**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:**

**The program I provided is a simplified rule ­-based chatbot that can understand a few pre defind intents and provided the response base on the those intents here’s the step by step process of how the program works**

**1.Import required libraries:**

**The program starts by importing any necessary In this case, there are no external libraies used.**

**2.Define intents and responses :**

**Intents are predefined categories of user input , for instance ,common intents include greeting , goodbye and thanks for each intent there is a a list of associated keyworks that the chatbot uses to identify the users intention**

**3.Preprocess input :**

**When the user enters a message the input is pre processed . the typically involves converting the input to lowercase of case intensitivity and removing punctuation**

**4. Match intent :**

**The program then matches the preprocessed user input to one of the defind intents by checking if any of the intents associated keywords are present in the user input**

**. the get\_intent function handles this matching process.**

**5:Generete response:**

**Once the intent is identified the program selects a response associated with that intent from the response associated with that intent from the response dictionary .if the there is no predefined respone for the intent it falls back to generic response**

**6. Intertact with the user :**

**The program enters a loop repeatedly the user for input ,recieive their messages and respond accordingly the chatbot continues to intract with the user utill the user types bye to exit the conservation .**

***HERE’S AN EXAMPLE OF HOW THE PROGRAM EXECUTION PROCEEDS*:**

**.**The program starts and display a greetings message from the chatbot

.It enters a loop where the user can provide input

. The user enter a message

. The program pre processes the user input (converts it to lowercase )

. It matches the users input to one of the defind intents based on keywords

. It generates a response based on the identified intent or provides a generic response if the intent is not recognized

. The chatbot displays the response to the user

. The loop continues ,and the user can provide more input

The program continues to loop and respond to the user until the user types “bye” to exit the conversation