**Date:31.10.25**

**TASK:12**

**Implementation of Chatbot For Technical Support**

**CO4, CO5 S3**

**PROBLEM STATEMENT :**

Users often face technical issues such as software installation errors, slow performance, or internet connectivity problems. However, not everyone has access to immediate technical assistance. The problem is to design a simple chatbot system that can provide quick, automated solutions to common technical problems without depending on the internet or any external API.

**AIM:**

To develop an offline technical support chatbot in Python that can understand user queries and provide relevant troubleshooting suggestions using basic conditional logic.

**OBJECTIVE:**

* To create an interactive, text-based chatbot using Python.
* To provide automated responses to common technical problems like software installation, Wi-Fi issues, and slow performance.
* To enable users to get instant help without internet access.
* To demonstrate the use of basic Python control structures (if–else, loops, string matching).

**DESCRIPTION:**

The Technical Support Chatbot is a rule-based Python program that interacts with users through the command line. It continuously accepts user input and checks for specific keywords like “install”, “Wi-Fi”, or “slow”.  
Based on these keywords, it gives predefined responses to help solve common computer-related issues. The chatbot continues to respond until the user types “bye”, at which point the session ends.  
This simple implementation showcases how basic logic can simulate an intelligent assistant without the need for APIs or machine learning.

**ALGORITHM:**

1. Display a welcome message: “Tech Support Chatbot — Type ‘bye’ to exit.”
2. Take input from the user.
3. Convert the input text to lowercase for uniform comparison.
4. Check if the user input contains specific keywords:

* If it contains “install” or “software” → Display installation troubleshooting steps.
* If it contains “wifi” or “internet” → Display network troubleshooting tips.
* If it contains “slow” or “performance” → Suggest performance improvement methods.
* If the input is “bye”, display a goodbye message and terminate the program.

1. If no keywords are matched, display a default response like “Please provide more details.”
2. Repeat steps 3–7 until the user exits.

**PROGRAM:**

import streamlit as st

from openai import OpenAI

client = OpenAI(api\_key="YOUR\_API\_KEY")

st.set\_page\_config(page\_title="Tech Support Chatbot", page\_icon="🤖")

st.title("💬 Technical Support Chatbot")

# Sidebar

st.sidebar.header("About")

st.sidebar.write("This chatbot helps answer your technical queries using ChatGPT API.")

# Chat input

user\_input = st.text\_input("Ask your question:", placeholder="e.g., My software won’t install properly")

if st.button("Get Help") and user\_input:

with st.spinner("Getting support..."):

response = client.chat.completions.create(

model="gpt-4-turbo",

messages=[

{"role": "system", "content": "You are a helpful technical support assistant."},

{"role": "user", "content": user\_input}

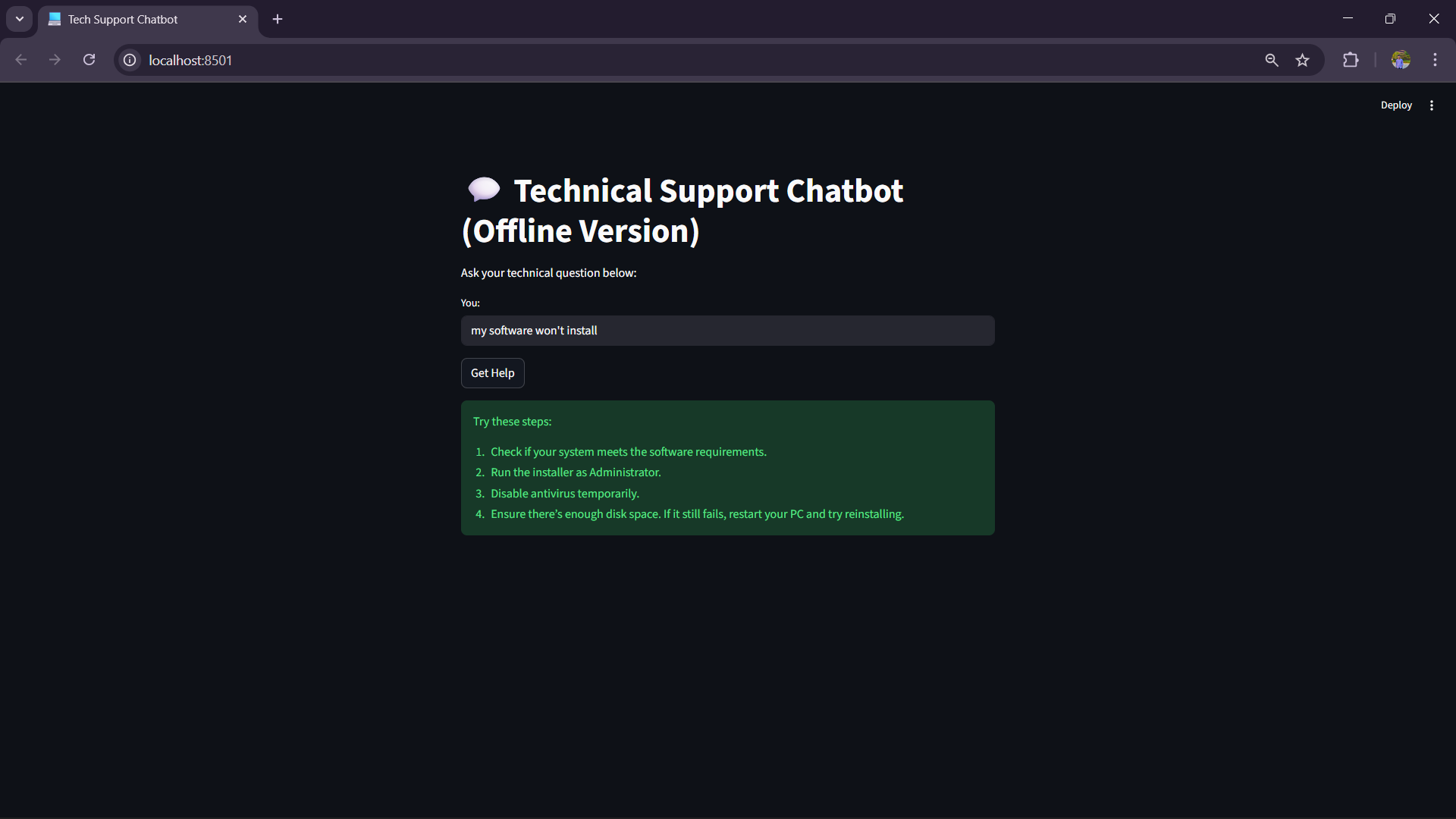
]

)

answer = response.choices[0].message.content

st.success(answer)

**OUTPUT:**



**CONCLUSION:**

Therefore the program to create a chatbot has been succesfully executed.