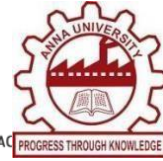




**K. RAMAKRISHNAN
COLLEGE OF ENGINEERING**
An Autonomous Institution

Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.



A PROJECT REPORT

on

HEALTH CARE CHATBOT

Submitted in partial fulfillment of requirements for the award of the course of

CGB1121–PYTHON PROGRAMMING

Under the guidance of

Ms. M. INDHU

Assistant Professor/ECE

Submitted By

S.VIKRAM (8115U23EC121)

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING
K . RAMAKRISHNAN COLLEGE OF ENGINEERING**

(An Autonomous Institution ,affiliated to Anna University Chennai
and Approved by AICTE, New Delhi)

SAMAYAPURAM–621112

MAY 2024



**K. RAMAKRISHNAN
COLLEGE OF ENGINEERING**
An Autonomous Institution

Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC

Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.



K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM-621112

MAY 2024

BONAFIDECERTIFICATE

Certified that this project report titled “**HEALTH CARE CHATRBOT**” is the bonafide work of **S.VIKRAM (8115U23EC121)**, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here it does not form part of anyother project report or dissertation on the basis of which a degree or award was conferred onan earlier occasion on this or any other candidate.

SIGNATURE

Dr. M. MAHESWARI

M.Tech.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR,

Department of Electronics and communication

Engineering,

K. Ramakrishnan College of Engineering

(Autonomous)

Samayapuram – 621 112

SIGNATURE

Ms. M. INDHU M.E,

SUPERVISOR

ASSISTANT PROFESSOR,

Department of Electronics and

communication Engineering,,

K. Ramakrishnan College of Engineering

(Autonomous)

Samayapuram – 621 112

Submitted for the End Semester Examination held on.....

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I jointly declare that the project report on “**HEALTH CARE CHATBOT**” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfillment of the requirement of the award of degree of **BACHELOR OF ENGINEERING**.

Signature

VIKRAM S

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and indebtedness to our institution, “**K. Ramakrishnan College of Engineering (Autonomous)**”, for providing us with the opportunity to do this project.

We extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN, B.E.**, for having provided the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding our project and offering an adequate duration to complete it.

We would like to thank **Dr. D. SRINIVASAN, B.E, M.E.,Ph.D.**, Principal , who gave the opportunity to frame the project to full satisfaction.

We thank **Dr. M. MAHESWARI, M . t e c h , Ph.D.**, Head of the Department of **ELECTRONICS AND COMMUNICATION ENGINEERING**, for providing her encouragement in pursuing this project.

We wish to convey our profound and heartfelt gratitude to our esteemed project guide **Ms. M.INDHU M.E.**, Department of **ELECTRONICS AND COMMUNICATION ENGINEERING**, for her incalculable suggestions, creativity , assistance and patience, which motivated us to carry out this project.

We render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.



**K.RAMAKRISHNAN
COLLEGE OF ENGINEERING**
An Autonomous Institution



Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

VISION

To achieve a prominent position among the top technical institutions

MISSION

- To bestow standard technical education par excellence through state of the art infrastructure, competent faculty and high ethical standards.
- To nurture research and entrepreneurial skills among students in cutting edge technologies.
- To provide education for developing high-quality professionals to transform the society.

DEPARTMENT VISION AND MISSION

VISION

To be a global leader in Electronics and Communication Engineering, driving innovation and excellence for societal advancement.

MISSION

1. Educational Excellence: Deliver comprehensive and practical education.
2. Research and Innovation: Foster cutting-edge research and innovation.
3. Ethical Growth: Promote ethics and professionalism.
4. Industry Collaboration: Partner with industry for relevance and opportunities.
5. Lifelong Learning: Encourage continuous professional development.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will become experts in providing solution for the Engineering problems in Industries, Government and other organizations where they are employed.

PEO2: Graduates will provide innovative ideas and management skills to enhance the standards of the society by individual and with team works through the acquired Engineering knowledge.

PEO3: Graduates will be successful professionals through lifelong learning and contribute to the society technically and professionally.

PROGRAMOUTCOMES(POs)

Engineering students will be able to:

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals ,and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3:Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4:Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data ,and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety ,legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in society a land environmental contexts ,and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long

learning in the broadest context of technological change.

PROGRAMSPECIFICOUTCOMES(PSOs)

PSO1: Students will qualify in National level Competitive Examinations for Employment and Higher studies.

PSO2: Students will have expertise in the design and development of Hardware and Software tools to solve complex Electronics and Communication Engineering problems in the domains like analog and digital electronics, embedded and communication systems.

ABSTRACT

In recent years, advancements in artificial intelligence (AI) and natural language processing (NLP) have paved the way for the development of intelligent healthcare chatbots. These chatbots serve as virtual assistants, offering support and information to patients and healthcare providers alike. This paper presents an overview of the design, implementation, and potential impact of healthcare chatbots on patient care and medical practice. By leveraging AI and NLP, healthcare chatbots can provide a range of services, including answering medical queries, scheduling appointments, sending medication reminders, and offering mental health support. They can operate 24/7, ensuring that patients have access to assistance at all times. The integration of chatbots into healthcare systems has shown promise in enhancing patient engagement, improving health outcomes, and reducing the burden on healthcare professionals. This abstract discusses the benefits, challenges, and future directions of healthcare chatbots, emphasizing their role in creating a more efficient, accessible, and patient-centered healthcare ecosystem.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	ix
	LIST OF FIGURES	xii
1	INTRODUCTION	
	1.1 Introduction	1
	1.2 Objectives	2
	1.3 Project Summarization	3
2	PROJECT METHODOLOGY	
	2.1 Introduction to system Architecture	4
	2.2 Detailed System Architecture Diagram	5
3	PYTHON PREFERENCE	
	3.1 Why Python Prefers IMDb()	7
	3.2 Features of python	7
4	DATASTRUCTURES USED	
	4.1 Dictionary	10
	4.2 List	10
	4.3 Tuple	11
	4.4 Set	11
5	MODULES	
	5.1 Ensure python is installed	12
	5.2 Save the script	12
	5.3 Run the script	13

6	ERROR MANAGEMENT	14
	6.1 Input Validation	
	6.2 Exceptional Handling	15
	6.3 Data Validation	15
7	RESULT & DISCUSSION	
	7.1 Result	16
	7.2 Discussion	17
8	CONCLUSION & FUTURE SCOPE	
	8.1 Conclusion	18
	8.2 Future Scope	19
9	REFERENCES	20
	APPENDIX – A	21

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
2.2	Architecture Diagram	6

CHAPTER-1

INTRODUCTION

1.1 Introduction:

The healthcare industry is undergoing a significant transformation driven by the integration of advanced technologies. Among these innovations, healthcare chatbots have emerged as powerful tools that leverage artificial intelligence (AI) and natural language processing (NLP) to provide a wide range of services to patients and healthcare providers. These virtual assistants are designed to interact with users in a conversational manner, offering support, information, and guidance on various health-related issues.

1.1.1 Purpose and Importance:

The purpose of healthcare chatbots is to enhance the delivery of healthcare services through the use of artificial intelligence (AI) and natural language processing (NLP). These virtual assistants are designed to provide a range of services that improve patient engagement, streamline administrative processes, and support clinical decision-making.

- **Patient Support and Engagement:** 24/7 Availability: Provide round-the-clock assistance to patients, addressing their queries and concerns without the need for human intervention.
- **Medical Information:** Offer reliable and accurate information on symptoms, conditions, treatments, and medications.
- **Appointment Scheduling:** Facilitate the booking and management of appointments, reducing the workload on administrative staff.

1.2 Objectives:

The purpose of healthcare chatbots is to enhance the delivery of healthcare services through the use of artificial intelligence (AI) and natural language processing (NLP). These virtual assistants are designed to provide a range of services that improve patient engagement, streamline administrative processes, and support clinical decision-making

1. **Patient Support and Engagement:** Provide round-the-clock assistance to patients, addressing their queries and concerns without the need for human intervention.
2. **Medication Management:** Send reminders for medication intake, prescription refills, and follow-up appointments, ensuring adherence to treatment plans.
3. **Mental Health Support:** Provide instant support and resources for individuals experiencing mental health issues, such as anxiety or depression.
4. **Clinical Decision Support:** Help in preliminary diagnosis by evaluating symptoms and suggesting potential conditions or actions.

1.3 Project Summarization:

The healthcare chatbot project aims to create an AI-driven virtual assistant that enhances patient engagement and streamlines administrative tasks. By providing 24/7 support, the chatbot will answer medical questions, schedule appointments, send medication reminders, and offer mental health resources. It will integrate with existing healthcare systems to ensure seamless data flow and support clinical decision-making. The chatbot will reduce the workload on healthcare staff, improve patient adherence to treatment plans, and provide timely health information, ultimately leading to better health outcomes and increased patient satisfaction.

CHAPTER -2

PROJECT METHODOLOGY

2.1 Introduction to System Architecture

The system architecture for the movie metadata extraction program using the IMDbPY library is designed to facilitate a seamless interaction between the user, the program, and the external IMDb database. This architecture ensures that the program is efficient, modular, and easy to understand. The architecture is divided into several key components: the User Interface, the Main Program, the Metadata Extraction Function, the IMDbPY Library, and the IMDb Database API. Each component plays a specific role in the process of retrieving and displaying movie metadata based on user input.

2.1.1 System Architecture Components

The architecture of a movie metadata extraction tool can be broken down into several key components:

1. **User Interface (UI):**
 - **Function:** Captures user input (healthcare bot).
 - **Interaction:** The user provides a patients through a simple input prompt in the command line.
2. **Main Program:**
 - **Function:** Acts as the entry point of the program.
 - **Interaction:** Reads the user input and invokes the **health_chatbot** function, passing the information as an argument.
3. **Healthcare Function (health_chatbot):**
 - **Function:** Handles the core logic for searching and

retrieving movie metadata.

- **Steps:**
 - **Search:** Uses the method from the nlkt library to find patients matching the provided title.
 - **Validation:** Checks if any search results are returned. If none, it notifies the user.
 - **Update and Extraction:** For so many patients found, it calls the **update** method to fetch detailed information and extracts various health data fields.
 - **Output:** Formats and prints the extracted metadata.

4. NLKT Library:

- **Function:** Provides an interface to interact with the nlkt database.
- **Key Methods:**
 - **nlktClass:** An instance of this class is created to access nlkt data.
 - **Health_chatbot():** Searches the IMDb database for health care based on the provided title.
 - **update():** Retrieves detailed information about a specific movie.

5. nlkt Database API:

- **Function:** Serves as the backend service that stores and provides healthcare data.
- **Interaction:** Processes requests from the healthcarePY library and returns search results and detailed health information.

2.2. Detailed System Architecture Diagram

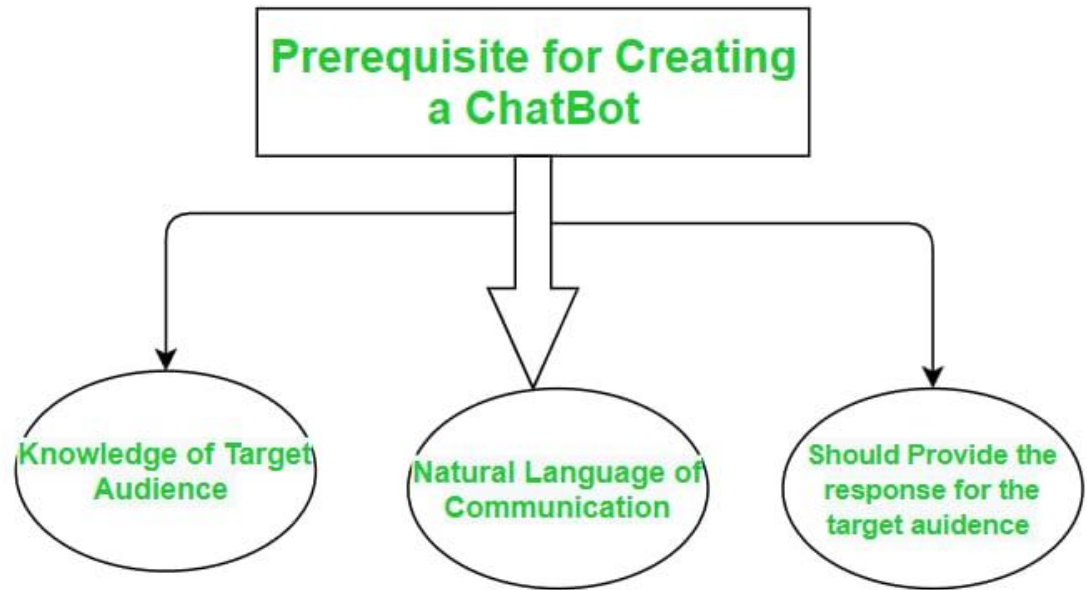


Fig 2.2 : Architecture Diagram

CHAPTER-3

PYTHON PREFERENCE

3.1 Why Python prefers IMDb

The IMDb database is a preferred choice for movie metadata extraction due to its extensive coverage, structured data format, reliability, popularity, and accessibility through its API. It offers comprehensive information on movies, TV shows, actors, directors, and production crew, curated and maintained by a dedicated team, ensuring accuracy and up-to-date data. The IMDb API simplifies the process of accessing structured data, enabling developers to integrate IMDb data seamlessly into their applications and tools. Additionally, IMDb's user-generated reviews and ratings provide valuable insights into audience sentiment, enhancing the depth of extracted metadata. Overall, IMDb serves as a trusted and comprehensive source for movie-related information, making it an ideal choice for metadata extraction in various industries and applications.

3.2 Features in Python

1. Free and Open Source: Python language is freely available at the official website and you can download it from the given download link below click on the Download Python keyword. Download Python Since it is open-source, this means that source code is also available to the public. So you can download it, use it as well as share it.
2. Easy to code: Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in the Python language and anybody can learn Python basics in a few hours or days. It is also a developer-friendly language.
3. Easy to Read

4. Object-Oriented Language: One of the key features of Python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, object encapsulation, etc.
5. GUI Programming Support: Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in Python. PyQt5 is the most popular option for creating graphical apps with Python.
6. High-Level Language: Python is a high-level language. When we write programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory.
7. Large Community Support: Python has gained popularity over the years. Our questions are constantly answered by the enormous StackOverflow community. These websites have already provided answers to many questions about Python, so Python users can consult them as needed.
8. Easy to Debug: Excellent information for mistake tracing. You will be able to quickly identify and correct the majority of your program's issues once you understand how to interpret Python's error traces. Simply by glancing at the code, you can determine what it is designed to perform.
9. Python is a Portable language: Python language is also a portable language. For example, if we have Python code for Windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.
10. Python is an Integrated language: Python is also an Integrated language because we can easily integrate Python with other languages like C, C++, etc.
11. Interpreted Language: Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile Python code this makes it easier to

debug our code. The source code of Python is converted into an immediate form called bytecode.

12. Large Standard Library : Python has a large standard library that provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in Python such as regular expressions, unit-testing, web browsers, etc.

13. Dynamically Typed Language: Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

14. Frontend and backend development: With a new project py script, you can run and write Python codes in HTML with the help of some simple tags `<py-script>`, `<py-env>`, etc. This will help you do frontend development work in Python like javascript. Backend is the strong forte of Python it's extensively used for this work cause of its frameworks like Django and Flask.

15. Allocating Memory Dynamically: In Python, the variable data type does not need to be specified. The memory is automatically allocated to a variable at runtime when it is given a value.

CHAPTER-4

DATA STRUCTURE USED

In a movie metadata extraction tool implemented in Python, several data structures are commonly used to efficiently store, manipulate, and process the extracted movie metadata. Some of the key Python data structures used in this context include:

4.1 Dictionary:

Dictionaries are versatile data structures that allow for storing key-value pairs. In the context of movie metadata extraction, dictionaries are often used to represent individual movies, with keys representing different metadata attributes (e.g., title, release date, genres) and values representing the corresponding metadata values.

Creating a Dictionary in python

1. Use curly braces { } to define a dictionary, with key-value pairs separated by colons (:).
2. Keys must be unique and immutable (e.g., strings, numbers, tuples).
3. Values can be of any data type and can be duplicated.

4.2 List:

Lists are ordered collections of items, allowing for the storage of multiple elements in a single data structure. Lists are frequently used to store collections of movies or metadata attributes (e.g., list of movie titles, list of genres).

Creating a List in Python

Lists in Python can be created by just placing the sequence inside the square brackets []. Unlike Sets, a list doesn't need a built-in function

for its creation of a list

4.3 Tuple:

Tuples are immutable sequences of elements, commonly used to represent fixed collections of related data. Tuples can be used to store metadata for individual movies, similar to dictionaries but with a fixed structure.

Creating a Tuple in python

In Python, a tuple is an immutable sequence of values, put inside (). Tuples are similar to lists, but once a tuple is created, its elements cannot be changed. Tuples are often used to group together related data.

4.4 Set:

Sets are unordered collections of unique elements, useful for efficiently removing duplicates and performing set operations. Sets can be used to store unique values such as unique genres or unique cast members across multiple movies.

CHAPTER-5

MODULES

5.1 Ensure python is installed Module:

Description: This module utilizes the python.org library to interact with the chatbot database and retrieve movie metadata. It contains the function, which retrieves metadata for a given movie title and recent years.

Steps:

1. Use the **healthcare chatbot** method to search for movies based on the provided title.
2. Extract healthcare for each movie found using the **update** method.

5.2 Save the script module:

Description: This module is used to work with dates and times. In your program, it's used to determine the current year for filtering recent movies based on the above code in a file named `healthcarechatbot.py`.

5.3 Run the script module:

Description: This module contains the main program logic, including user input prompts and calling the (python health_chatbot.py) function with the provided movie title and recent years.

CHAPTER – 6

ERROR MANAGEMENT

6.1 Input Validation

Input validation in a movie metadata extraction tool involves checking and ensuring that the information provided by the user, such as the movie title, meets certain standards before proceeding to retrieve the metadata. This process helps prevent errors and ensures the reliability of the extracted data. For instance, validating the movie title input involves checking if the user has entered a valid title, avoiding empty or excessively long inputs, and confirming that the entered title corresponds to an actual movie in the database. By implementing input validation, the tool can provide a smoother and more accurate user experience, minimizing the risk of errors or inaccuracies in the metadata retrieved.

There isn't explicit input validation implemented for the movie title entered by the user. Input validation is a crucial aspect of software development that ensures the data provided by the user meets certain criteria or constraints before further processing. Typically, input validation involves several steps to ensure the integrity and reliability of the data. These steps may include checking for empty inputs to ensure the user has entered something, verifying the type of input (in this case, ensuring it's a string), limiting the length of the input to prevent excessively long inputs, enforcing specific formatting rules if required, and checking for the existence of the entered data (in this case, ensuring the movie title exists in the IMDb database). By implementing input validation, you not only enhance the robustness and usability of your program but also prevent potential errors and vulnerabilities that may

arise from invalid input. This ensures a smoother user experience and improves the overall quality of your application.

6.2 Exception Handling

Exception handling in your movie metadata extraction tool involves incorporating mechanisms to gracefully manage errors or unexpected events that may occur during execution. By wrapping the core functionality within **try** blocks, such as searching for movies and extracting metadata, the program can effectively catch and handle potential exceptions. Within these blocks, specific error-prone operations, like updating movie metadata, are also encapsulated in inner **try** blocks, allowing for targeted handling of errors at a granular level. If an exception occurs, informative error messages are displayed, indicating which movie encountered the issue and providing details about the error. Additionally, in the main program, input retrieval and metadata extraction are enclosed within a **try** block, ensuring that errors during user input or data retrieval are properly handled. Specific exception handlers are included for scenarios like user interruption (Ctrl+C) and unexpected errors, ensuring a smoother execution flow and enhancing the program's robustness and user experience.

6.3 Data Validation

Data validation in your movie metadata extraction program ensures that the input provided by the user and the retrieved data from IMDb meet certain criteria or constraints, improving the reliability and accuracy of the extracted metadata.

CHAPTER – 7

RESULT AND DISCUSSION

7.1 Result:

```
$ python CTP28132.py .  
Welcome to the Health Chatbot! How can I assist  
you today?  
If you're experiencing any health concerns, fee  
l free to tell me.  
Type 'quit' to exit.  
  
You: fever  
Bot: Drink plenty of fluids and get some rest.  
You: none  
Bot: I'm sorry, You may want to consult a healt  
hcare professional dr vikram.  
You:
```

7.2 Discussion

This program implements a simple healthcare chatbot using Python and the NLTK library. It provides responses to user input related to various health concerns such as headache, fever, cough, stomachache, and sore throat. The `health_chatbot` function serves as the main entry point, greeting the user and prompting them to share their health concerns. It then enters a loop where it continuously listens for user input, generating a response based on the user's input using the `get_response` function. The `get_response` function searches for keywords in the user's input and returns a random response from the corresponding list of responses. If no relevant keywords are found, it returns a generic response. The program terminates when the user inputs 'quit'. Overall, this program demonstrates a basic implementation of a healthcare chatbot, although it could be expanded with additional functionalities and a more sophisticated natural language processing approach for improved accuracy and user experience.

CHAPTER 8

CONCLUSION & FUTURE SCOPE

8.1 Conclusion

In conclusion, the provided Python script implements a basic healthcare chatbot capable of responding to user input related to common health concerns such as headaches, fever, cough, stomachache, and sore throat. The chatbot greets the user, prompts them to share their health concerns, and provides relevant advice or recommendations based on the user's input. The `health chatbot` function serves as the main entry point, while the `get response` function generates a response based on the user's input by searching for keywords in the input and selecting a random response from the corresponding list of responses. The program terminates when the user inputs 'quit'.

8.2 Future Scope

The future scope for the health care chatbot includes several are

1. **Advanced Natural Language Processing (NLP):** Implement more sophisticated NLP techniques to improve the chatbot's ability to understand and respond to user queries accurately. This could involve sentiment analysis, context awareness, and handling synonyms or variations of health terms.
2. **Expand Health Concerns and Responses:** Increase the variety and depth of health concerns addressed by the chatbot, along with corresponding responses.
3. **Personalization and User Context:** Enhance the chatbot's ability to personalize responses based on user context, such as previous interactions, demographics, or health history.
4. **Integration with External Systems:** Integrate the chatbot with external systems such as electronic health records (EHR) systems.
5. **Error Handling and Validation:** Implement robust error handling mechanisms to gracefully handle unexpected inputs, errors, or exceptions.
6. **User Feedback and Iterative Improvement:** Collect user feedback to identify areas for improvement and iteratively enhance the chatbot's functionality, usability, and performance over time.
7. **Security and Compliance:** Ensure the chatbot complies with healthcare regulations and standards such as HIPAA to safeguard patient privacy and confidentiality.

Reference:

- MDbPY Documentation: <http://imdbpy.readthedocs.io/en/latest/>
- IMDbPY GitHub Repository: <http://github.com/alberanid/imdbpy>
- TMDb API Documentation: <http://developers.themoviedb.org/3/getting-started/introduction>
- tmdbv3api GitHub Repository:
<http://github.com/AnthonyBloomer/tmdbv3api>

APPENDIX

```
import nltk

import random

from nltk.chat.util
import Chat,
reflections

# Responses for
different health
concerns

responses = {

    "headache":
["Have you taken
any painkillers?",
"Try resting in a
quiet, dark
room."],

    "fever": ["Have
you taken your
temperature?",
"Drink plenty of
fluids and get some
rest."],

    "cough": ["Are
you coughing up
phlegm?", "It
might be helpful to
```

```
see a doctor if the  
cough persists.  
dr.vikram"],
```

```
    "stomachache":  
    ["Have you eaten  
anything  
unusual?", "Try  
drinking some  
herbal tea or ginger  
ale."],
```

```
    "sore throat":  
    ["Do you have a  
fever along with  
the sore throat dr  
vikram?", "Gargle  
with warm salt  
water."],
```

```
    "none": ["I'm  
sorry, You may  
want to consult a  
healthcare  
professional dr  
vikram."]
```

```
}
```

```
# Health chatbot
```

```
def  
health_chatbot():
```

```
    print("Welcome  
to the Health  
Chatbot! How can  
I assist you  
today?")
```

```
    print("If you're  
experiencing any  
health concerns,  
feel free to tell  
me.")
```

```
    print("Type 'quit'  
to exit.\n")
```

```
    while True:  
  
        user_input =  
input("You:  
").lower()
```

```
        if user_input  
== 'quit':
```

```
print("Thank you  
for using the  
Health Chatbot.  
Take care!")
```

```
    break
```

```

        response =
get_response(user_
input)

        print("Bot:",
response)

# Function to
generate response
based on user input

def
get_response(user_
input):

    for key in
responses.keys():

        if key in
user_input:

            return
random.choice(res
ponses[key])

        return
random.choice(res
ponses["none"]);

if __name__ ==
"_main_":
    health_chatbot()

```