

Implementation of Chatbot For Technical Support**PROBLEM STATEMENT:**

Develop a simple chatbot system that can provide automated responses to basic technical support queries. The chatbot should recognize keywords in user input (like “internet”, “login”, or “password”) and reply with predefined helpful answers. This helps reduce human workload in customer support by handling frequent questions automatically.

AIM:

To build a simple Python chatbot that assists users by responding to common technical support queries using simple rule-based keyword matching.

OBJECTIVE:

1. Create a chatbot that can process user input from the command line.
2. Identify the topic of the query using keyword matching.
3. Generate relevant predefined responses for basic technical issues.
4. Continue conversation until the user types “bye”.
5. Provide a simple foundation for extending into an AI-based chatbot later.

DESCRIPTION:

A chatbot is an AI-based program that interacts with users through text. In this task, we create a rule-based chatbot that matches user input with keywords stored in a dictionary and responds accordingly. If the chatbot doesn't recognize the input, it replies with a default message like "I'm not sure, please contact support." This basic version demonstrates how chatbot logic works without needing any external APIs or NLP libraries.

ALGORITHM:

1. Start the program.
2. Create a dictionary containing keywords and their responses.
3. Take user input in a loop.
4. Convert input to lowercase and check if any keyword matches.
5. If a keyword is found → print the corresponding response.
6. If no keyword is found → print a default response.
7. End the chat when the user types "bye".

PROGRAM:

```
# Task12_easy_chatbot.py

print("TechBot: Hello! I'm your technical support assistant. How can I help you today?")

print("(Type 'bye' to end the chat)\n")

while True:

    user_input = input("You: ").lower() # Convert input to lowercase for easier matching

    if user_input == "bye":

        print("TechBot: Thank you for chatting! Have a great day!\n")

        break
```

```
elif "internet" in user_input or "wifi" in user_input:

    print("TechBot: Please check your router connection and try restarting it.\n")

elif "password" in user_input:

    print("TechBot: You can reset your password by clicking on 'Forgot Password' on the login
page.\n")

elif "login" in user_input:

    print("TechBot: Ensure you are using the correct username and password.\n")

elif "email" in user_input:

    print("TechBot: Check if your email is entered correctly and your internet connection is
stable.\n")

elif "update" in user_input:

    print("TechBot: Please make sure your system is connected to the internet before checking for
updates.\n")

elif "error" in user_input:

    print("TechBot: Could you please specify the error message? I'll help you troubleshoot.\n")

elif "printer" in user_input:

    print("TechBot: Check if the printer is connected properly and has enough paper and ink.\n")

elif "slow" in user_input:

    print("TechBot: Try restarting your system or closing unused programs to improve speed.\n")

elif "support" in user_input:

    print("TechBot: You can contact our technical support team at support@example.com.\n")

else:

    print("TechBot: I'm sorry, I didn't understand that. Please rephrase or contact support.\n")
```

OUTPUT:

```
| PS C:\VTU24535> & C:/VTU24535/myenv/Scripts/Activate.ps1
● (myenv) PS C:\VTU24535> python task_12.py
TechBot: Hello! I'm your technical support assistant. How can I help you today?
(Type 'bye' to end the chat)

You: my internet is not working
TechBot: Please check your router connection and try restarting it.

You: my laptop is slow
TechBot: Try restarting your system or closing unused programs to improve speed.

You: i forgot my password
TechBot: You can reset your password by clicking on 'Forgot Password' on the login page.

You: bye
TechBot: Thank you for chatting! Have a great day!

○ (myenv) PS C:\VTU24535> █
```

CONCLUSION:

The Chatbot for Technical Support was developed using basic Python logic and keyword matching. It responded accurately to common technical queries and displayed clear, user-friendly outputs. The project met its objectives and was successfully executed.