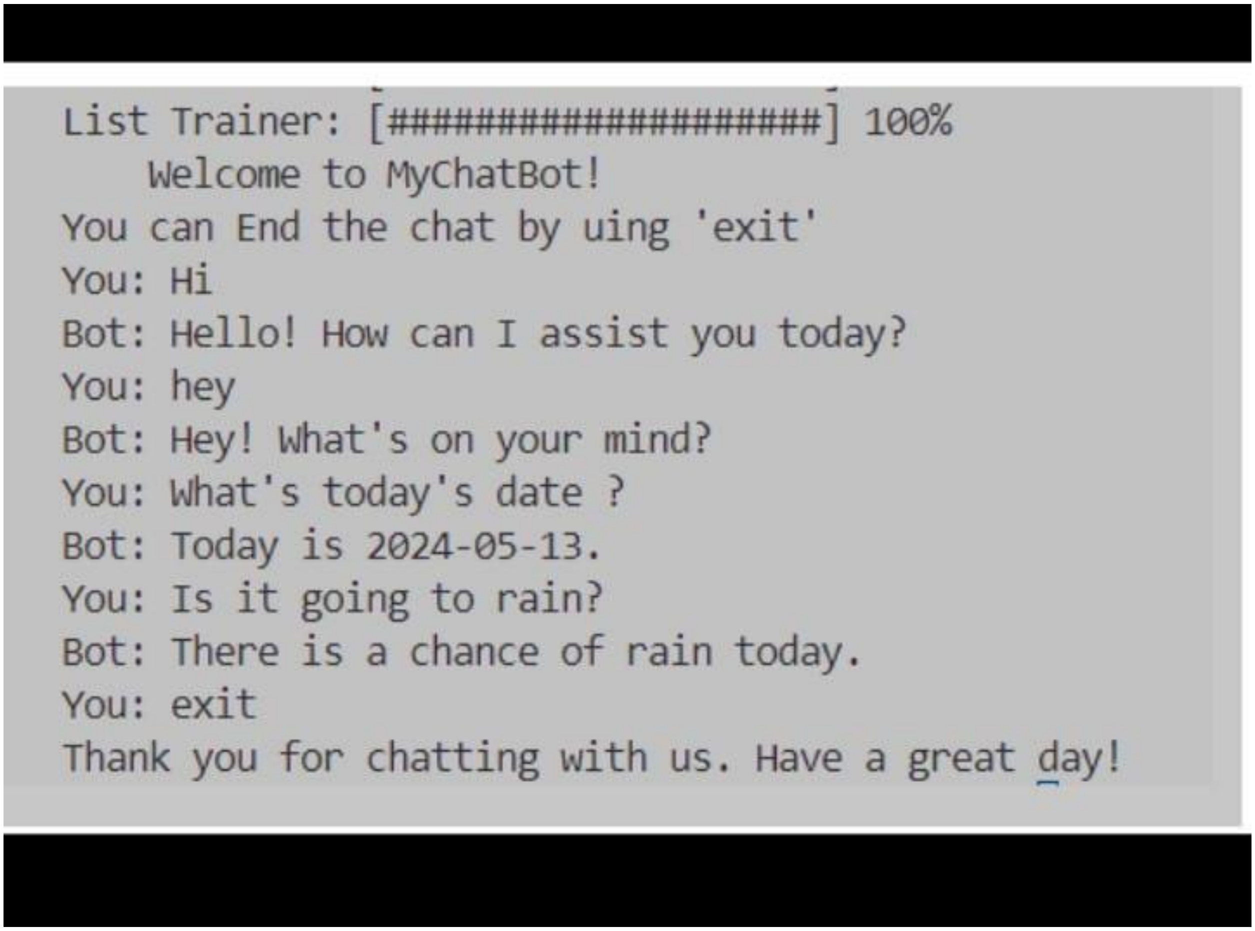
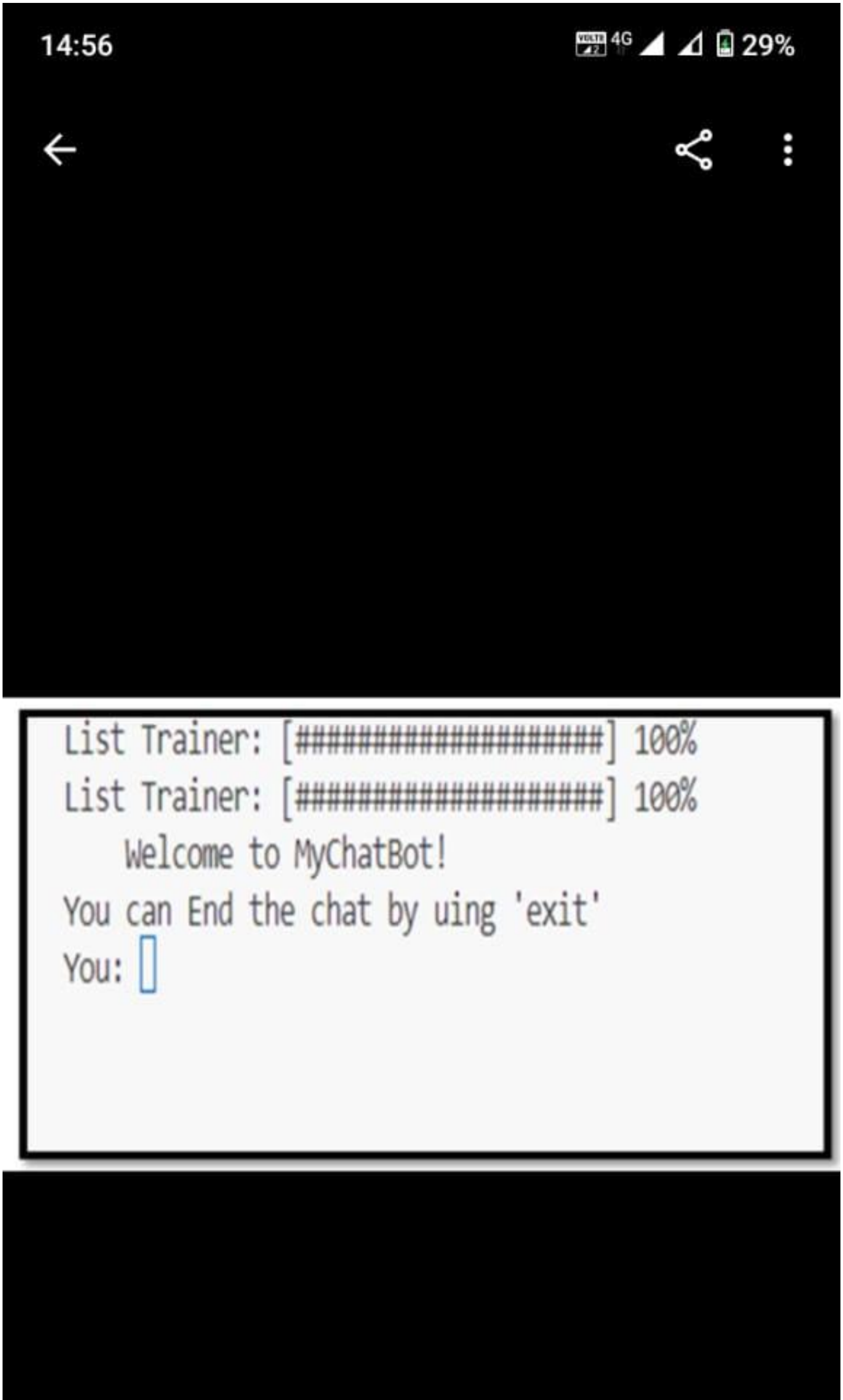
```
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer
from datetime import datetime
import pytz

# Create a chatbot instance
# Create a chatbot instance
chatbot = ChatBot(
    "MyChatBot",
    storage_adapter='chatterbot.storage.SQLStorageAdapter',
    database_uri='sqlite:///database.sqlite3'
)

# Function to get current time in Bengaluru
def get_current_time():
    timezone = pytz.timezone('Asia/Kolkata')
    return datetime.now(timezone).strftime('%H:%M:%S')

# Function to get current date in Bengaluru
def get_current_date():
    timezone = pytz.timezone('Asia/Kolkata')
    return datetime.now(timezone).strftime('%Y-%m-%d')

# Function to get current day in Bengaluru
def get_current_day():
    timezone = pytz.timezone('Asia/Kolkata')
    return datetime.now(timezone).strftime('%A')

# Function to get current year in Bengaluru
def get_current_year():
    timezone = pytz.timezone('Asia/Kolkata')
    return str(datetime.now(timezone).year)

# Custom function to check for single word matches
def get_response_for_single_word_match(user_input):
    # Check if the user input is 'date' or 'time'
    if 'date' in user_input.lower():
        return f"Today is {get_current_date()}."
    elif 'time' in user_input.lower():
        return f"The current time is {get_current_time()}."
    elif 'rain' in user_input.lower() or 'weather' in user_input.lower():
        # Here you can integrate an API call to a weather service to get real-time weather information
        # For now, we will return a placeholder response
        return "There is a chance of rain today."
    # Split the user input into words
    user_words = set(user_input.lower().split())
    # Iterate through each conversation pair
    for question, answer in conversation_pairs:
        # Split the question into words
        question_words = set(question.lower().split())
        # Check if any word in the user input matches any word in the question
        if user_words & question_words:
            return answer
    # If no single word match is found, return None
    return None


# Train the chatbot with a list of conversations
conversation_pairs = [
    ("Hi!", "Hello! How can I assist you today?"),
    ("Hello there!", "Hi! What can I help you with?"),
    ("Good morning!", "Good morning! How can I be of service?"),
    ("Hey!", "Hey! What's on your mind?"),
    ("What time is it?", f"The current time is {get_current_time()}."),
    ("What's today's date ?", f"Today is {get_current_date()}."),
    ("What day is it?", f"Today is {get_current_day()}."),
    ("What year are we in?", f"We are currently in {get_current_year()}."),
    ("Is it going to rain?", "There is a chance of rain today."),
    ("your village ?","Thallapaka")
]

trainer = ListTrainer(chatbot)
for pair in conversation_pairs:
    trainer.train(pair)

# Start the conversation loop
print("    Welcome to MyChatBot!")
print("You can End the chat by uing 'exit' " )
while True:
    user_input = input("You: ")
    if user_input.lower() == 'exit':
        print("Thank you for chatting with us. Have a great day!")
        break
    # Check for single word matches first
    response = get_response_for_single_word_match(user_input)
    if response:
        print("Bot:", response)
    else:
        # If no single word match, use the chatbot's default get_response method
        response = chatbot.get_response(user_input)
        print("Bot:", response)
```

output:



```
List Trainer: [####################] 100%
    Welcome to MyChatBot!
You can End the chat by uing 'exit'
You: Hi
Bot: Hello! How can I assist you today?
You: hey
Bot: Hey! What's on your mind?
You: What's today's date ?
Bot: Today is 2024-05-13.
You: Is it going to rain?
Bot: There is a chance of rain today.
You: exit
Thank you for chatting with us. Have a great day!
```



```
List Trainer: [####################] 100%
List Trainer: [####################] 100%
    Welcome to MyChatBot!
You can End the chat by uing 'exit'
You: 
```