**PROBLEM STATEMENT : Create a chatbot using python**

**ABSTRACT:**

* This Python chatbot project explores the development of a conversational agent using Natural Language Processing (NLP) techniques.
* The chatbot is designed to engage in human-like conversations with users, offering assistance, answering questions, and providing information on a variety of topics.
* The code leverages popular NLP libraries, such as NLTK

and spacy, along with machine learning models like GPT-3, to enhance the chatbot's language understanding and Generation capabilities.

* This abstract provides an overview of the key components and methods used in creating the chatbot, including data preprocessing, model training, and integration with user interfaces.
* The resulting chatbot demonstrates the potential of NLP and AI technologies in building intelligent and interactive applications.

# DESIGN THINKING:

* Here's a simple Python code for a chatbot using a dictionary-based approach to provide predefined responses to specific user inputs.
* In this code, we define a dictionary called responses that

maps user inputs to predefined responses. The get response function checks if the user input exists in the dictionary and returns the corresponding response.

* The main loop allows the user to input messages, and the

chatbot responds with the appropriate message based on the dictionary or a default "I don't understand that" message. You can exit the chatbot by typing "exit".

**PROGRAM:**

**import nltk**

**import random**

**from nltk.chat.util import Chat, reflections**

**# Define pairs of patterns and responses for the chatbot**

**pairs = [**

**["hello|hi|hey", ["Hello!", "Hi there!", "Hey!"]],**

**["how are you", ["I'm good, thanks. How about you?", "I'm just a bot, but I'm here to help."]],**

**["what is your name?", ["I'm a chatbot. You can call me ChatGPT.", "I'm ChatGPT, your friendly chatbot."]],**

**["bye|goodbye", ["Goodbye!", "Have a great day!"]],**

**["(.\*)(rain)(.\*)", ["Yes, I love the rain!", "Rain is so refreshing."]],**

**["(.\*) weather (.\*)", ["The weather is always changing.", "I don't have access to real-time weather data."]],**

**]**

**# Create a chatbot**

**chatbot = Chat(pairs, reflections)**

**# Function to start the chat**

**def start\_chat():**

**print("Hello! I'm your chatbot. You can start the conversation. Type 'quit' to exit.")**

**while True:**

**user\_input = input("You: ")**

**if user\_input.lower() == "quit":**

**print("Chatbot: Goodbye!")**

**break**

**response = chatbot.respond(user\_input)**

**print("Chatbot:", response)**

**# Start the chat**

**start\_chat()**

**PROCESS;**

**1:IMPORT LIBRARIES:**

Impot the necessary libiaries at the beginning of your python script .in this code ,we import nltk,random and the chat class from ntlk.chat.utill.

**2: DEFINE PATTERNS AND RESPONSE:**

The pairs list contains patterns and their corresponding responses you can customize this list to create specific interactions .each element in the list has the format :[“pattern[‘response1]”, “response2”,…]]

**3:CREATE THE CHATBOT:**

Initialize the chatbot by creating a chat instance with the defind patterns and reflections (provided by nltk.chat.utill)

. Set up a function , start\_chat(),to facilitate the conservation

**4:START THE CONVERSATION :**

**.** In the start\_chat() function ,initiate a conversation loop that waits for users input.

**.** The user can type messages , and the chat bot will respond based on the defined patterns

**.** The loop continues until the user types “quit”, at which point the chatbot says “Good bye!” and the conversation ends.

**5: RUNNING THE CODE:**

**.** After you saved the code in a python file (eg., simple\_chatbot.py),you can run it by executing python simple \_chatbot .py in your terminal or command prompt.

**.** the chatbotwill greet the user and respond to their inputs based on the patterns definded in the code .

**.** To exit the conversation ,the user can type “quit”.

**6:customization:**

You can customize the patterns and responses in the pairs list to create chatbot that responds to specific inputs ans behaves accoding to your requirements . you can also expand the chatbot’s capabilities by adding more patterns and responses .

This is a simple and rule-based chatbot .for more advanced chatbots ,might explore machine learning techniques, such as using deep learning models and frameworks like TensorFlow and pytorch and natural language processing tasks