothat specific document. By using TF-IDF, we can identify the most important words orterms in a corpus, which can help with various natural language processing tasks, such as information retrieval, text classification, and topic modeling.

To elaborate, IDF is used to determine the relative importance of a word in thecontext of the entire corpus of documents. It is calculated by taking the logarithm of the total number of documents in the corpus divided by the number of documents thatcontain the word. This means that words that are rare in the corpus but appear frequently in a particular document will have a higher IDF score, indicating their importance in that document. In contrast, words that are common across all documents will have a lower IDF score and may be considered less important. The combination of TF and IDF gives the TF-IDF score, which is used to rank the relevance of words in a document or corpus.

$$idf(w) = log(\frac{N}{df_t})$$

The IDF, or Inverse Document Frequency, is a metric used in the TF-IDF technique to evaluate the importance of a word in the context of a corpus ofdocuments. It is calculated by taking the logarithm of the total number of documents in the corpus divided by the number of documents containing the word. The IDF is important because it helps to identify words that are rare or infrequent in a corpus. Words that are common across all documents in a corpus, such as "the" or "and," willhave a low IDF score, while words that are rare in the corpus will have a higher IDF score. The idea is that words that are rare are more important and informative for distinguishing between documents.

The logarithmic transformation is used to dampen the effect of very high or very low values. Without this transformation, a word that appears in only a few documents might have an IDF score that is too high, and a word that appears in almostall documents might have an IDF score that is too low. The logarithmic transformationhelps to smooth out the distribution of IDF scores, making them more suitable for usein natural language processing and text mining applications.

5.3 TESTING

Testing of healthcare chatbots is an essential part of the development processto ensure that the chatbot functions as intended and provides accurate information and guidance to users. To achieve optimal results, the following tests are conducted.

Functional testing: This type of testing ensures that the chatbot's functions are working as intended. For example, the chatbot should be able to recognize user input,understand the intent behind the input, and provide relevant responses.

Performance testing: Performance testing evaluates the chatbot's speed, accuracy, and responsiveness under different conditions. This testing can help identify any issues related to the chatbot's speed, capacity, or ability to handle a high volume of users.

Security testing: Security testing ensures that the chatbot is secure and protects users' confidential information. This testing can include vulnerability scans, penetration testing, and testing of access controls.

User acceptance testing: User acceptance testing involves testing the chatbot withreal users to ensure that it meets their needs and expectations. This testing can provide valuable feedback for improving the chatbot's usability and functionality.

Compliance testing: Compliance testing ensures that the chatbot complies with legaland regulatory requirements, such as HIPAA regulations for handling sensitive healthinformation.

Usability testing: Usability testing evaluates the chatbot's ease of use and user experience. This testing can involve gathering feedback from users on the chatbot's design, functionality, and overall performance.

CHAPTER 6
RESULTS AND DISCUSSION

The increased use of messaging applications has led to the rise of chatbots asa new form of healthcare delivery. With the help of tailored treatment provided by medical chatbots, lives can be saved, and people can become more aware of their health issues. Chatbots are convenient as they can be accessed from anywhere, provided there is an internet connection on a desktop computer or smartphone. Thismakes them a valuable tool for people who live in remote or underserved areas.
The conversation between the chatbot and patient is designed to be intuitive and user-friendly, allowing the patient to list their symptoms, and the chatbot responds accordingly.

Overall, the use of chatbots in healthcare has the potential to revolutionize theindustry, making healthcare more accessible and affordable to people around the world. The healthcare industry deals with complex medical conditions, and accurate diagnosis and treatment require in-depth knowledge and experience. Thus, healthcare chatbots must be highly accurate to provide reliable and trustworthy healthcare advice and guidance to patients. The accuracy for this model is represented through line graph.

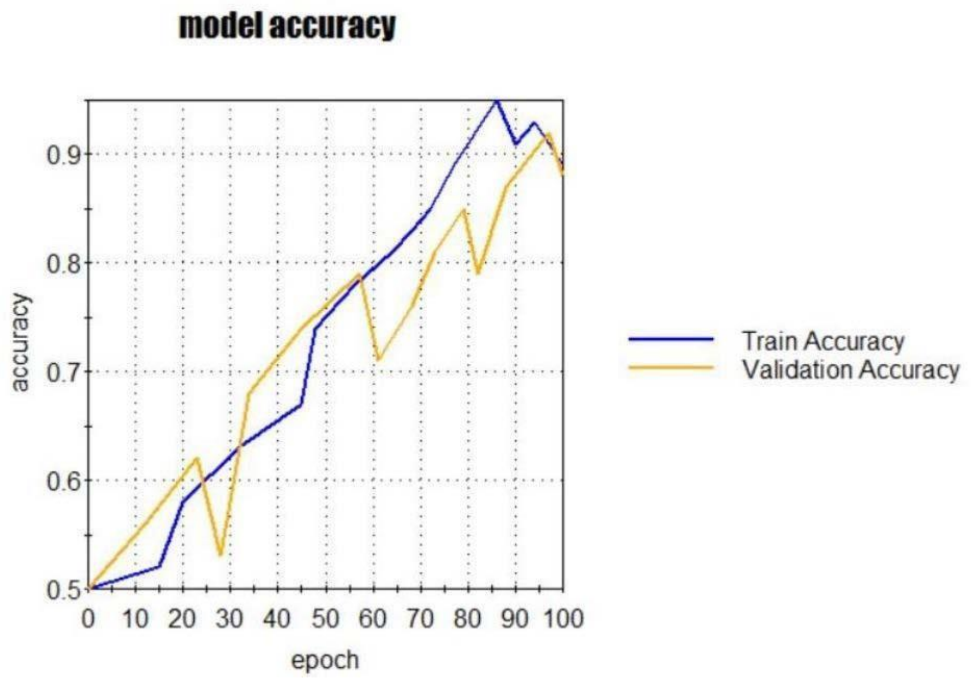


Fig 6 Accuracy of the Model

Moreover, high accuracy healthcare chatbots can help reduce the burden on healthcare providers by providing 24/7 access to healthcare advice, enabling patientsto receive timely guidance without having to visit a medical facility. This can lead to increased patient satisfaction, improved health outcomes, and cost savings for both patients and healthcare providers.

Therefore, the accuracy of healthcare chatbots is of utmost importance, and efforts should be made to continuously improve their accuracy through the use of advanced technologies and data analysis techniques.

CHAPTER 7
CONCLUSION

7.1 CONCLUSION

A chatbot is an artificial intelligence (AI) application that utilizes natural language processing (NLP) to communicate with users via text or voice commands. Chatbots can be designed to simulate human-like conversations with users, providing fast and accurate responses to their inquiries.

Chatbots use a combination of machine learning algorithms, data processing,and AI technologies to analyze and understand the user's queries, intent, and contextto provide relevant responses. These responses can be predefined or generated dynamically based on the user's input.

Chatbots have numerous applications, including customer service, e- commerce, healthcare, education, and many more. For instance, a customer servicechatbot can assist customers with their queries, provide solutions, and help resolve issues quickly and efficiently. In the healthcare industry, chatbots can provide medicaladvice based on the user's symptoms and medical history.

This model can be further made to broaden its scope. To increase the efficiency of medical chatbots, more word combinations can be added to theirdatabases, allowing them to manage a wider range of ailments. By adding more dataand improving algorithms, medical chatbots will become increasingly accurate in diagnosing and treating patients.

Chatbots can be integrated with various messaging platforms, such as Facebook Messenger, WhatsApp, and Slack, making them accessible to users on their preferred messaging platform. The use of chatbots has become increasingly popular in recent years, and it is expected to continue to grow as businesses and organizations look for ways to improve customer engagement and streamline their operations.

7.2 FUTURE WORKS

Healthcare chatbots have already made significant progress in providingsupport and guidance to patients. However, there are several future works in healthcare chatbots that can further enhance their capabilities and impact:

Personalization: Chatbots can be further developed to provide personalized adviceand support to patients based on their medical history, lifestyle, and preferences.

Integration with Electronic Health Records (EHRs): Chatbots can be integrated with EHRs to access patient data and provide more accurate and personalized responses.

Collaboration with Healthcare Professionals: Chatbots can work in conjunction with healthcare professionals to provide patients with continuous support and monitortheir health remotely.

Predictive Analytics: Chatbots can be used to analyze patient data and predict potential health risks, which can then be communicated to healthcare professionals for early intervention.

Integration with Wearable Devices: Chatbots can be integrated with wearable devices to collect real-time health data, monitor progress, and provide feedback to patients.

Mental Health Support: Chatbots can be developed to provide mental health supportto patients, including counseling, therapy, and referrals to mental health professionals.

Multilingual Support: Chatbots can be designed to support multiple languages, making healthcare more accessible to non-native speakers.

CHAPTER 8
REFERENCES

[1] K. Oh, D. Lee, B. Ko and H. Choi, "A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation," 2017 18th IEEE International Conference on Mobile Data Management (MDM), Daejeon, 2017, pp. 371-375. doi: 10.1109/MDM.2017.64.

[2] Du Preez, S.J. & Lall, Manoj & Sinha, S. (2009). An intelligent web-based voice chatbot. 386 - 391.10.1109/EURCON.2009.5167660.

[3] Bayu Setiaji, Ferry Wahyu Wibowo, "Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling", Intelligent Systems Modelling.

[4] Dahiya, Menal. (2017). A Tool of Conversation: Chatbot. INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING. 5. 158-161.2017.

[5] C.P. Shabariram, V. Srinath, C.S. Indhuja, Vidhya (2017). Ratatta: Chatbot Application Using Expert System, International Journal of Advanced Research in Computer Science and Software Engineering,2017.

[6] Mrs. Rashmi Dharwadkar1, Dr.Mrs. Neeta A. Deshpande, A Medical ChatBot, International Journal of Computer Trends and Technology (IJCTT) – Volume 60 Issue 1-June2018.

[7] Farheen Naaz, Farheen Siddiqui, modified n-gram based model for identifying and filteringnear-duplicate documents detection, International Journal of Advanced Computational Engineering and Networking, ISSN: 2320- 2106, Volume-5, Issue-10, Oct.-2017.

[8] N-gram Accuracy Analysis in the Method of Chatbot Response, International Journal of Engineering & Technology. (2018).

[9] Akshay Kumar, Pankaj Kumar Meena, Debiprasanna Panda, Ms. Sangeetha, "Chatbot inPython", Internaltional Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue:11, Nov 2019.

[10] Ayanouz, S., Abdelhakim, B.A., Benhmed, M.: A smart chatbot architecture based NLP and machine learning for health care assistance. In: Proceedings of the 3rd International Conference on Networking, Information Systems & Security, Marrakech, Morocco (2020) Association for Computing Machinery. ACM ISBN 978-1-4503-7634-1.

[11] Ayanouz Soufyane, Boudhir Anouar Abdelhakim, Mohamed Ben Ahmed, An Intelligent Chatbot Using NLP and TF-IDF Algorithm for Text Understanding Applied to the Medical Field, Jan 2021.

[12] L Athota, VK Shukla, N Pandey, Chatbot for Healthcare System using Artificial Intelligence, 8th International Conference on Reliability, Infocom Technologies and Optimization, June 2020.

[13] Shukla, V.K, Verma, A, "Enhancing LMS Experience through AIML Base and Retrieval Base Chatbot using R Language", 2019 International Conference on Automation, Computational and Technology Management (ICACTM).

[14] Thorat SA, Jadhav V (2020) A review on implementation issues of rule-based chatbot systems. In: Proceedings of the international conference on innovative computing andcommunications (ICICC), Apr 2020.

[15] R. Kavitha B., Murthy Chethana R., Chatbot foe Healthcare Sytem using Artificial Intelligence, International Journal of Advance Research, Ideas and Innovations in Technology, ISSN:2454-132X, Volume 5, Issue 3, 2019.

CHAPTER 9
APPENDIX

9.1 SOURCE CODE

server.py

```
from flask import Flask, render_template, request, jsonify, make_response
from bot2 import chat
app = Flask(__name__)

@app.route('/', methods = ['GET', 'POST'])
def indexpage():
    if request.method == "POST":

        print(request.form.get('name'))
        return render_template("index2.html")
    return render_template("index2.html")

@app.route("/entry", methods=['POST'])
def entry():
    req = request.get_json()
    print(req)
    res =
make_response(jsonify({"name":"{}.".format(chat(req)),"message":"OK"}), 200)
    return res


if_name == "_main_":
    app.run(debug=True)
```

bot2,py

```
 # Meet Pybot: your friend
import nltk
import warnings
warnings.filterwarnings("ignore")
# nltk.download() # for downloading packages
#import tensorflow as tf
import numpy as np
import random
```

```python
import string # to process standard python strings

f=open('symptom.txt','r',errors = 'ignore')
m=open('pincodes.txt','r',errors = 'ignore')
checkpoint = "./chatbot_weights.ckpt"
#session = tf.InteractiveSession()
#session.run(tf.global_variables_initializer())
#saver = tf.train.Saver()
#saver.restore(session, checkpoint)

raw=f.read()
rawone=m.read()
nltk.download('punkt') # first-time use only
nltk.download('wordnet') # first-time use only
sent_tokens = nltk.sent_tokenize(raw)# converts to list of sentences
word_tokens = nltk.word_tokenize(raw)# converts to list of words
sent_tokensone = nltk.sent_tokenize(rawone)# converts to list of sentences
word_tokensone = nltk.word_tokenize(rawone)# converts to list of words


sent_tokens[:2]
sent_tokensone[:2]

word_tokens[:5]
word_tokensone[:5]

lemmer = nltk.stem.WordNetLemmatizer()
def LemTokens(tokens):
    return [lemmer.lemmatize(token) for token in tokens]
remove_punct_dict = dict((ord(punct), None) for punct in string.punctuation)
def LemNormalize(text):
    return
LemTokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)))

Introduce_Ans = [" "]
GREETING_INPUTS = ("hello", "hi","hiii","hii","hiiii","hiiii", "greetings",
"sup", "what's up","hey",)
GREETING_RESPONSES = ["Hi,are you suffering from any health issues?(Y/N)",
"Hey,are you having any health issues?(Y/N)", "Hii there,are you having any
health issues?(Y/N)"]
Basic_Q = ("yes","y","yep","yup")
Basic_Ans = "Describe your symptoms"
Basic_Om = ("no","n","nope")
Basic_AnsM ="Thanks for stopping by, we hope to see you again in the near
future"
fev=("iam suffering from fever", "i affected with fever","i have
fever","fever")
feve_r=("which type of fever you have and please mention your symptoms")
```

```python
# Checking for greetings
def greeting(sentence):
    """If user's input is a greeting, return a greeting response"""
    for word in sentence.split():
        if word.lower() in GREETING_INPUTS:
            return random.choice(GREETING_RESPONSES)

# Checking for Basic_Q
def basic(sentence):
    for word in Basic_Q:
        if sentence.lower() == word:
            return Basic_Ans
def fever(sentence):
    for word in fev:
        if sentence.lower() == word:
            return feve_r

# Checking for Basic_QM
def basicM(sentence):
    """If user's input is a greeting, return a greeting response"""
    for word in Basic_Om:
        if sentence.lower() == word:


            return Basic_AnsM
# Checking for Introduce
def IntroduceMe(sentence):
    return random.choice(Introduce_Ans)


from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity


# Generating response
def response(user_response):
    robo_response=''
    sent_tokens.append(user_response)
    TfidfVec = TfidfVectorizer(tokenizer=LemNormalize, stop_words='english')
    tfidf = TfidfVec.fit_transform(sent_tokens)

    vals = cosine_similarity(tfidf[-1], tfidf)

    idx=vals.argsort()[0][-2]
    flat = vals.flatten()
    flat.sort()
    req_tfidf = flat[-2]
    if(req_tfidf==0):
        robo_response=robo_response+"I'm sorry, I didn't quite understand what
you said. Could you please try rephrasing it?"
        return robo_response
    else:
        robo_response = robo_response+sent_tokens[idx]
        return robo_response

# Generating response

# Generating response
def responseone(user_response):
    robo_response=''
    sent_tokensone.append(user_response)
    TfidfVec = TfidfVectorizer(tokenizer=LemNormalize, stop_words='english')
    tfidf = TfidfVec.fit_transform(sent_tokensone)
    vals = cosine_similarity(tfidf[-1], tfidf)
    idx=vals.argsort()[0][-2]
    flat = vals.flatten()
    flat.sort()
    req_tfidf = flat[-2]
    if(req_tfidf==0):
        robo_response=robo_response+"I am sorry! I don't understand you"
        return robo_response
    else:
        robo_response = robo_response+sent_tokensone[idx]
        return robo_response

def chat(user_response):
    user_response=user_response.lower()
    keyword = " module "
    keywordone = " module"
    keywordsecond = "module "

    if(user_response!='bye'):

        if(user_response=='thanks' or user_response=='thank you' ):
            flag=False
            #print("ROBO: You are welcome..")
            return "You are welcome."
        elif(basicM(user_response)!=None):
            return basicM(user_response)
        else:
            if(user_response.find(keyword) != -1 or
user_response.find(keywordone) != -1 or user_response.find(keywordsecond) != -
1):
                #print("ROBO: ",end="")
                #print(responseone(user_response))
                return responseone(user_response)
                sent_tokensone.remove(user_response)
            elif(greeting(user_response)!=None):
                #print("ROBO: "+greeting(user_response))
                return greeting(user_response)
            elif(user_response.find("your name") != -1 or user_response.find("
your name") != -1 or user_response.find("your name ") != -1 or
user_response.find(" your name ") != -1):
                return IntroduceMe(user_response)
            elif(basic(user_response)!=None):
                return basic(user_response)
            elif(fever(user_response)!=None):
                return fever(user_response)
            else:
                #print("ROBO: ",end="")
                #print(response(user_response))
                return response(user_response)
                sent_tokens.remove(user_response)

    else:
        flag=False
        #print("ROBO: Bye! take care..")
        return "Bye! take care.."
```

index2.html

```
 <html>
   <head>
     <style>
      .title{
        text-align: center;
        color:rgb(0, 0, 0);
        font-size: 28px;
      }
     </style>
   </head>
  <body>


  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css
">
```

```
  <link rel='stylesheet prefetch'
href='https://cdnjs.cloudflare.com/ajax/libs/malihu-custom-scrollbar-
plugin/3.1.3/jquery.mCustomScrollbar.min.css'>

  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">




  <div class="chat">
    <div class="chat-title">
      <h1>HEALTHCARE CHATBOT</h1>
      <figure class="avatar">
        <img src="{{ url_for('static', filename='robo1.jpg') }}" /></figure>

        <div  class="r-nav">
       <ul>
            <li> <a>X</a></li>

            <li> <a><img
src="data:image/svg+xml;base64,PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iaXNvLTg
4NTktMSI/ width="26px" /></a></li>


       </ul>

       </div>


    </div>
    <div class="messages">
      <div class="messages-content"></div>
    </div>
    <div class="message-box">
      <textarea type="text" class="message-input" placeholder="Type
message..."></textarea>
      <button type="submit" class="message-submit sound-on-
click">Send</button>
    </div>

  </div>





  <div class="bg"></div>
```

```
    <script
src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></scrip
t>
  <script src='https://cdnjs.cloudflare.com/ajax/libs/malihu-custom-scrollbar-
plugin/3.1.3/jquery.mCustomScrollbar.concat.min.js'></script>

      <script src="{{ url_for('static', filename='sound.js') }}"></script>
</body>
  </html>
```

style.css

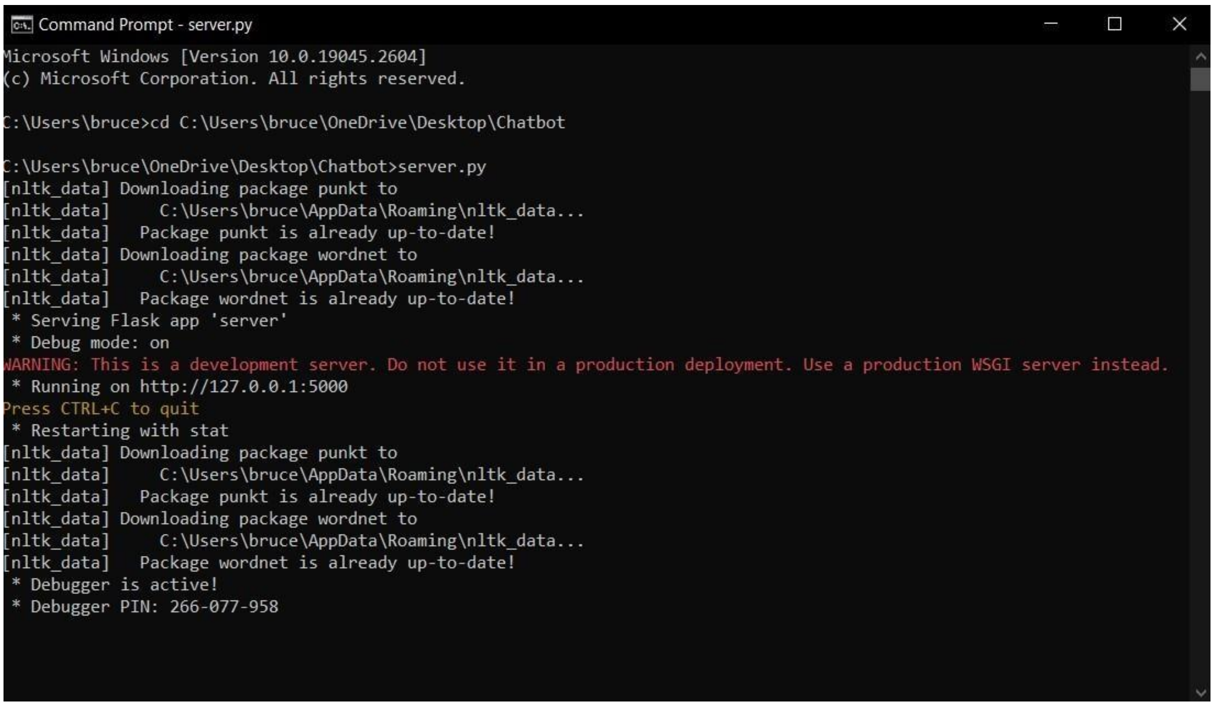
```
/*--------------------------------------------
Mixins
---------------------------------------------*/
/*-----------------------------------
Body
-----------------------------------*/
*,
*::before,
*::after {
  box-sizing: border-box;
}

html,
body {

  height: 100%;

}

body {
  background: -webkit-linear-gradient(315deg, #044f4800, #2a756100);
  background: linear-gradient(135deg, #044f4800, #2a756100);
  background-size: cover;
  font-family: 'Open Sans', sans-serif;
  font-size: 12px;
  line-height: 1.3;
  overflow: hidden;
}


.r-nav{     position: absolute;
    right: 9px;
    width: 100px;
    top: 0px;
    height: 45px;
    }

.r-nav li {
   list-style: none;
    float:  right;
    display: inline-block;
    width: 30px;

    text-align: center;
}
.r-nav li a {
       margin: 0;
    padding: 0;
    line-height: 18px;
    font-size: 16px;
    color: #97979700;

}

.bg {
  width: 100%;
  height: 100%;
  top: 0;
  left: 0;
  z-index: 1;
  background: url(ba.jpg);
  background-size: 3000px 1000px;
  background-repeat: no-repeat ;



}


.loading-img{
    width: 100px;
    display: inline;
    margin: 10px 80px;
    height: 100px;}


.buttonx {
    background: #0070ff;;
    padding:  5px  30px;
    margin: 5px 5px 5px 0px;
    border: 1px solid #0070ff;
    color: #fff;
    border-radius: 10px;
```

```
}

.buttony { background: transparent;
    padding: 5px 30px;
    margin: 5px 5px 5px 0px;
    border: 1px solid #0070ff;
  color: #0070ff;;
    border-radius: 10px;
}


.oracle-search{width: 100%;
    margin: 7px auto;
    height: 29px;
    padding: 5px;
    font-size: 12px;
    border-radius: 15px;
    border: 1px solid rgba(255, 255, 255, 0.6);
    color: #ffffff00;
    background: rgba(68, 68, 68, 0.47);}

/*-------------------------------------------
Chat
----------------------------------------*/


input[type="range"] {
 -webkit-appearance: none;
    width: 340px;
    height: 2px;
    background: #0070ff;
    background-position: center;
    background-repeat: no-repeat;
    position: absolute;
    top:  0px;
    bottoM: 0px;
    left:  0px;
    right: 0px;
    margin: auto;
    display: block;
    margin-top: 60px;
}

input[type="range"]::-webkit-slider-thumb {
  -webkit-appearance: none;
  width: 20px;
  height: 20px;
  border-radius: 100%;
  background: #434343;
  position: relative;
  border: 3px solid #0070ff;
  z-index: 3;
  cursor: pointer;
  content: counter(3)
}




.chat {
  position: absolute;
  top: 53%;
  left: 50%;
  -webkit-transform: translate(-50%, -50%);
          transform: translate(-50%, -50%);
  width: 1300px;
  height: 90vh;
  max-height: 640px;
  z-index: 2;
  overflow: hidden;
  box-shadow: 0 5px 30px rgba(0, 0, 0, 0);
  background: rgba(0, 0, 0, 0);
  border-radius: 20px;
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-pack: justify;
      -ms-flex-pack: justify;
          justify-content: space-between;
  -webkit-box-orient: vertical;
  -webkit-box-direction: normal;
      -ms-flex-direction:  column;
          flex-direction: column;
}

/*--------------------------------------------
Chat Title
--------------------------------------------*/
.chat-title {
  -webkit-box-flex: 0;
      -ms-flex: 0 1 45px;
          flex: 0 1 45px;
  position: relative;
  z-index: 2;
  background: rgb(126, 186, 233);
  color: #000000;
  text-transform: uppercase;
  text-align: left;
  padding: 10px 10px 10px 50px;
  font-weight: bolder;
}
.chat-title h1, .chat-title h2 {
  font-weight: bold;
  font-size: 14px;
  margin: 0;
  padding: 0;
}

.chat-title .avatar {
  position: absolute;
  z-index: 1;
  top: 8px;
  left: 9px;
  border-radius: 30px;
  width: 30px;
  height: 30px;
  overflow: hidden;
  margin: 0;
  padding: 0;
  border: 2px solid rgba(255, 255, 255, 0.24);
}
.chat-title .avatar img {
  width: 100%;
  height: auto;
}

/*--------------------
Messages
--------------------*/
.messages {
  -webkit-box-flex: 1;
      -ms-flex: 1 1 auto;
          flex: 1 1 auto;
  color: rgb(239, 245, 243);
  overflow: hidden;
  position:  relative;
  width: 100%;
}
.messages .messages-content {
  position: absolute;
  top: 0;
  left: 0;
  height: 101%;
  width: 100%;
}
.messages .message {
  clear: both;
  float: left;
  padding: 6px 10px 7px;
  border-radius: 10px 10px 10px 0;
  background: rgba(0, 0, 0, 0.3);
  margin: 8px 0;
  font-size: 14px;
  line-height: 1.4;
  margin-left: 35px;
  position: relative;
  text-shadow: 0 1px 1px rgba(0, 0, 0, 0.2);
}
.messages .message .timestamp {
  position: absolute;
  bottom: -19px;
  font-size: 9px;
  color: rgb(148, 222, 245);
}
.messages .message::before {
  content: '';
  position: absolute;
  bottom: -6px;
  border-top: 6px solid rgba(0, 0, 0, 0.3);
  left: 0;
  border-right: 7px solid transparent;
}
.messages .message .avatar {
  position: absolute;
  z-index: 1;
  bottom: -15px;
  left: -35px;
  border-radius: 30px;
  width: 30px;
  height: 30px;
  overflow: hidden;
  margin: 0;
  padding: 0;
  border: 2px solid rgb(145, 198, 236);
}
.messages .message .avatar img {
  width: 100%;
  height: auto;
}
.messages .message.message-personal {
  float: right;
  color: #000000;
  text-align: right;
  background: linear-gradient(120deg, #a1f6f9, #a1f6f9);
  border-radius: 10px 10px 0 10px;
}
.messages .message.message-personal::before {
  left: auto;
  right: 0;
  border-right: none;
  border-left: 5px solid transparent;
  border-top: 4px solid #257287;
  bottom: -4px;
}
.messages .message:last-child {
  margin-bottom: 30px;
}
.messages .message.new {
  -webkit-transform: scale(0);
          transform: scale(0);
  -webkit-transform-origin: 0 0;
          transform-origin: 0 0;
  -webkit-animation: bounce 500ms linear both;
          animation: bounce 500ms linear both;
}
.messages .message.loading::before {
  position: absolute;
  top: 50%;
  left: 50%;
  -webkit-transform: translate(-50%, -50%);
          transform: translate(-50%, -50%);
  content: '';
  display: block;
  width:  3px;
  height: 3px;
  border-radius: 50%;
  background: rgb(115, 189, 204);
  z-index: 2;
  margin-top: 4px;
  -webkit-animation: ball 0.45s cubic-bezier(0, 0, 0.15, 1) alternate
infinite;
          animation: ball 0.45s cubic-bezier(0, 0, 0.15, 1) alternate
infinite;
  border: none;
  -webkit-animation-delay: .15s;
          animation-delay: .15s;
}
.messages .message.loading span {
  display: block;
  font-size: 0;
  width: 20px;
  height: 10px;
  position: relative;
}
.messages .message.loading span::before {
  position: absolute;
  top: 50%;
  left: 50%;
  -webkit-transform: translate(-50%, -50%);
          transform: translate(-50%, -50%);
  content: '';
  display: block;
  width:  3px;
  height: 3px;
  border-radius: 50%;
  background: rgb(136, 232, 249);
  z-index: 2;
  margin-top: 4px;
  -webkit-animation: ball 0.45s cubic-bezier(0, 0, 0.15, 1) alternate
infinite;
          animation: ball 0.45s cubic-bezier(0, 0, 0.15, 1) alternate
infinite;
  margin-left: -7px;
}
.messages .message.loading span::after {
  position: absolute;
  top: 50%;
  left: 50%;
  -webkit-transform: translate(-50%, -50%);
          transform: translate(-50%, -50%);
  content: '';
  display: block;
  width:  3px;
  height: 3px;
  border-radius: 50%;
  background: rgb(158, 227, 248);
  z-index: 2;
  margin-top: 4px;
  -webkit-animation: ball 0.45s cubic-bezier(0, 0, 0.15, 1) alternate
infinite;
          animation: ball 0.45s cubic-bezier(0, 0, 0.15, 1) alternate
infinite;
  margin-left: 7px;
  -webkit-animation-delay: .3s;
          animation-delay: .3s;
}

/*--------------------
Message Box
--------------------*/
```

```css
.message-box {
  -webkit-box-flex: 0;
      -ms-flex: 0 1 40px;
          flex: 0 1 40px;
  width: 100%;
  background: rgba(0, 0, 0, 0.3);
  padding: 10px;
  position: relative;
}
.message-box .message-input {
  background: none;
  border: none;
  outline: none !important;
  resize: none;
  color: rgb(255, 255, 255);
  font-size: 11px;
  height: 40px;
  margin: 0;
  padding-right: 20px;
  width: 1200px;
}
textarea {
  font-family: sans-serif; /* 1 */
  font-size: 100%; /* 1 */
  line-height: 2; /* 1 */
  margin: 0; /* 2 */
}
.message-box textarea:focus:-webkit-placeholder {
  color: transparent;
}
.message-box .message-submit {
  position: absolute;
  z-index: 1;
  top: 9px;
  right: 10px;
  color: #000000;
  border: none;
  background: #9ad0de;
  font-size: 10px;
  text-transform: uppercase;
  line-height: 1.5;
  padding: 6px 10px;
  border-radius: 10px;
  outline: none !important;
  -webkit-transition: background .2s ease;
  transition: background .2s ease;
}
.message-box .message-submit:hover {
  background: #ffffff;
}

/*-----------------------------------------
Custom Srollbar
----------------------------------------*/
.mCSB_scrollTools {
  margin: 1px -3px 1px 0;
  opacity: 0;
}

.mCSB_inside > .mCSB_container {
  margin-right: 0px;
  padding: 0 10px;
}

.mCSB_scrollTools .mCSB_dragger .mCSB_dragger_bar {
  background-color: rgba(0, 0, 0, 0.5) !important;
}

/*-----------------------------------------
Bounce
----------------------------------------*/
@-webkit-keyframes bounce {
  0% {
    -webkit-transform: matrix3d(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
1);
            transform: matrix3d(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
1);
  }
  4.7% {
    -webkit-transform: matrix3d(0.45, 0, 0, 0, 0, 0.45, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.45, 0, 0, 0, 0, 0.45, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  9.41% {
    -webkit-transform: matrix3d(0.883, 0, 0, 0, 0, 0.883, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.883, 0, 0, 0, 0, 0.883, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  14.11% {
    -webkit-transform: matrix3d(1.141, 0, 0, 0, 0, 1.141, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(1.141, 0, 0, 0, 0, 1.141, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  18.72% {
    -webkit-transform: matrix3d(1.212, 0, 0, 0, 0, 1.212, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(1.212, 0, 0, 0, 0, 1.212, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  24.32% {
    -webkit-transform: matrix3d(1.151, 0, 0, 0, 0, 1.151, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(1.151, 0, 0, 0, 0, 1.151, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  29.93% {
    -webkit-transform: matrix3d(1.048, 0, 0, 0, 0, 1.048, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(1.048, 0, 0, 0, 0, 1.048, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  35.54% {
    -webkit-transform: matrix3d(0.979, 0, 0, 0, 0, 0.979, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.979, 0, 0, 0, 0, 0.979, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  41.04% {
    -webkit-transform: matrix3d(0.961, 0, 0, 0, 0, 0.961, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.961, 0, 0, 0, 0, 0.961, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  52.15% {
    -webkit-transform: matrix3d(0.991, 0, 0, 0, 0, 0.991, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.991, 0, 0, 0, 0, 0.991, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  63.26% {
    -webkit-transform: matrix3d(1.007, 0, 0, 0, 0, 1.007, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(1.007, 0, 0, 0, 0, 1.007, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  85.49% {
    -webkit-transform: matrix3d(0.999, 0, 0, 0, 0, 0.999, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.999, 0, 0, 0, 0, 0.999, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  100% {
    -webkit-transform: matrix3d(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
1);
            transform: matrix3d(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
1);
  }

}
@keyframes bounce {
  0% {
    -webkit-transform: matrix3d(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
1);
            transform: matrix3d(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
1);
  }
  4.7% {

    -webkit-transform: matrix3d(0.45, 0, 0, 0, 0, 0.45, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.45, 0, 0, 0, 0, 0.45, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  9.41% {
    -webkit-transform: matrix3d(0.883, 0, 0, 0, 0, 0.883, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.883, 0, 0, 0, 0, 0.883, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  14.11% {
    -webkit-transform: matrix3d(1.141, 0, 0, 0, 0, 1.141, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(1.141, 0, 0, 0, 0, 1.141, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  18.72% {
    -webkit-transform: matrix3d(1.212, 0, 0, 0, 0, 1.212, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(1.212, 0, 0, 0, 0, 1.212, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  24.32% {
    -webkit-transform: matrix3d(1.151, 0, 0, 0, 0, 1.151, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(1.151, 0, 0, 0, 0, 1.151, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  29.93% {
    -webkit-transform: matrix3d(1.048, 0, 0, 0, 0, 1.048, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
```

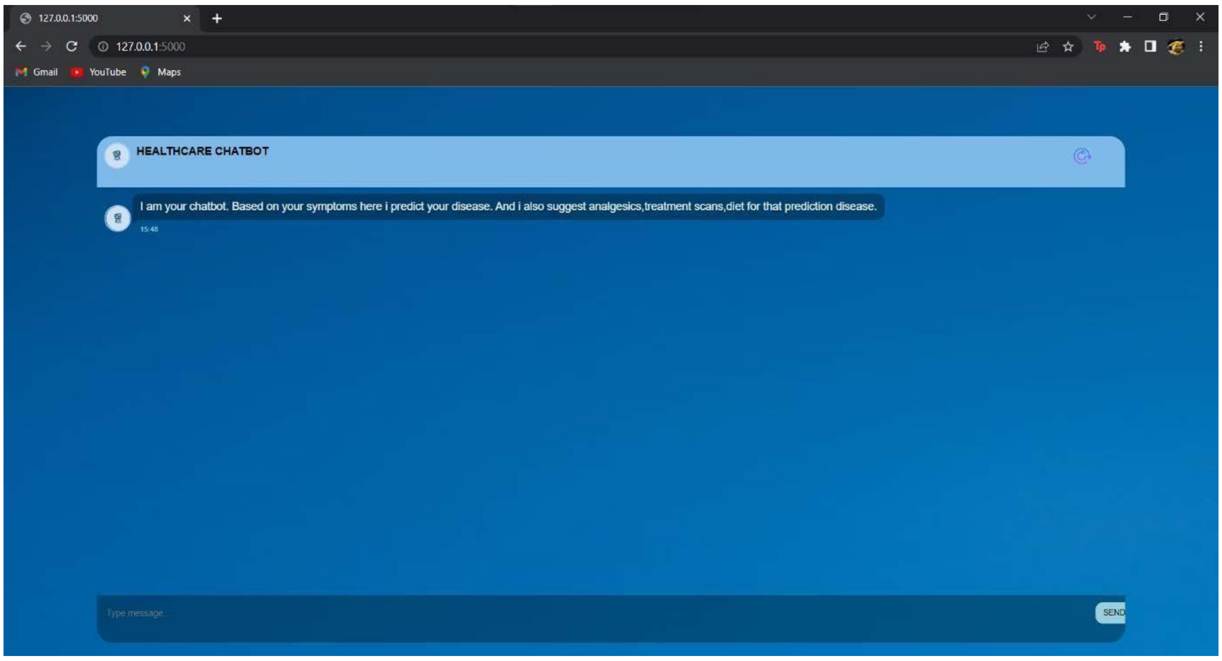
```
            transform: matrix3d(1.048, 0, 0, 0, 0, 1.048, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  35.54% {
    -webkit-transform: matrix3d(0.979, 0, 0, 0, 0, 0.979, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.979, 0, 0, 0, 0, 0.979, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  41.04% {
    -webkit-transform: matrix3d(0.961, 0, 0, 0, 0, 0.961, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.961, 0, 0, 0, 0, 0.961, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  52.15% {
    -webkit-transform: matrix3d(0.991, 0, 0, 0, 0, 0.991, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.991, 0, 0, 0, 0, 0.991, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  63.26% {
    -webkit-transform: matrix3d(1.007, 0, 0, 0, 0, 1.007, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(1.007, 0, 0, 0, 0, 1.007, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  85.49% {
    -webkit-transform: matrix3d(0.999, 0, 0, 0, 0, 0.999, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
            transform: matrix3d(0.999, 0, 0, 0, 0, 0.999, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1);
  }
  100% {

    -webkit-transform: matrix3d(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
1);
            transform: matrix3d(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
1);
  }

}
@-webkit-keyframes ball {
  from {
    -webkit-transform: translateY(0) scaleY(0.8);
            transform: translateY(0) scaleY(0.8);
  }
  to {
    -webkit-transform: translateY(-10px);
            transform: translateY(-10px);
  }
}
@keyframes ball {
  from {
    -webkit-transform: translateY(0) scaleY(0.8);
            transform: translateY(0) scaleY(0.8);
  }
  to {
    -webkit-transform: translateY(-10px);
            transform: translateY(-10px);
  }
}
```

9.2 SCREENSHOTS



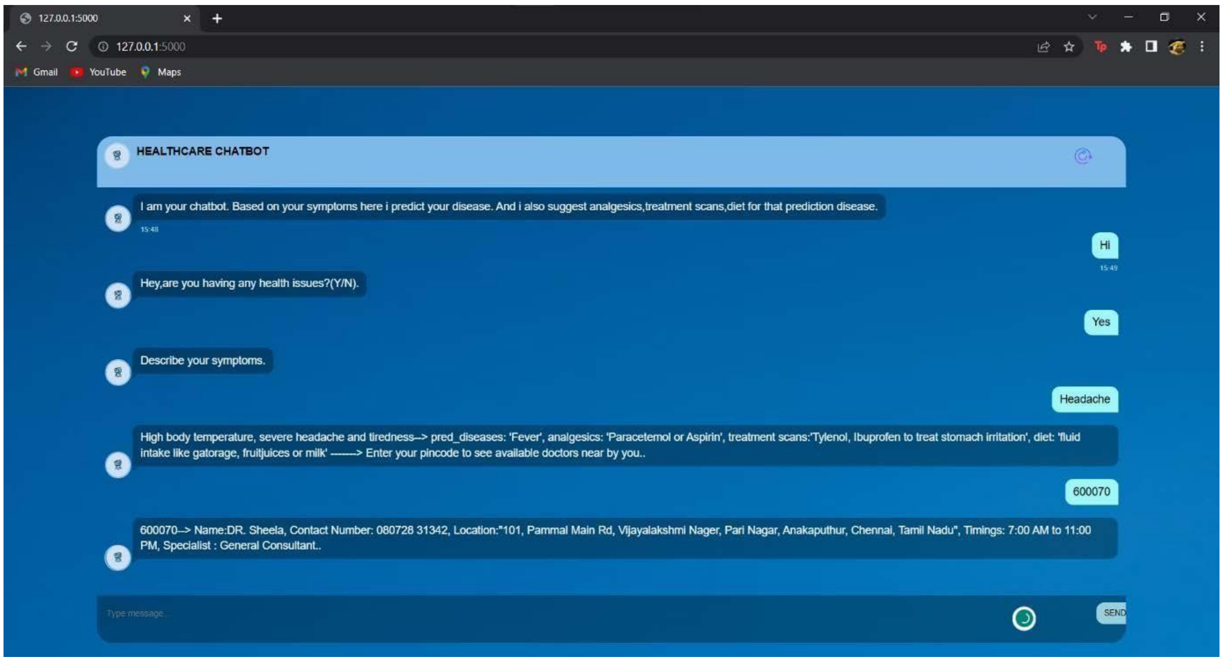
```
Command Prompt - server.py
Microsoft Windows [Version 10.0.19045.2604]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bruce>cd C:\Users\bruce\OneDrive\Desktop\Chatbot

C:\Users\bruce\OneDrive\Desktop\Chatbot>server.py
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\bruce\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\bruce\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
 * Serving Flask app 'server'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\bruce\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\bruce\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
 * Debugger is active!
 * Debugger PIN: 266-077-958
```

Fig 9.2.1 Initiating the server



I am your chatbot. Based on your symptoms here i predict your disease. And i also suggest analgesics,treatment scans,diet for that prediction disease.
15:48

9.2.2 Chatbot is initiated



I am your chatbot. Based on your symptoms here i predict your disease. And i also suggest analgesics,treatment scans,diet for that prediction disease.
15:48

Hi
15:49

Hey,are you having any health issues?(Y/N).

Yes

Describe your symptoms.

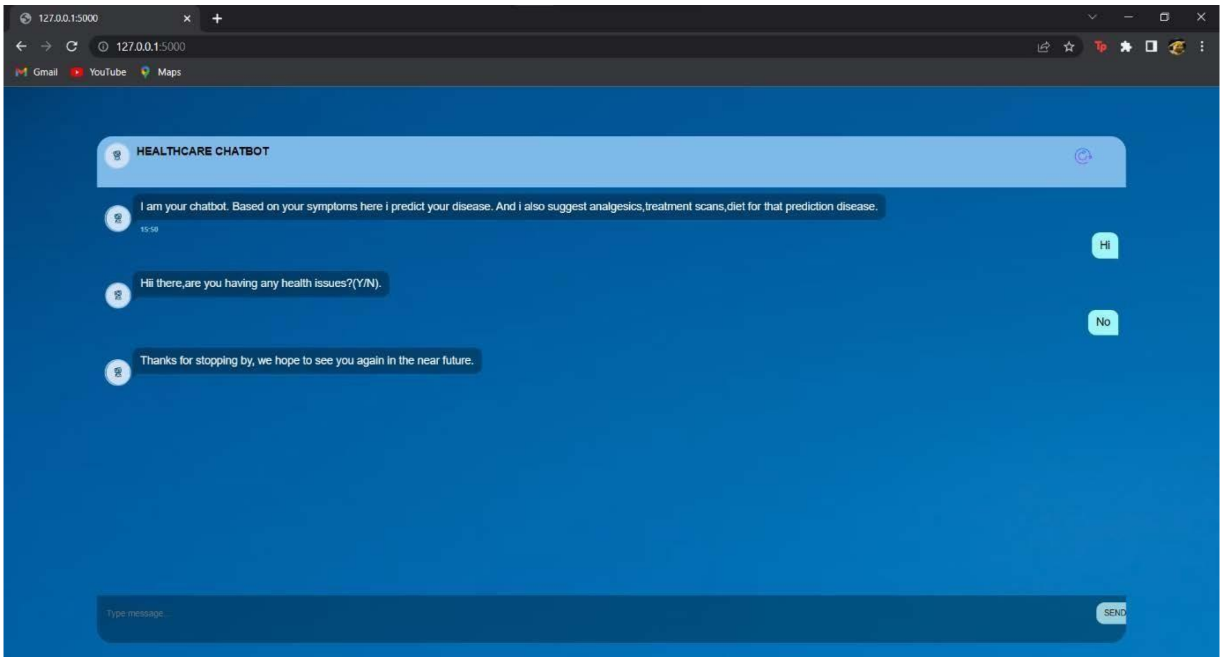
Headache

High body temperature, severe headache and tiredness--> pred_diseases: 'Fever', analgesics: 'Paracetemol or Aspirin', treatment scans:'Tylenol, Ibuprofen to treat stomach irritation', diet: 'fluid intake like gatorage, fruitjuices or milk' -------> Enter your pincode to see available doctors near by you..

600070

600070--> Name:DR. Sheela, Contact Number: 080728 31342, Location:"101, Pammal Main Rd, Vijayalakshmi Nager, Pari Nagar, Anakaputhur, Chennai, Tamil Nadu", Timings: 7:00 AM to 11:00 PM, Specialist : General Consultant..

9.2.3 Determining the illness using symptoms



I am your chatbot. Based on your symptoms here i predict your disease. And i also suggest analgesics,treatment scans,diet for that prediction disease.

15:50

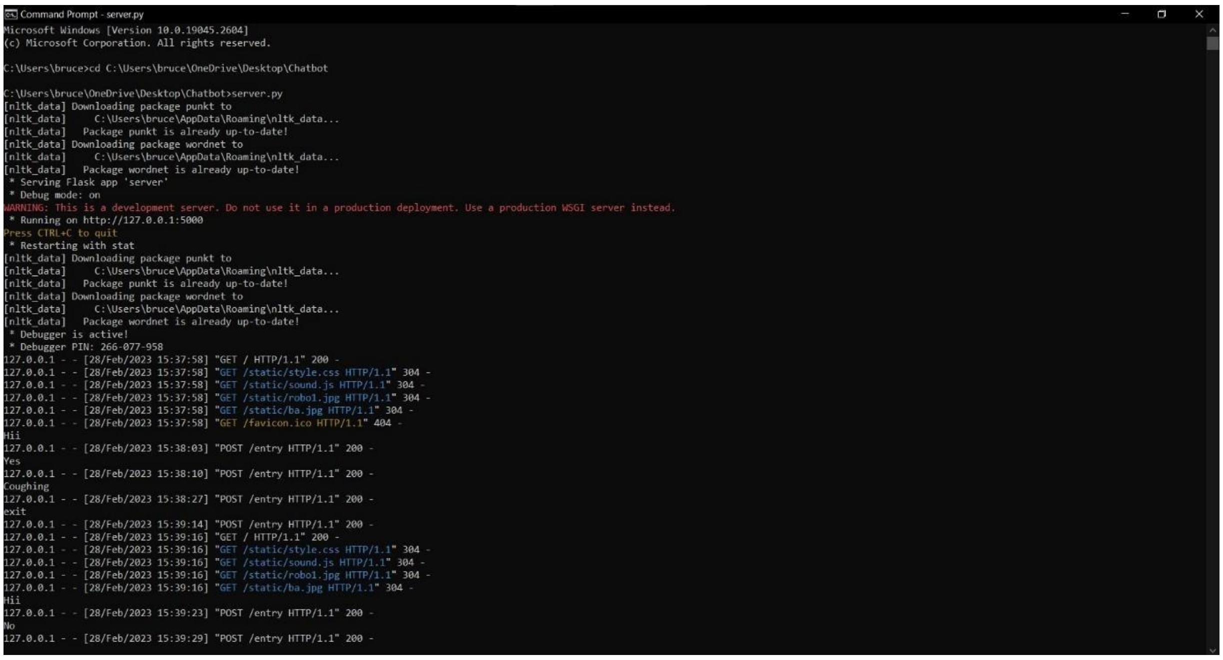
Hi

Hii there,are you having any health issues?(Y/N).

No

Thanks for stopping by, we hope to see you again in the near future.

9.2.4 Casual interaction with Chatbot



```
Microsoft Windows [Version 10.0.19045.2604]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bruce>cd C:\Users\bruce\OneDrive\Desktop\Chatbot

C:\Users\bruce\OneDrive\Desktop\Chatbot>server.py
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\bruce\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\bruce\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
 * Serving Flask app 'server'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\bruce\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\bruce\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
 * Debugger is active!
 * Debugger PIN: 266-077-958
127.0.0.1 - - [28/Feb/2023 15:37:58] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Feb/2023 15:37:58] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [28/Feb/2023 15:37:58] "GET /static/sound.js HTTP/1.1" 304 -
127.0.0.1 - - [28/Feb/2023 15:37:58] "GET /static/robo1.jpg HTTP/1.1" 304 -
127.0.0.1 - - [28/Feb/2023 15:37:58] "GET /static/ba.jpg HTTP/1.1" 304 -
127.0.0.1 - - [28/Feb/2023 15:37:58] "GET /favicon.ico HTTP/1.1" 404 -
Hii
127.0.0.1 - - [28/Feb/2023 15:38:03] "POST /entry HTTP/1.1" 200 -
Yes
127.0.0.1 - - [28/Feb/2023 15:38:10] "POST /entry HTTP/1.1" 200 -
Coughing
127.0.0.1 - - [28/Feb/2023 15:38:27] "POST /entry HTTP/1.1" 200 -
exit
127.0.0.1 - - [28/Feb/2023 15:39:14] "POST /entry HTTP/1.1" 200 -
127.0.0.1 - - [28/Feb/2023 15:39:16] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Feb/2023 15:39:16] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [28/Feb/2023 15:39:16] "GET /static/sound.js HTTP/1.1" 304 -
127.0.0.1 - - [28/Feb/2023 15:39:16] "GET /static/robo1.jpg HTTP/1.1" 304 -
127.0.0.1 - - [28/Feb/2023 15:39:16] "GET /static/ba.jpg HTTP/1.1" 304 -
Hii
127.0.0.1 - - [28/Feb/2023 15:39:23] "POST /entry HTTP/1.1" 200 -
No
127.0.0.1 - - [28/Feb/2023 15:39:29] "POST /entry HTTP/1.1" 200 -
```

9.2.5 Communication between the server and the chatbot