CODING FOR CREATING A NEW CHATBOX

import tkinter as tk

from tkinter import scrolledtext

class ChatBotGUI:

def \_\_init\_\_(self, master):

self.master = master

master.title("ChatBot")

self.chat\_history = scrolledtext.ScrolledText(master, width=50, height=20, wrap=tk.WORD)

self.chat\_history.pack(padx=10, pady=10)

self.input\_label = tk.Label(master, text="User Input:")

self.input\_label.pack(pady=(10, 0))

self.user\_input = tk.Entry(master, width=50)

self.user\_input.pack(padx=10, pady=(0, 10))

self.send\_button = tk.Button(master, text="Send", command=self.send\_message)

self.send\_button.pack()

# Dictionary to store previous questions and responses

self.qa\_pairs = {}

self.chat\_history.insert(tk.END, "ChatBot: Hello! How can I help you today?\n")

self.chat\_history.configure(state='disabled')

def send\_message(self):

message = self.user\_input.get()

self.user\_input.delete(0, tk.END)

if message.strip() == "":

return

self.chat\_history.configure(state='normal')

self.chat\_history.insert(tk.END, f"You: {message}\n")

# Check if we have a response in previous QA pairs

if message.lower() in self.qa\_pairs:

response = self.qa\_pairs[message.lower()]

else:

response = bot\_response(message)

self.qa\_pairs[message.lower()] = response # Store new question and response

self.chat\_history.insert(tk.END, f"ChatBot: {response}\n")

self.chat\_history.configure(state='disabled')

def bot\_response(message):

# Responses about NLP models

if "hello" in message.lower() or "hi" in message.lower():

return "Hello! How are you?"

elif "bye" in message.lower():

return "Goodbye! Have a nice day."

elif "thank you" in message.lower():

return "your welcome!, tell me if i can assist you further"

elif "bert" in message.lower():

return "BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model designed for pretraining deep bidirectional representations. It has achieved state-of-the-art results on various NLP tasks, including question answering and natural language inference."

elif "transformer" in message.lower():

return "The Transformer is a deep learning model known for its attention mechanism that helps it learn contextual relationships in input sequences. It consists of encoder and decoder layers, which process input and output sequences by attending to different parts of the input to understand context."

elif "gpt-4" in message.lower():

return "GPT-4 (Generative Pretrained Transformer 4) is an advanced version of OpenAI's GPT series, known for its large scale and capabilities in understanding and generating human-like text. It builds upon the architecture and improvements of GPT-3, potentially with larger capacity and better performance."

elif "xlnet" in message.lower():

return "XLNet (eXtreme Multi-Label Learning Network) is a transformer-based model that integrates autoregressive and autoencoding approaches for pretraining. It aims to capture bidirectional dependencies better than standard autoregressive models by using permutation language modeling."

elif "roberta" in message.lower():

return "RoBERTa (Robustly Optimized BERT Approach) is an optimized version of BERT with modifications to hyperparameters, training data, and objectives, aiming to improve performance. It uses dynamic masking and training strategies to achieve better results on NLP benchmarks."

# Additional responses

elif "natural language processing" in message.lower():

return "Natural Language Processing (NLP) is a field of artificial intelligence focused on enabling machines to understand and process human language. It involves tasks such as text classification, machine translation, sentiment analysis, and more."

elif "text generation" in message.lower():

return "Text generation is the task of automatically generating coherent and contextually appropriate text based on a given prompt or input. Models like GPT-4 and Transformer excel at this task by learning to predict the next word in a sequence."

elif "question answering" in message.lower():

return "Question Answering (QA) is a task in NLP where the goal is to provide accurate and relevant answers to questions posed in natural language. Models such as BERT and XLNet have been used successfully for QA tasks."

elif "sentiment analysis" in message.lower():

return "Sentiment Analysis is the process of determining the sentiment expressed in a piece of text, whether it's positive, negative, or neutral. NLP models can classify sentiment based on the language used in the text."

# Default response

else:

return "Sorry, I don't have information on that topic. Can I help you with something else?"

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

chatbot\_gui = ChatBotGUI(root)

root.mainloop()

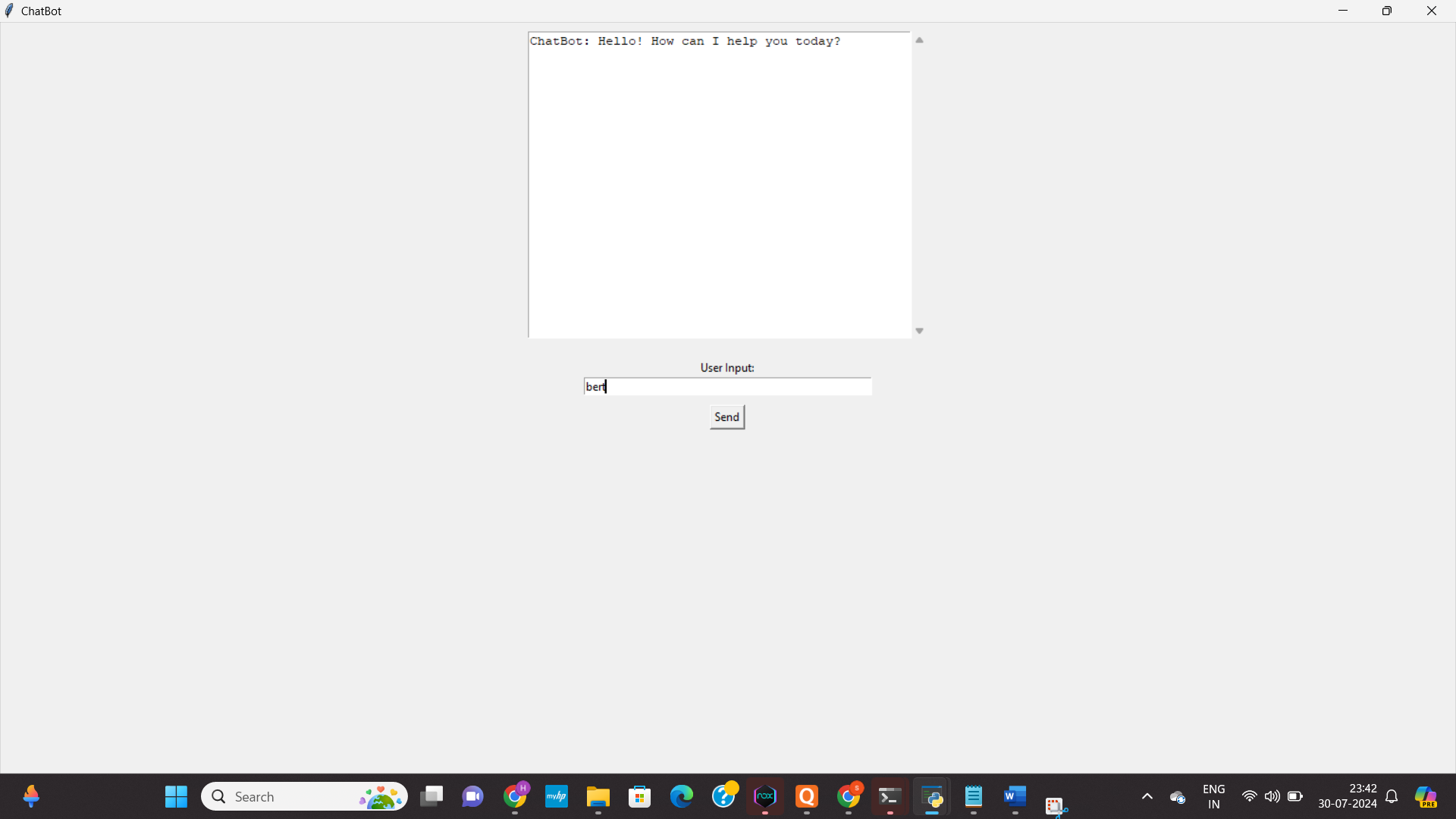
QUESTION AND ANSWER

STEPS TO BE FOLLOWED

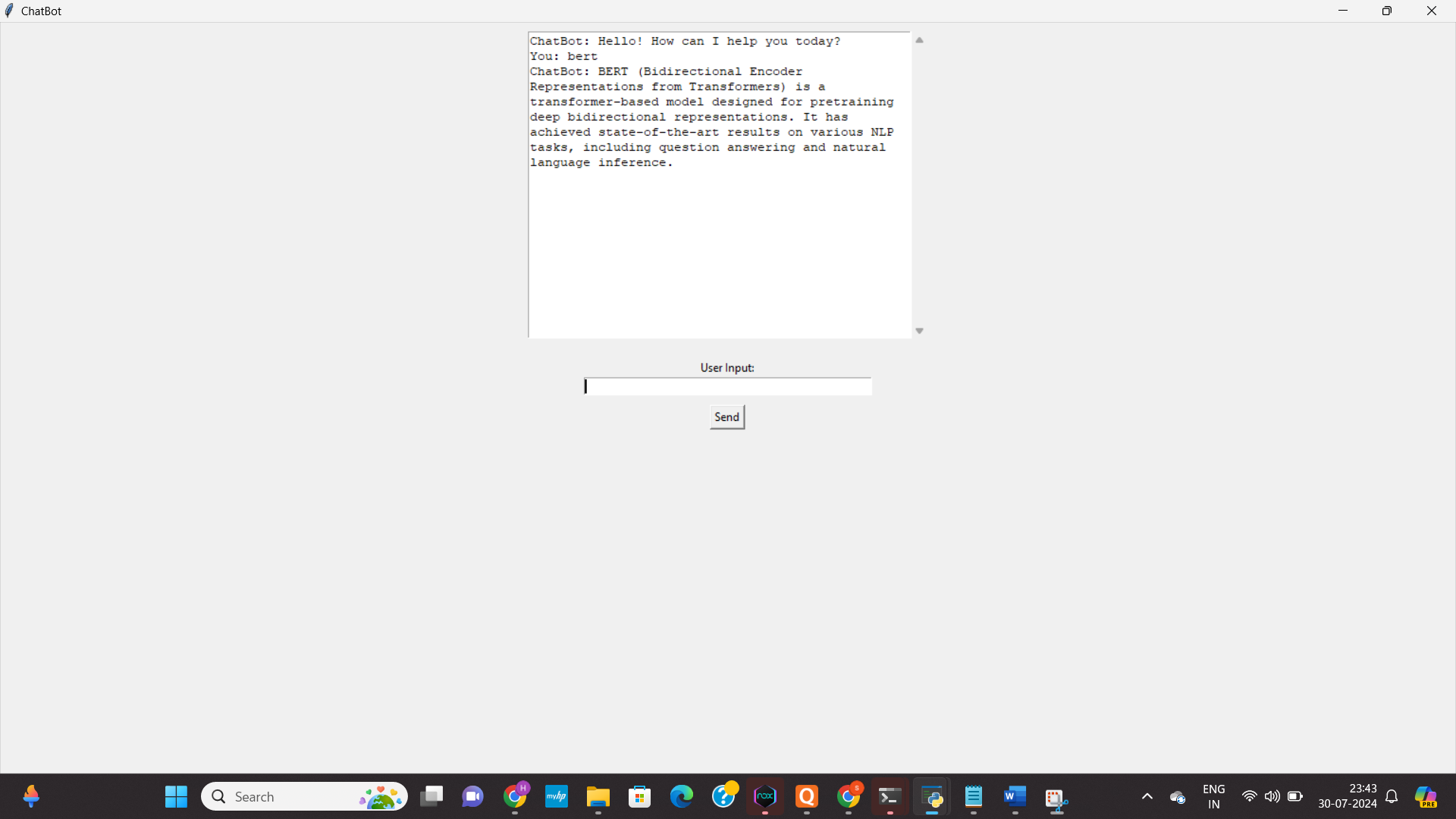
1.ENTER THE WORD TO BE SEARCHED WHICH IS SPECFIED IN THE CODE

EXAMPLE:

bert



2. PRESS SEND WHICH IS DISPLAYED IN SCREEN IF THE QUESTION IS MENTIONED IN THE CODE THEN U WILL GET ANSWER



LIST OF QUESTIONS THAT CAN BE ASKED

Hello

thank you

transformer

bert

gpt-4

xlnet

roberta