Date:29.10.25

TASK:12

Implementation of Chatbot for Technical Support

CO4, CO5 S3

PROBLEM STATEMENT:

In many organizations, providing timely and effective technical support to users is challenging due to the large number of queries and limited human resources. Users often face delays in getting solutions to common technical problems. The problem is to develop an intelligent chatbot for technical support that can automatically understand user queries, provide instant and accurate responses, and assist users in troubleshooting issues related to software, hardware, or services. This system aims to improve response time, reduce workload on support staff, and enhance overall user satisfaction.

AIM:

To develop an intelligent chatbot system that provides automated technical support by understanding user queries and delivering instant, accurate solutions**.**

OBJECTIVE:

1. To design a chatbot capable of understanding and responding to user queries related to technical issues.

2. To use Natural Language Processing (NLP) techniques for interpreting user input.

3. To provide instant and accurate solutions to common technical problems.

4. To reduce the workload on human technical support staff.

5. To improve user experience and response time in technical assistance.

DESCRIPTION:

The Chatbot for Technical Support is an intelligent virtual assistant designed to provide instant help and guidance to users facing technical issues. It uses Natural Language Processing (NLP) to understand user queries and respond with accurate, predefined or AI-generated solutions. The chatbot can assist with common problems such as internet issues, software installation, password resets, and printer errors. By automating routine support tasks, it reduces the workload on human support staff, improves response time, and ensures 24/7 availability of technical assistance. This project demonstrates the application of AI in enhancing customer service efficiency and user satisfaction**.**

ALGORITHM:

1. Start

2. Input User Query:

The user enters a technical question or issue into the chatbot interface.

3. Preprocess Query:

Convert text to lowercase

Remove unnecessary characters and stop words

Tokenize the sentence for analysis

4. Intent Recognition:

Use Natural Language Processing (NLP) to identify the intent or category of the user's query (e.g., software issue, password reset, connection error).

5. Response Generation:

If the query matches a known issue, fetch the predefined solution from the knowledge base.

If the query is new or unclear, use an AI model to generate an appropriate response.

6. Output Response:

Display the solution or response to the user through the chatbot interface.

7. Feedback Collection (Optional):

Ask the user if the solution was helpful and store feedback for improvement.

8. End

[9:38 pm, 27/10/2025] Harshith Vtu: # Chatbot for Technical Support

PROGRAM:

```
import random
# Step 1: Predefined responses
responses = {
    "internet": ["Please check your internet connection.",
                 "Try restarting your router.",
                 "Ensure your Wi-Fi is turned on."]
  "password": ["You can reset your password using the 'Forgot Password' option.",
                 "Make sure Caps Lock is off while typing your password."]
    "software": ["Please reinstall the software and restart your system.",
                 "Ensure the software is updated to the latest version."],
    "printer": ["Check if the printer is connected and turned on.",
                "Reinstall the printer drivers and try again."],

    "default": ["I'm sorry, I didn't understand your query. Could you please rephrase it?"]
}
# Step 2: Chatbot function
def chatbot():
    print("TechBot: Hello! I’m your technical support assistant. Type 'bye' to exit.")
    while True:
        user_input = input("You: ").lower()
        if user_input == "bye":
            print("TechBot: Goodbye! Have a great day.")
            break

        # Step 3: Check for keywords in user input
        found = False
        for key in responses.keys():
            if key in user_input:
                print("TechBot:", random.choice(responses[key]))
                found = True
                break

        # Step 4: Default response if no match found
        if not found:
            print ("TechBot:", random.choice(responses["default"]))

# Step 5: Run chatbot
chatbot ()
```

OUTPUT:

```
[9:38 pm, 27/10/2025] Harshith Vtu: TechBot: Hello! I’m your technical
support assistant. Type 'bye' to exit.

You: My internet is not working

TechBot: Try restarting your router.

You: My printer is not printing

TechBot: Check if the printer is connected and turned on.

You: bye

TechBot: Goodbye! Have a great day.
```

CONCLUSION:

The Chatbot for Technical Support provides an efficient and automated way to handle user queries and technical issues. It uses simple NLP techniques to understand user input and deliver instant, accurate responses. This system helps reduce response time, minimizes the workload on human support teams, and enhances user satisfaction by providing 24/7 assistance. It demonstrates how AI and automation can improve the efficiency and accessibility of technical support services.