PROJECT REPORT

TOPIC-HEALTHCARE CHATBOT

NAME	SARISHA LAMBOR
REGISTRATION NO	25BHI10077

TABLE OF CONTENTS

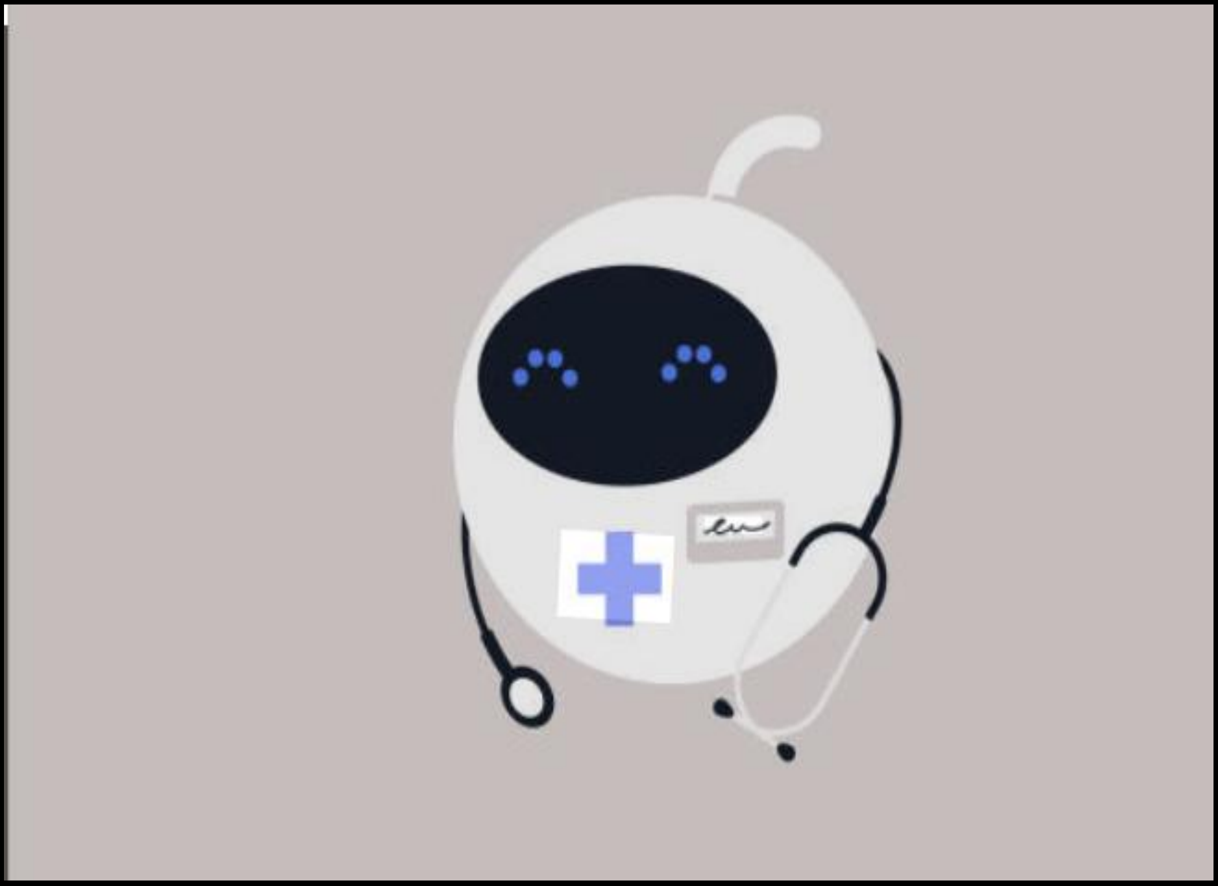
CONTENT	PAGE NO
INTRODUCTION	3
PROBLEM DESCRIPTION	5
SOFTWARE REQUIREMENTS	6
PROJECT DESIGN AND ARCHITECTURE	7
SCOPE OF THE PROJECT	10
INPUT AND OUTPUT RESULTS	11
CONCLUSION AND FUTURE ENHANCEMENTS	13

INTRODUCTION

ABOUT THE PROJECT:

The title of my project is "HEALTHCARE CHATBOT". Healthcare is rapidly moving toward patient -centred care, and technology has been essential in developing the quality of personalised healthcare. The idea is to make a medical chatbot using python which will supply basic details about the disease before consulting a doctor. Chatbot was developed with an aim of enhancing patient participation, provisioning relevant messages in time, as well building linkage between doctors and patients. It will help to reduce the healthcare costs and will help to enhance the patient's health.it will provide a personalized diagnosis supported their symptoms. My python based healthcare chatbot system connects patient with the chatbot twill

HELP THEM GIVE CORRECT ANSWERS AND PRECAUTIONS TO THEIR QUESTIONS.



Problem description

An enormous need for easily accessible and effective patient assistance systems exists in the healthcare industry.

This IS A RULE- BASED chatbot designed to chat or respond to the user regarding any health issue. This healthcare is programmed using python language. chatbots can be programmed to respond the same way each time and to repond differently to messages containing certain keywords. This healthcare chatbot will help to provide general information. It helps patients by guiding what exactly he/she is looking for. An enormous need for easily accessible and effective patient assistance systems exists in the healthcare industry.

SOFTWARE REQUIREMENTS

LANGUAGE: PYTHON

EDITOR: VISUAL STUDIO CODE

APPLICATION: CHROME

OS: WINDOWS 11

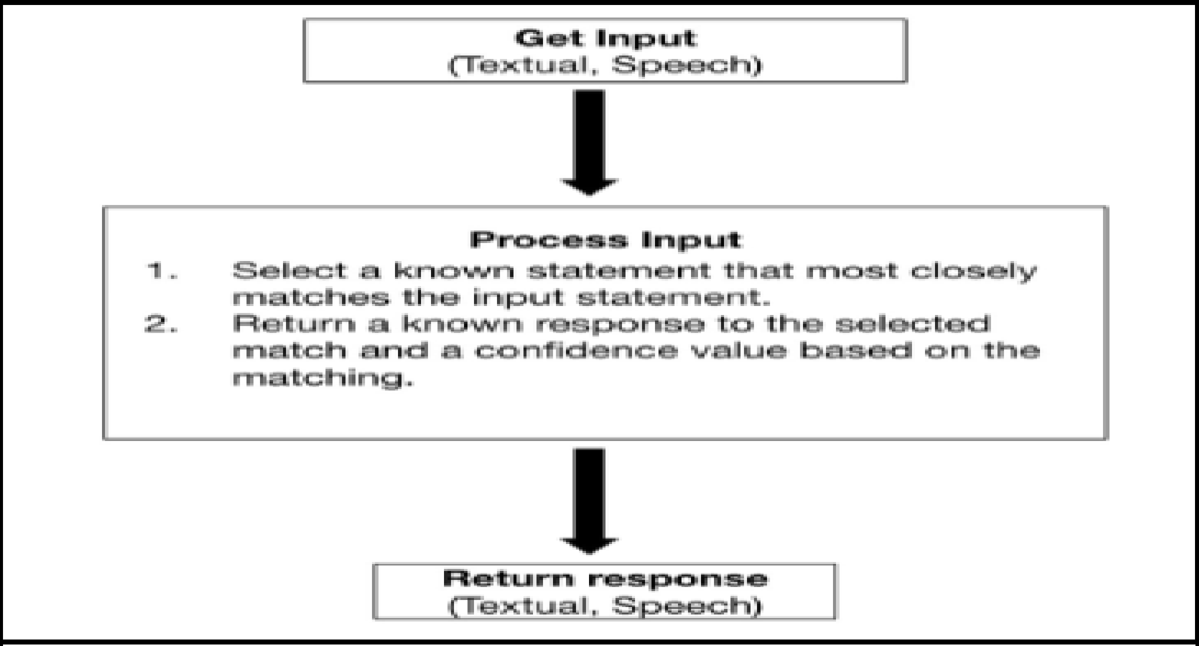
DATASETS

- PYTHON IS A DYNAMIC LANGUAGE AND REDUCES COMPLEXITY MEANING YOU CAN IMPLEMENT FUNCTIONALLY WITH LESS CODE. PYTHON CODE CAN BE RUN ON ANY PLATFORM WITHOUT WASTING TIME ON SPECIFIC CONFIGERATIONS.
- VISUAL STUDIO CODE IS FREE SOURCE CODE EDITOR MADE BY MICROSOFT. FEATURES ARE DEBUGGING AND SYNTAX HIGHLIGHTING.
- FOR THIS HEALTHCARE CHATBOT , I HAVE USED 2-3 DATASETS WHICH ARE SYMPTOM DESCRIPTION, SYMPTOM PRECAUTION AND DOCTOR CONSULTATION.

PROJECT DESIGN AND ARCITECTURE

To provide a dependable and efficient healthcare chatbot solution, correct input and keywords are essentially given top priority in the entire design and architecture. A rule based chatbot is a conversational tool that uses a predefined set of rules, and a decision tree structure to respond to user inputs.it operates on an if-then logic recognizing only specific keywords to give a programmed response these bots are ideal for handling simple task but lack flexibility.

FLOWCHART:


Get Input
(Textual, Speech)
Process Input
1. Select a known statement that most closely matches the input statement.
2. Return a known response to the selected match and a confidence value based on the matching.
Return response
(Textual, Speech)

PYTHON CONCEPTS APPLIED:

Variables and Data Types:

- Used to store user input, predefined responses, patterns, and internal state.
- Data Types Used: Strings (for user messages and responses), Lists (to hold multiple potential responses for a rule or to store rules themselves), and Dictionaries .

Functions (def):

- Used to modularize the cod. Functions are created for specific tasks like pre process input(text),

  Find matching rule, and generate response (rule).

Conditional Statements (if/elif/else):

- Crucial for the rule-based logic. They determine *which* rule matches the user's input and *what* response should be returned.

Loops (for/while):

- for loops are used to iterate through the list of defined rules and patterns to check for a match against the user's input.

- A while loop is used to keep the chatbot running continuously, accepting and processing input until the user types an exit command.

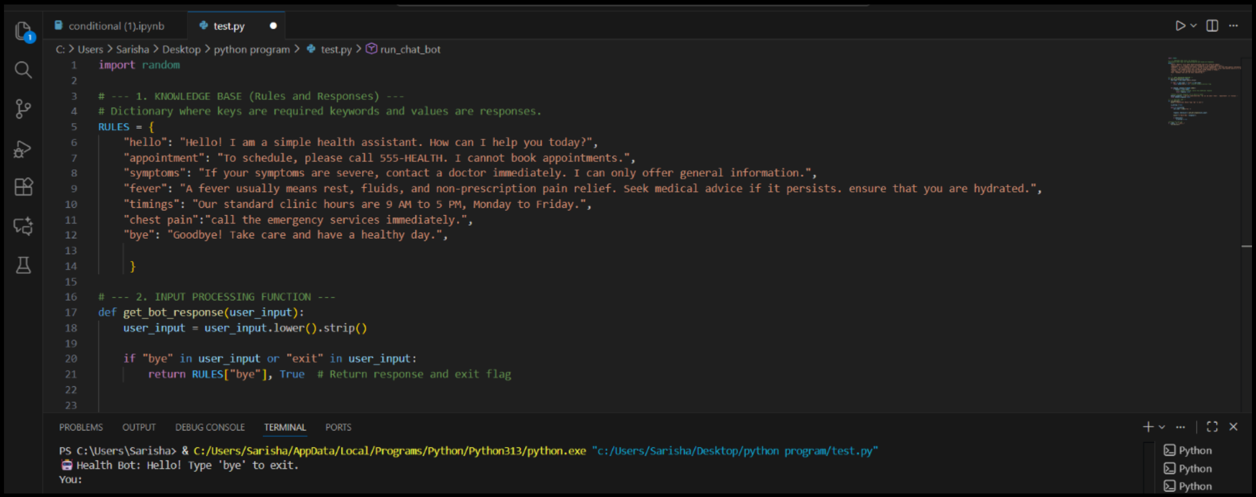
SCOPE OF THE PROJECT

THERE ARE 61% OF CONSUMERS PREFER TO COMMUNICATE WITH CHATBOTS FOR EFFECTIVE INTERACTION.NOWADAYS CHATBOTS ARE BECOMING SO TRENDY. DOCTORS CAN USE THESE CHATBOTS TO IMPROVE THE QUALITY OF HEALTHCARE.THE HEALTHCARE CHATBOT SYSTEM WILL HELP TO PROVIDE HEALTHCARE SUPPORT AND INFORMATION. It is developed with the aim to provide helpful information instantly. This initiative bears relevance as it successfully bridges the gulf between sophisticated technology and healthcare, The project& goal is to encourage patients to take a more educated

INPUT AND OUTPUT RESULTS

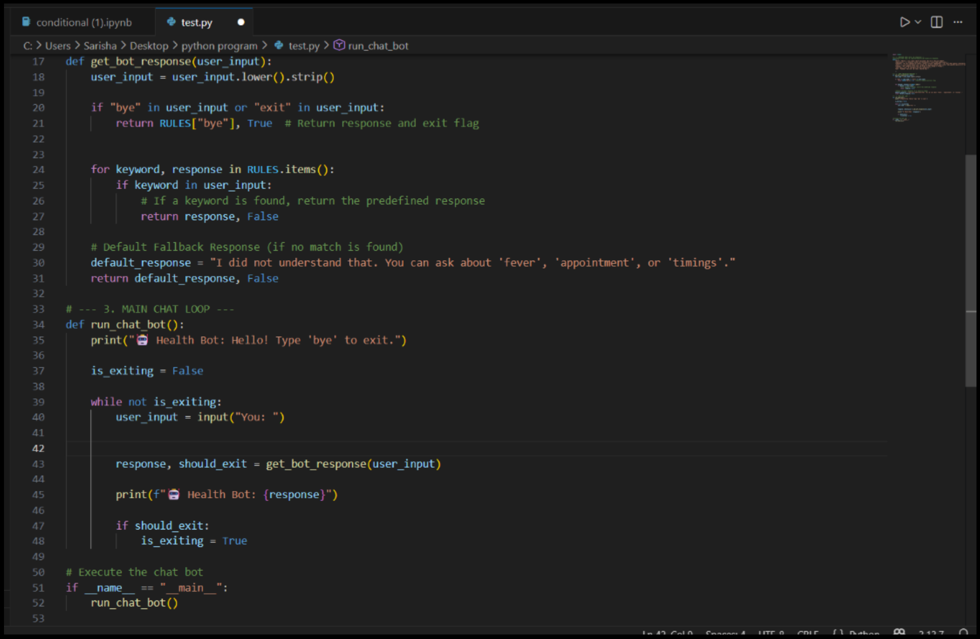
USER INTERFACE:

RULE – BASED CHATBOT INTERACTION:



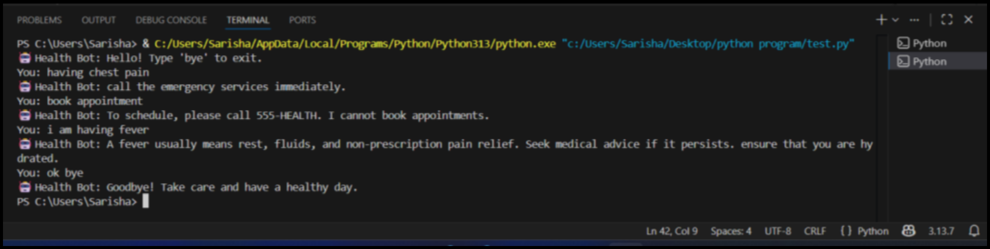
```python
import random

# --- 1. KNOWLEDGE BASE (Rules and Responses) ---
# Dictionary where keys are required keywords and values are responses.
RULES = {
    "hello": "Hello! I am a simple health assistant. How can I help you today?",
    "appointment": "To schedule, please call 555-HEALTH. I cannot book appointments.",
    "symptoms": "If your symptoms are severe, contact a doctor immediately. I can only offer general information.",
    "fever": "A fever usually means rest, fluids, and non-prescription pain relief. Seek medical advice if it persists. ensure that you are hydrated.",
    "timings": "Our standard clinic hours are 9 AM to 5 PM, Monday to Friday.",
    "chest pain":"call the emergency services immediately.",
    "bye": "Goodbye! Take care and have a healthy day.",

    }

# --- 2. INPUT PROCESSING FUNCTION ---
def get_bot_response(user_input):
    user_input = user_input.lower().strip()

    if "bye" in user_input or "exit" in user_input:
        return RULES["bye"], True  # Return response and exit flag

```

```
PS C:\Users\Sarisha> & C:/Users/Sarisha/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Sarisha/Desktop/python program/test.py"
🤖 Health Bot: Hello! Type 'bye' to exit.
You:
```


conditional (1).ipynb
test.py
C: > Users > Sarisha > Desktop > python program > test.py > run_chat_bot

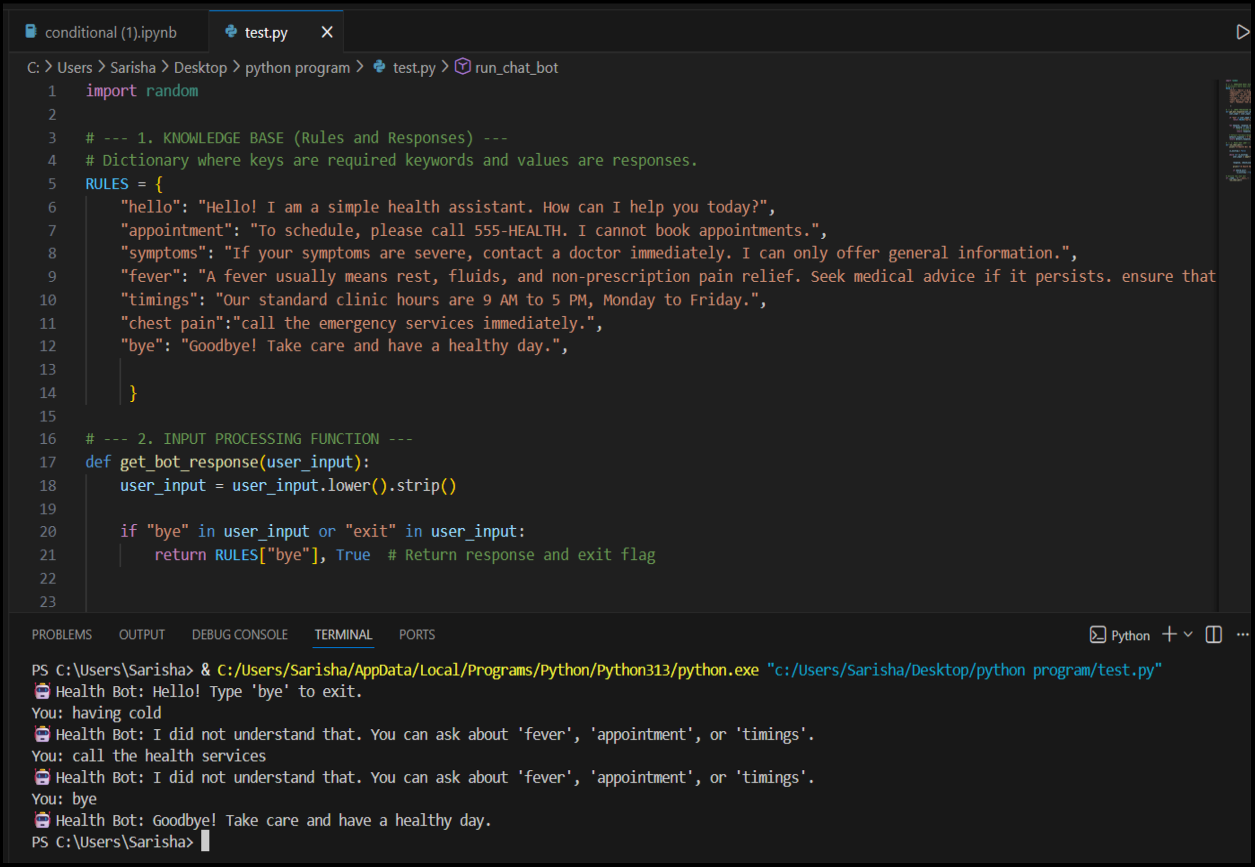
```python
def get_bot_response(user_input):
    user_input = user_input.lower().strip()

    if "bye" in user_input or "exit" in user_input:
        return RULES["bye"], True  # Return response and exit flag


    for keyword, response in RULES.items():
        if keyword in user_input:
            # If a keyword is found, return the predefined response
            return response, False

    # Default Fallback Response (if no match is found)
    default_response = "I did not understand that. You can ask about 'fever', 'appointment', or 'timings'."
    return default_response, False

# --- 3. MAIN CHAT LOOP ---
def run_chat_bot():
    print("🤖 Health Bot: Hello! Type 'bye' to exit.")

    is_exiting = False

    while not is_exiting:
        user_input = input("You: ")


        response, should_exit = get_bot_response(user_input)

        print(f"🤖 Health Bot: {response}")

        if should_exit:
            is_exiting = True

# Execute the chat bot
if __name__ == "__main__":
    run_chat_bot()

```

SERIES OF QUESTIONING:



```
PS C:\Users\Sarisha> & C:/Users/Sarisha/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Sarisha/Desktop/python program/test.py"
🤖 Health Bot: Hello! Type 'bye' to exit.
You: having chest pain
🤖 Health Bot: call the emergency services immediately.
You: book appointment
🤖 Health Bot: To schedule, please call 555-HEALTH. I cannot book appointments.
You: i am having fever
🤖 Health Bot: A fever usually means rest, fluids, and non-prescription pain relief. Seek medical advice if it persists. ensure that you are hy
drated.
You: ok bye
🤖 Health Bot: Goodbye! Take care and have a healthy day.
PS C:\Users\Sarisha> █
```

Analysis and termination:



```python
import random

# --- 1. KNOWLEDGE BASE (Rules and Responses) ---
# Dictionary where keys are required keywords and values are responses.
RULES = {
    "hello": "Hello! I am a simple health assistant. How can I help you today?",
    "appointment": "To schedule, please call 555-HEALTH. I cannot book appointments.",
    "symptoms": "If your symptoms are severe, contact a doctor immediately. I can only offer general information.",
    "fever": "A fever usually means rest, fluids, and non-prescription pain relief. Seek medical advice if it persists. ensure that
    "timings": "Our standard clinic hours are 9 AM to 5 PM, Monday to Friday.",
    "chest pain":"call the emergency services immediately.",
    "bye": "Goodbye! Take care and have a healthy day.",

     }

# --- 2. INPUT PROCESSING FUNCTION ---
def get_bot_response(user_input):
    user_input = user_input.lower().strip()

    if "bye" in user_input or "exit" in user_input:
        return RULES["bye"], True  # Return response and exit flag


```

```
PS C:\Users\Sarisha> & C:/Users/Sarisha/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Sarisha/Desktop/python program/test.py"
🤖 Health Bot: Hello! Type 'bye' to exit.
You: having cold
🤖 Health Bot: I did not understand that. You can ask about 'fever', 'appointment', or 'timings'.
You: call the health services
🤖 Health Bot: I did not understand that. You can ask about 'fever', 'appointment', or 'timings'.
You: bye
🤖 Health Bot: Goodbye! Take care and have a healthy day.
PS C:\Users\Sarisha> █
```

CONCLUSION AND FUTURE ENHANCEMENTS

IN THE FUTURE, I WOULD LIKE TO ADD SOME MORE FEATURES TO THIS RULE-BASED HEALTHCARE CHATBOT SO THAT IT CAN BE MORE INTERACTIVE WITH PEOPLE. I WOULD LIKE TO ADD MORE MODULES TO MAKE IT A FULLY FUNCTIONAL AI HEATHCARE CHATBOT.

I WILL MAKE THE CHATBOT SO DYNAMIC WITH ITS REPONSES THAT WILL GIVE USERS A HUMANIZED EXPERIENCE, SO THE USERS WILL FEEL AS THOUGH THEY ARE TALKING TO A REAL PERSON.Frequent upgrades are necessary for it to remain current and keep becoming better. These improvements not only show the system& dedication to improving virtual healthcare assistance, but they also guarantee privacy standards are followed and a user-centric design approach is maintained. All things considered, the healthcare chatbot is a promising advancement in virtual healthcare providing precise diagnosis, tailored advice, and a smooth user experience.

THANK YOU