**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 random**

**# Define a dictionary of predefined responses**

**responses = {**

**"hello": ["Hi there!", "Hello!", "Hey!"],**

**"how are you": ["I'm good, thanks!", "I'm just a chatbot, but I'm here to help!"],**

**"bye": ["Goodbye!", "See you later!", "Have a great day!"],**

**"default": ["I'm not sure what you mean.", "Can you please rephrase that?", "Sorry, I didn't understand."],**

**}**

**# Function to generate a response**

**def get\_response(user\_input):**

**user\_input = user\_input.lower()**

**# Check if the user input is in the responses dictionary**

**if user\_input in responses:**

**return random.choice(responses[user\_input])**

**else:**

**return random.choice(responses["default"])**

**# Chat loop**

**print("Chatbot: Hello! I'm a simple chatbot. You can say 'bye' to exit.")**

**while True:**

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

**if user\_input.lower() == 'bye':**

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

**break**

**response = get\_response(user\_input)**

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

**PROCESS:**

1. **Import necessary modules:**

‘import random’

We import the random module to select random responses from the predefined lists.

1. **Define predefined responses:**

‘responses = { "hello": ["Hi there!", "Hello!", "Hey!"], "how are you": ["I'm good, thanks!", "I'm just a chatbot, but I'm here to help!"], "bye": ["Goodbye!", "See you later!", "Have a great day!"], "default": ["I'm not sure what you mean.", "Can you please rephrase that?", "Sorry, I didn't understand."], } ‘

We create a dictionary called responses where keys represent user inputs, and values are lists of possible responses to those inputs. We also include a "default" key for generic responses when there's no specific match.

1. **Define the get\_response function:**

‘def get\_response(user\_input): user\_input = user\_input.lower() if user\_input in responses: return random.choice(responses[user\_input]) else: return random.choice(responses["default"])’

The get\_response function takes a user input, converts it to lowercase for case insensitivity, checks if it matches any of the predefined responses, and returns a random response from the corresponding list. If no match is found, it returns a random default response.

1. **Start the chat loop:**

‘print("Chatbot: Hello! I'm a simple chatbot. You can say 'bye' to exit.") while True: user\_input = input("You: ") if user\_input.lower() == 'bye': print("Chatbot: Goodbye!") break response = get\_response(user\_input) print("Chatbot:", response)’

The chat loop is initiated with a greeting. It runs indefinitely using a while True loop until the user enters "bye." In each iteration of the loop, it prompts the user for input, checks if the input is "bye" to exit the loop, otherwise, it gets a response from the chatbot using the get\_response function and prints the chatbot's response.