CREATE A CHATBOT IN PYTHON
FINAL DEVELOPMENT PHASE

A.Paulin Grace Angelin
team member
Reg No :950521104026

1. **Data Collection:**

- Gather or create a dataset of conversations, which will be used for training the chatbot. You can collect text data from various sources, or you can use pre-existing datasets.

2. **Data Preprocessing:**

- Preprocess the conversation data to clean and format it. Common preprocessing steps include tokenization, removing stopwords, and stemming or lemmatization.

*3. **Natural Language Processing (NLP):***

- Utilize NLP libraries like NLTK (Natural Language Toolkit) or spaCy to perform text analysis and extract meaningful information from the text. NLP libraries can help with tasks like named entity recognition, sentiment analysis, and part-of-speech tagging.

*4. **Machine Learning Models:***

- Train a machine learning model to understand and generate text. A common approach is to use a sequence-to-sequence model with recurrent neural networks (RNNs) or transformer models like GPT (Generative Pre-trained Transformer). Libraries like TensorFlow or PyTorch are useful for implementing and training these models.

*5. **Dialog Management:***

- Implement dialog management to keep track of the conversation context and maintain a coherent chat flow. This can be achieved by using rule-based or state-based systems to manage the conversation.

*6. **User Interaction:***

- Create a user interface for interacting with the chatbot. This could be a web-based interface, a command-line interface, or integration with messaging platforms like Facebook Messenger or Slack. You can use libraries like Flask or Django for web-based interfaces.

*7. **Integration with Chat Platforms:***

- If your chatbot is meant to be used on messaging platforms, you will need to integrate it with platforms' APIs. For example, you can use the Facebook Messenger API, Slack API, or Telegram Bot API to enable interaction on those platforms.

*8. **Deployment:***

- Deploy your chatbot on a server or cloud platform. You can use services like AWS, Azure, or Heroku to host your chatbot application.

*9. **Testing and Training:***

- Continuously test and train your chatbot with real user interactions to improve its performance. You can use user feedback and automated testing to identify and address issues.

*10. **Monitoring and Maintenance:***

- Monitor the chatbot's performance and maintain it by updating the model and handling any issues or errors that may arise over time.

Here's a simple example of Python code using NLTK for text preprocessing:

``p`ython

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

nltk.download('punkt')
nltk.download('stopwords')

# Sample text preprocessing
text = "This is a sample sentence for text preprocessing."

# Tokenization
tokens = word_tokenize(text)

# Removing stopwords
stop_words = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word.lower() not in stop_words]

print(filtered_tokens)
```

Data collection:

```
import random

# Define a dictionary with some sample responses for the chatbot
responses = {
"hello": "Hello! How can I assist you today?",
 "how are you": "I'm just a computer program, so I don't have feelings, but
thanks for asking!",
   "what's your name": "I'm just a chatbot, so I don't have a name. You can call
me ChatGPT.",
"bye": "Goodbye! If you have more questions, feel free to come back
anytime.",
}


def chatbot_response(user_input):
   user_input = user_input.lower() # Convert user input to lowercase for case-
insensitive matching
for key in responses:
if key in user_input:
return responses[key]
  return "I'm sorry, I don't understand that. Can you please rephrase or ask a
different question?"


# Main chat loop
print("Chatbot: Hello! How can I assist you today? (Type 'bye' to exit)")


while True:
user_input = input("You: ")


if user_input.lower() == "bye":
print("Chatbot: Goodbye!")
break


response = chatbot_response(user_input)
```

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

Output:

```
Chatbot: Hello! How can I assist you today? (Type 'bye' to exit)

You: What's your name?

Chatbot: I'm just a chatbot, so I don't have a name. You can call me Chatbot.

You: How are you?

Chatbot: I'm just a computer program, so I don't have feelings, but thanks for
asking!
You: Bye

Chatbot: Goodbye!
```

Data processing:

```
import random


# Define a dictionary of predefined responses

responses = {

"hello": "Hello! How can I help you today?",

   "how are you": "I'm just a computer program, but I'm here to assist you!",

"what's your name": "I'm a chatbot. You can call me ChatGPT.",

"bye": "Goodbye! Have a great day!",

}


# Function to get the chatbot's response

def get_response(user_input):

user_input = user_input.lower() # Convert the input to lowercase
```

```
response = responses.get(user_input, "I'm sorry, I don't understand that.")

return response


# Main chat loop
print("Chatbot: Hello! I'm a chatbot. You can start chatting with me. Type '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)
```

Output:

```
Chatbot: Hello! I'm a chatbot. You can start chatting with me. Type 'bye' to exit.
You: Hello
Chatbot: Hello! How can I help you today?
You: What's your name?
Chatbot: I'm a chatbot. You can call me chatbot.
You: How are you?
Chatbot: I'm just a computer program, but I'm here to assist you!
You: Goodbye
Chatbot: Goodbye!
```

Dialog management:

LANGUAGE MODEL:

```
pip install chatterbot

pip install chatterbot_corpus
```

CODING:

```
from chatterbot import ChatBot

from chatterbot.trainers import ChatterBotCorpusTrainer


# Create a chatbot instance

chatbot = ChatBot('MyBot')


# Create a new trainer for the chatbot

trainer = ChatterBotCorpusTrainer(chatbot)


# Train the chatbot on English language data

trainer.train('chatterbot.corpus.english')


# Main chat loop

while True:

user_input = input('You: ')


# Exit the loop if the user says 'exit'

if user_input.lower() == 'exit':
 print('Bot: Goodbye!')
break

# Get the bot's response
   response = chatbot.get_response(user_input)

print('Bot:', response)
```

User interaction:

```
import random

# Define a dictionary of predefined responses
responses = {
"hello": ["Hi there!", "Hello!", "Hey!"],
   "how are you": ["I'm just a computer program, so I don't have feelings, but
I'm here to help you!", "I'm doing well, thanks for asking."],
  "what's your name": ["I'm just a chatbot, so I don't have a name. You can call
me ChatGPT.", "I'm nameless, but you can call me Chatbot."],
"bye": ["Goodbye!", "See you later!", "Have a great day!"],
}

def get_response(user_input):
user_input = user_input.lower()
   response = responses.get(user_input, "I'm not sure how to respond to that.")
```

```
 if isinstance(response, list):
return random.choice(response)
else:
return response

print("Hello! I'm your chatbot. You can start a conversation, and I'll do my best to respond.")
while True:
user_input = input("You: ")
if user_input.lower() == "exit":
print("Chatbot: Goodbye!")
break
else:
response = get_response(user_input)
print("Chatbot:", response)
```

Testing and training:

INSTALL NLTK:

```
pip install nltk
```

PYTHON SCRIPT;

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

```
nltk.download('punkt') # Download the NLTK data required for tokenization

# Define chatbot responses and patterns
pairs = [
[
r"(hello|hi|hey|good morning|good afternoon)",
["Hello!", "Hi there!", "Hey!", "Hello, how can I assist you today?"]
],
[
r"what is your name?",
["I'm a chatbot.", "I don't have a name.", "You can call me Chatbot."]
],
[
r"how are you?",
    ["I'm just a computer program, so I don't have feelings, but I'm here to help
you!"]
],
[
r"(.*) your name?",
["I'm a chatbot.", "I don't have a name.", "You can call me Chatbot."]
],
[
r"(what|tell me) (about|something about) (yourself|you)?",
["I'm a simple rule-based chatbot created with Python.", "I'm here to
answer your questions."]
],
[
 r"(bye|goodbye|see you later)",
        ["Goodbye!", "Have a great day!", "See you later!"]
],
]

# Create a chatbot
chatbot = Chat(pairs, reflections)

# Start the conversation
print("Hello, I'm your chatbot. Type 'bye' to exit.")
while True:
user_input = input("You: ")
if user_input.lower() == 'bye':
print("Chatbot: Goodbye!")
break
response = chatbot.respond(user_input)
print("Chatbot:", response)
```

CONVERSATION :

```
Hello, I'm your chatbot. Type 'bye' to exit.
You: Hi
Chatbot: Hello, how can I assist you today?
You: What's your name?
Chatbot: You can call me Chatbot.
You: How are you?
```

```
Chatbot: I'm just a computer program, so I don't have feelings, but I'm here to help you!

You: Tell me about yourself

Chatbot: I'm a simple rule-based chatbot created with Python.

You: Goodbye

Chatbot: Goodbye!
```