## CREATE CHATBOT IN PYTHON

**911721104122-VINOTH KUMAR.S**

**Phase 3 Submission Document Project Title :** Creating chatbot

# Phase 3:Development Part 1

**Topic:**Start building a chatbot by preparing the environment and implementing basic user interactions.

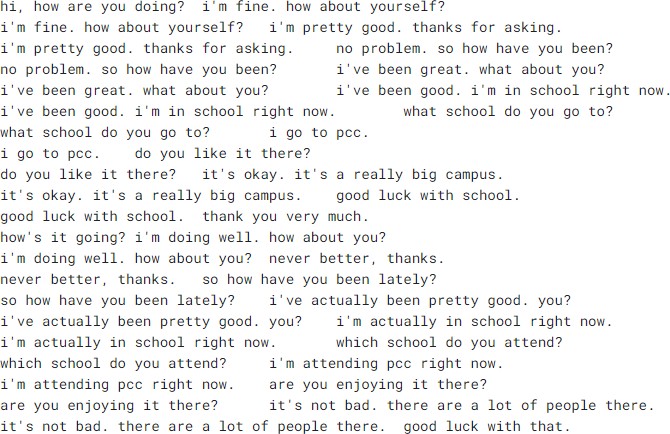
# Creating Chatbot

**Introduction:**

Building a chatbot using a specific dataset involves several steps, including setting up the environment and implementing basic user interactions. In this example, we will demonstrate how to create a chatbot using a dataset obtained from Kaggle. We'll utilize Python and libraries such as ChatterBot to facilitate this process.

By following the steps in this example, you'll learn how to set up the environment, load and preprocess the Kaggle dataset, and implement basic user interactions with your chatbot. While our chatbot's capabilities will be confined to the dialogues present in the dataset, this project serves as a foundation for understanding how to leverage external datasets for chatbot training.

# Given Data set:



To build a chatbot using the dataset from Kaggle, you can follow these steps:

## Download and Prepare the Dataset:

Download the dataset from Kaggle (http[s://www.k](http://www.kaggle.com/datasets/grafstor/simple-dialogs-for-)agg[le.co](http://www.kaggle.com/datasets/grafstor/simple-dialogs-for-)m[/datasets/grafstor/simple-dialogs-for-](http://www.kaggle.com/datasets/grafstor/simple-dialogs-for-) chatbot).

## Install Dependencies:

Install the necessary Python libraries for working with data and building a chatbot. We'll use pandas, ChatterBot, and ChatterBot's natural language processing library, spacy.

## pip install pandas pip install chatterbot

**pip install chatterbot\_corpus pip install spacy**

## Create a Python Script:

Create a Python script, e.g., chatbot\_with\_dataset.py.

## Implement the Chatbot: Program:

import pandas as pd

from chatterbot import ChatBot

from chatterbot.trainers import ListTrainer data = pd.read\_csv('dialogues.csv') chatbot = ChatBot('MyBot')

trainer = ListTrainer(chatbot) dialogs = data['User'] + data['Bot'] trainer.train(dialogs.tolist()) conversation\_history = []

def chat\_with\_bot():

print("Hello! I'm your chatbot. You can start a conversation, or type 'exit' to quit.")

while True:

user\_input = input("You: ")

if user\_input.lower() == 'exit': print("Bot: Goodbye!") break

elif user\_input.lower() == 'history': print("Bot: Conversation History") for entry in conversation\_history:

print(entry)

elif user\_input.lower() == 'clear history': conversation\_history.clear()

print("Bot: Conversation history cleared.") else:

response = chatbot.get\_response(user\_input) conversation\_history.append(f"You: {user\_input}") conversation\_history.append(f"Bot: {response}")

print("Bot:", response) chat\_with\_bot()

## Run the Chatbot:

Run the Python script by executing python chatbot\_with\_dataset.py in terminal or IDE.

# Sample Output:

### Hello! I'm your chatbot. You can start a conversation, or type 'exit' to quit**.**

**You:** hi,how are you doing? **Bot:**i’m fine how about yourself ? **You:** What's the weather like today?

### **Bot:** I'm not sure about the weather. I'm just a chatbot.

**You:** history

### **Bot:** Conversation History

**You:** hi,how are you doing? **Bot:**i’m fine how about yourself ? **You:** What's the weather like today?

### **Bot:** I'm not sure about the weather. I'm just a chatbot.

**You:** clear history

### **Bot:** Conversation history cleared.

**You:** exit

**Bot:** Goodbye!

# key tasks involved in creating a chatbot:

## Define Purpose and Use Case:

Determine the specific purpose and use case for your chatbot. Consider whether it will provide customer support, answer frequently asked questions, assist with tasks, or engage in casual conversations.

## Select a Platform:

Decide on the platform where your chatbot will be deployed. This could be a website, messaging apps (e.g., Facebook Messenger, WhatsApp), or a custom application.

## Choose the Technology Stack:

Select the technologies and tools you'll use to build the chatbot, including programming languages, libraries, and frameworks.

Common choices include Python, JavaScript, Node.js, and machine learning libraries like TensorFlow or PyTorch.

## Data Collection and Preprocessing:

Collect and preprocess data for training your chatbot. This may involve gathering conversation datasets, cleaning and formatting the data, and extracting relevant information.

## Train the Chatbot:

Train your chatbot using appropriate datasets. This training can involve supervised learning, reinforcement learning, or rule-based approaches, depending on the complexity of your chatbot.

## Natural Language Processing (NLP):

Implement Natural Language Processing techniques to enable the chatbot to understand and generate human-like text. This may include tasks like tokenization, entity recognition, sentiment analysis, and intent detection.

# Conclusion:

Building a chatbot is an exciting and complex endeavor with the potential to revolutionize various industries and enhance user experiences. In this process, we've explored the fundamental steps and considerations involved in creating a chatbot