TITLE: CodTech IT Solutions Internship - Task Documentation: “**Python ChatBot**” Using PYTHON PROGRAMMING.

INTERN INFORMATION:

Name: Akash Siliveru

ID: COD5431

INTRODUCTION

In recent years, chatbots have emerged as powerful tools for automating tasks, providing assistance, and engaging users in interactive conversations. Python, with its simplicity and versatility, is an ideal choice for developing chatbot applications. In this essay, we embark on a journey to build a Python chatbot from the ground up. Our exploration encompasses key concepts of natural language processing, algorithm design, and user interaction. Through detailed explanations, code snippets, and accompanying images, we aim to provide a comprehensive guide for enthusiasts interested in delving into the exciting field of chatbot development.

Implementation

- **Define Responses:** Decide on the responses your chatbot will give for different inputs. You can start with a few basic responses and expand as needed
- **Implement Response Function:** Write a function that takes user input and returns an appropriate response based on the predefined responses.
- **Handle User Input:** Create a loop where the chatbot continuously waits for user input. Inside the loop, call the response function to generate a response based on the user input.
- **User Interaction:** Print the user's input and the chatbot's response to the screen.

CODE EXPLAINATION

Now, let's delve into the code implementation of our Python chatbot. We'll break down the code into several sections, explaining each component's functionality and providing code snippets.

Initialization: We begin by importing necessary libraries and initializing variables. Here's a breakdown of this section:

```
import random

# List of greetings and responses

greetings = ["hi", "hello", "hey"]

responses = ["Hi there!", "Glad you're here!", "What can I do for you today?"]

# Knowledge base for storing information

knowledge_base = {

"weather": "I can't access real-time weather information, but I can tell you some fun facts. Did you know...",

"time": "Unfortunately, I cannot access the current time directly. However, you can check your device's clock.",

"joke": "Why did the scarecrow win an award? Because he was outstanding in his field!",

}
```

In this section, we import the random module to enable random selection of responses. We define two lists, greetings and responses, to store greeting messages and corresponding responses. Additionally, we create a dictionary knowledge_base to store predefined information about various topics such as weather, time, and jokes.

Main Functionality - chat(message): The core functionality of our chatbot lies in the chat() function, which processes user input and generates responses. Let's examine the implementation:

```
def chat(message):

   message = message.lower()  # Convert message to lowercase for case-insensitive matching

    if any(word in message for word in greetings):

    return random.choice(responses)  # Respond to greetings

        elif message in knowledge_base:

      return knowledge_base[message]  # Retrieve information from knowledge base
    else:
     return "I am still learning, but I am always happy to chat! Tell me more about yourself."
```

In this function, we normalize the user input by converting it to lowercase to facilitate case-insensitive matching. We then check if the input contains any greeting words from the greetings list. If a greeting is detected, the chatbot responds with a random message from the responses list. If the input matches a key in the knowledge_base dictionary, the chatbot retrieves the corresponding value. Otherwise, it returns a default response indicating that it is still learning.

User Interaction Loop: To engage in a conversation with the chatbot, we implement a user interaction loop. Here's how it's done:

```
while True:
    user_input = input("You: ")
    print(user_input)
    response = chat(user_input)
   print("Chatbot:", response)
```

In this loop, the chatbot continuously prompts the user for input using the input() function. It then prints the user's input, passes it to the chat() function to generate a response, and finally prints the response from the chatbot.

User Interaction and Output: The user interacts with the chatbot by entering text input, to which the chatbot responds accordingly. Below is an example of a conversation between the user and the chatbot:

```
You: hello
hello
Chatbot: What can I do for you today?
You: Tell me a joke
Tell me a joke
Chatbot: Why did the scarecrow win an award? Because he was outstanding in his field!
```

You: What's the weather like?

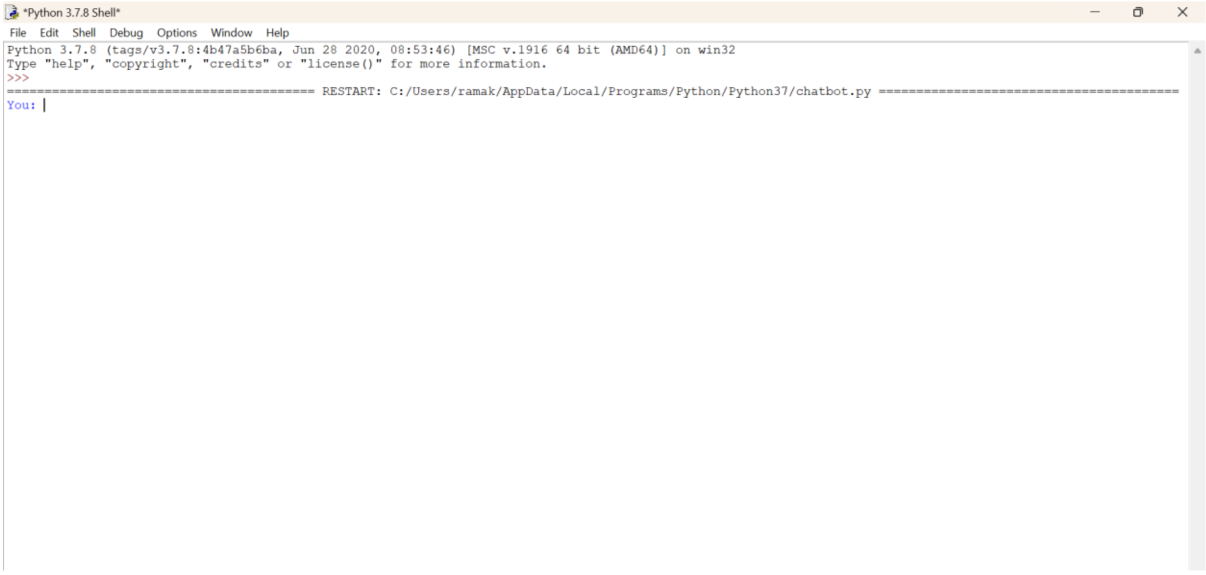
What's the weather like?

Chatbot: I can't access real-time weather information, but I can tell you some fun facts. Did you know…

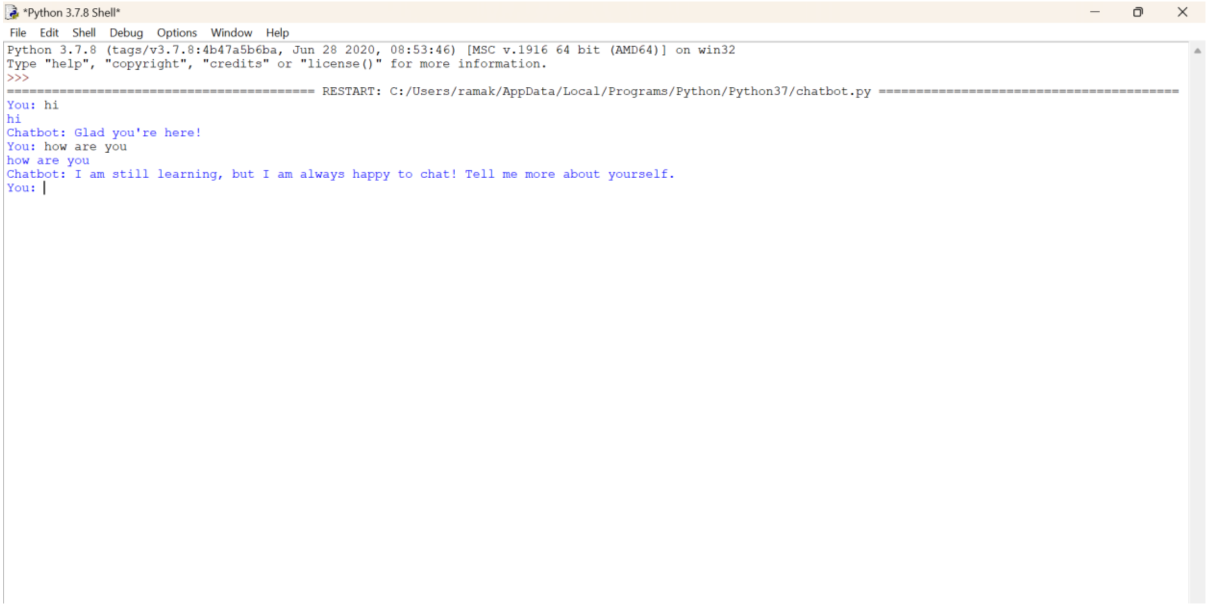
CONCLUSION

In conclusion, we have embarked on a journey to build a Python-based chatbot, exploring key components, code implementation, and user interaction. By following this comprehensive guide, readers gain insights into fundamental concepts of chatbot development, including natural language processing, response generation, and user interaction. Armed with this knowledge, they can customize and enhance the chatbot to suit their specific needs and preferences. As technology continues to evolve, chatbots will play an increasingly vital role in facilitating human-computer interactions, streamlining processes, and enriching user experiences

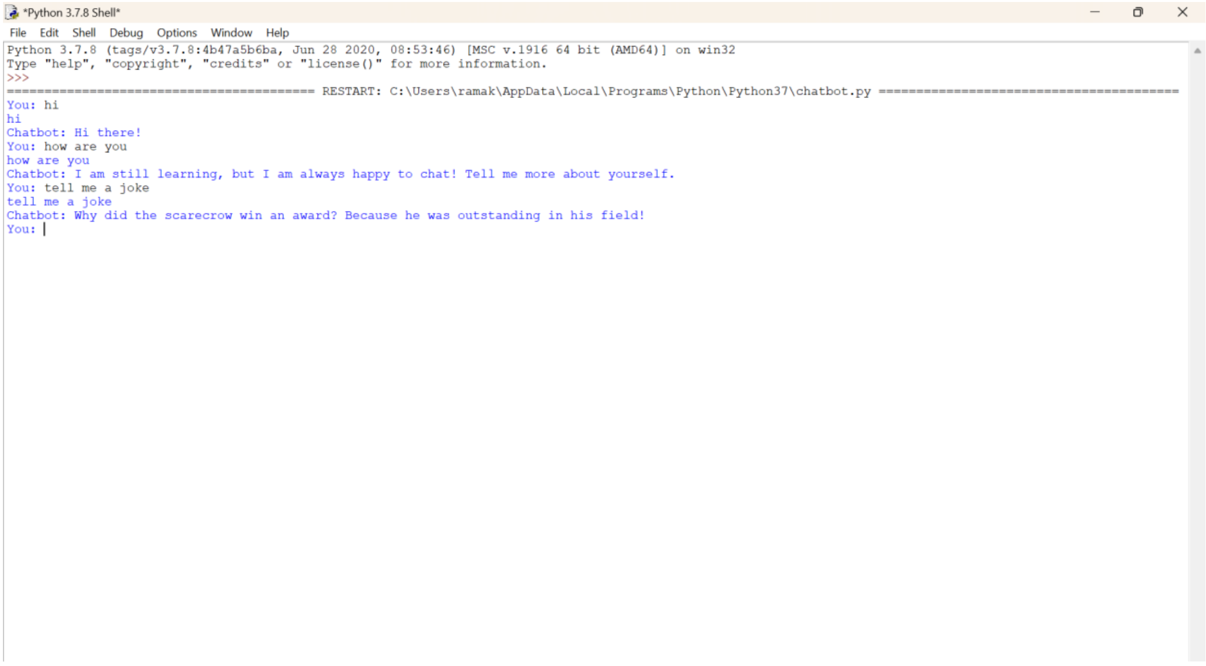
OUTPUT



```
Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========================================= RESTART: C:/Users/ramak/AppData/Local/Programs/Python/Python37/chatbot.py ========================================
You: |
```



```
Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
======================================== RESTART: C:/Users/ramak/AppData/Local/Programs/Python/Python37/chatbot.py ========================================
You: hi
hi
Chatbot: Glad you're here!
You: how are you
how are you
Chatbot: I am still learning, but I am always happy to chat! Tell me more about yourself.
You: |
```



```
Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
======================================== RESTART: C:\Users\ramak\AppData\Local\Programs\Python\Python37\chatbot.py ========================================
You: hi
hi
Chatbot: Hi there!
You: how are you
how are you
Chatbot: I am still learning, but I am always happy to chat! Tell me more about yourself.
You: tell me a joke
tell me a joke
Chatbot: Why did the scarecrow win an award? Because he was outstanding in his field!
You: |
```