SKILL UPGRADE - INTERNSHIP

FINAL TASK

TASK TITLE : Chatbot Implementation

INTERN NAME : ANBU ABDUL KAREEM A

Task Description:

- **Create a simple chatbot using a natural language processing (NLP) library like spaCy or Rasa**
- **Implement basic conversation flows and responses.**
- **Test the chatbot with different queries to ensure a coherent and accurate interaction**

1. Choose My Tool:

- spaCy: Great for NLP tasks like tokenization, named entity recognition, and dependency parsing. It's not a chatbot framework, but you can build a rule-based chatbot using it.
- Rasa: Specifically designed for building conversational AI and chatbots. It comes with built-in NLU (Natural Language Understanding) and dialogue management.

2. Set Up Your Environment:

- Install the necessary packages. You can use pip to install these.

For **spaCy**:

```
pip install spacy
python -m spacy download  en_core_web_sm
```

For **Rasa**:

```
pip install rasa
```

3. Building a Basic Chatbot with spaCy:

Here's an example of how you might create a rule-based chatbot using spaCy.

4. Building a Basic Chatbot with Rasa:

- Rasa requires a bit more setup but allows for more complex and dynamic conversations

```
rasa init
```

- Define Intents and Responses:

In the nlu.yml and domain.yml files, define your intents and the corresponding responses.

Example of `nlu.yml`:

```
nlu:
- intent: greet
 examples: |
  - hey
  - hello
  - hi
- intent: goodbye
 examples: |
  - bye
  - goodbye
  - see you
```

Example of domain.yml:

```
responses:
 utter_greet:
  - text: "Hello! How can I assist you today?"
 utter_goodbye:
  - text: "Goodbye! Have a great day!"
intents:
 - greet
 - goodbye
```

5.Train the Model: Train the model with the following command:

```
rasa train
```

6.Run the Chatbot: Start the chatbot server:

```
rasa shell
```

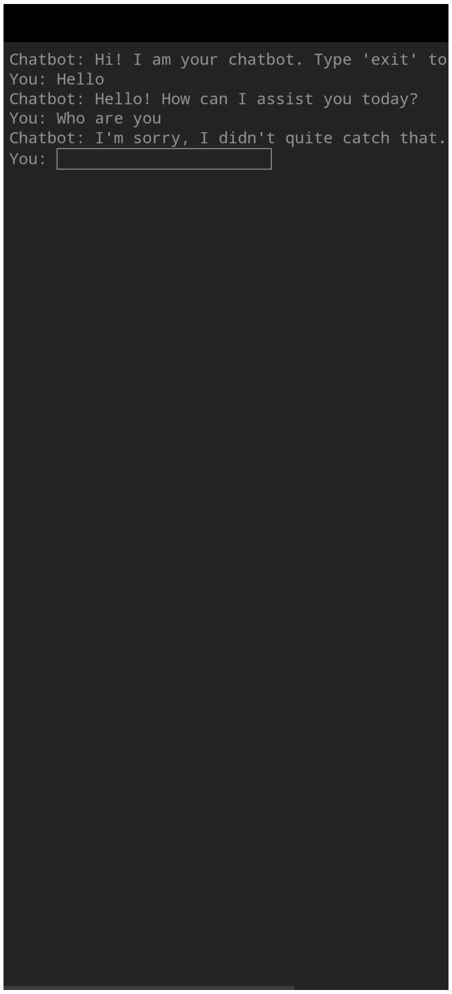
PROGRAM :

```
import spacy

# Load the spaCy model
nlp = spacy.load("en_core_web_sm")
```

```
# Define a dictionary of intents and responses
intents = {
  "greeting": ["hello", "hi", "hey", "good morning", "good evening"],
  "goodbye": ["bye", "goodbye", "see you later", "take care"],
  "thank_you": ["thank you", "thanks", "much appreciated", "thank you
so much"],
  "weather": ["what's the weather", "tell me the weather", "weather
forecast", "is it going to rain"],
  "name_query": ["what is your name", "who are you", "tell me your
name"],
  "creator": ["who created you", "who made you", "your creator", "who
built you"]
}

responses = {
  "greeting": "Hello! How can I assist you today?",
  "goodbye": "Goodbye! Have a great day!",
  "thank_you": "You're welcome! If you have more questions, feel
free to ask.",
  "weather": "I don't have access to weather data at the moment, but
you can check a weather app!",
  "name_query": "I'm your friendly chatbot! What can I help you
with?",
  "creator": "I was created by Anbu Abdul Kareem.",
  "fallback": "I'm sorry, I didn't quite catch that. Can you please
rephrase?",
}

# Function to get the intent of the user's query
def get_intent(user_input):
  doc = nlp(user_input.lower())
  for token in doc:
    for intent, keywords in intents.items():
      if token.lemma_ in keywords:
        return intent
  return "fallback"

# Function to get the response based on the intent
def get_response(intent):
  return responses.get(intent, responses["fallback"])

# Main chatbot function
def chatbot():
  print("Chatbot: Hi! I am your chatbot. Type 'exit' to end the
conversation.")
  while True:
    user_input = input("You: ")
```

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

        intent = get_intent(user_input)
        response = get_response(intent)
        print(f"Chatbot: {response}")

# Run the chatbot
if __name__ == "__main__":
    chatbot()
```

Output:



```
Chatbot: Hi! I am your chatbot. Type 'exit' to
You: Hello
Chatbot: Hello! How can I assist you today?
You: Who are you
Chatbot: I'm sorry, I didn't quite catch that.
You: 
```

Explanation:

- **Intent Recognition:** The `get_intent` function uses spaCy to lemmatize the user's input and checks if any keyword matches the predefined intents.

- **Response:** Based on the detected intent, the `get_response` function returns an appropriate response.

- **Fallback:** If the intent is not recognized, the chatbot provides a fallback response asking the user to rephrase.

EXPLANATION:

Now, if you ask the chatbot "Who created you?" or a similar question, it will respond with "I was created by Anbu Abdul Kareem."

Example Interaction:

vbnet

Copy code

You : who created you

Chatbot: I was created by Anbu Abdul Kareem.

This intent is now integrated into the chatbot, and it will correctly respond with the information about its creator.

****