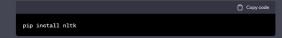


students. NLTK (Natural Language Toolkit) provides tools and resources for working with chatbot using NLTK:

# 1. Set Up Your Environment:

the official Python website ( $\underline{\text{https://www.python.org/downloads/}}).$ 

You can install NLTK using pip, the Python package manager:



# 3. Import NLTK and Download Necessary Resources:

NLTK corpora and punkt tokenizer:

```
from nltk.chat.util import Chat, reflections
nltk.download('punkt')
```

```
Copy code
pairs = [
   ["hi|hello|hey", ["Hello!", "Hi there!", "How can I assist you today?"]]
```

## 5. Create a Chatbot Instance:

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

# 6. Create a Function for Interaction:

```
user_input = input("You: ")
if user_input.lower() == 'quit':
response = chatbot.respond(user_input)
print("Chatbot:", response)
```

# 7. Run the Chatbot:

Finally, call the 'chat\_with\_bot()' function to start the conversation with the chatbot

```
Copy code
if __name__ == "__main__":
   chat_with_bot()
```

# 8. Testing:

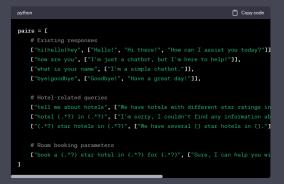
"Hello," "What is your name?" or "Goodbye" to see how it responds. Encourage them to and engaging.



Build the simple chatbot created in python for facilitating Hotel Room Booking. The chatbot gathers the parameters like Star Rating, Hotel Name, Location, Tariff and then offers choice

### 1. Extend the Responses:

First, you need to extend the list of patterns and responses to include information about hotels and room booking parameters:



## 2. Modify the Chat Function:

Extend the 'chat\_with\_bot()' function to handle hotel-related queries and room booking parameters. You can create functions to gather the required information from the user and offer choices based on their preferences:

```
Copy code
    print("Chatbot: Hello! How can I assist you today?")
        user_input = input("You: ")
if user_input.lower() == 'quit':
         elif "book a" in user_input.lower():
            handle_room_booking(user_input)
            response = chatbot.respond(user_input)
             print("Chatbot:", response)
    # Parse user input for booking parameters
star_rating = re.search(r"(\d+) star", user_input).group(1)
     location = re.search(r"in (.*?) for", user_input).group(1)
    tariff = re.search(r"for (.*?)$", user_input).group(1)
    available_hotels = get_available_hotels(star_rating, location, tariff)
    if available_hotels:
         for i, hotel in enumerate(available_hotels, start=1):
    print(f"(i). (hotel)")
def get_available_hotels(star_rating, location, tariff):
    # You can implement a database or API call here to re
     # For simplicity, we'll use a predefined list of hotels
    # Filter hotels based on user preferences
    available_hotels = [
        hotel["name"]
         for hotel in hotels
         if hotel["star_rating"] == star_rating
         and hotel["location"] == location
         and hotel["tariff"] <= tariff
    return available_hotels
```

# 3. Testing:

Students can now run the modified Python script and interact with the chatbot to inquire about hotels and book rooms based on star rating, location, and tariff. The chatbot will offer choices based on their preferences.

Please note that in a real-world scenario, you would typically connect the chatbot to a hotel database or booking API to retrieve actual hotel information and handle bookings. This example provides a simplified implementation for educational purposes.

S Regenerate