PROJECT : CREATE A CHATBOT IN PYTHON

1.DATA RETRIEVAL:

Obtain a dataset of movies with information like titles, release years, and genres. You can use publicly available movie datasets or APIs like IMDb

2.Natural Language Processing (NLP):

Implement NLP techniques to understand and parse user queries. You'll need to extract key information from user input, such as the movie title, release year, or genre.

3.User Interaction:

Develop a conversational interface where the chatbot asks the user questions to gather details about the movie they're interested in. For example:Bot: "Hello! How can I assist you today?"User: "I'm looking for a comedy movie released in 2020."Bot: "Sure! Could you please specify the title or any other details"

4.Search and Matching:

Use the extracted information to search your movie dataset or API for relevant matches. Apply matching algorithms to find movies that match the user's criteria.

5.Response Generation:

Once you've found relevant movies, present the information back to the user in a clear and friendly manner. For instance:Bot: "I found a comedy movie released in 2020: 'The Half of It.' Would you like to know more

6.User Feedback:

Allow users to provide feedback and make clarifications. The chatbot should be able to handle scenarios where the user's initial query wasn't specific enough or if there are multiple matching movies.

7.Error Handling:

Implement error handling to gracefully handle situations where the user's request is unclear or when no matches are found. Provide informative error messages and suggestions.

8.Testing and Improvement:

Continuously test your chatbot with various user inputs to identify and resolve issues. You can improve the chatbot's performance by fine-tuning the NLP models or expanding the dataset.

First, make sure you have NLTK installed. You can install it using pip:pip install nltkHere's a simple Python script for a calculator chatbot:import nltk

from nltk.chat.util import Chat, reflections

# Define a list of patterns and responses

patterns = [

(r'(\d+) plus (\d+)', lambda x, y: int(x) + int(y)),

(r'(\d+) minus (\d+)', lambda x, y: int(x) - int(y)),

(r'(\d+) times (\d+)', lambda x, y: int(x) \* int(y)),

(r'(\d+) divided by (\d+)', lambda x, y: int(x) / int(y))

]

# Create a Chatbot instance

calculator\_bot = Chat(patterns, reflections)

print("Hello, I am a calculator chatbot. You can ask me to perform calculations.")

while True:

user\_input = input("You: ")

if user\_input.lower() == 'exit':

print("Calculator Bot: Goodbye!")

break

response = calculator\_bot.respond(user\_input)

print("Calculator Bot:", response)This code defines patterns and their corresponding lambda functions to handle different types of mathematical operations. It uses regular expressions to match user input. You can extend this chatbot to handle more complex calculations and interactions as needed.

To use the chatbot, run the script and enter queries like "2 plus 3" or "10 divided by 2." To exit the chat, type "exit."