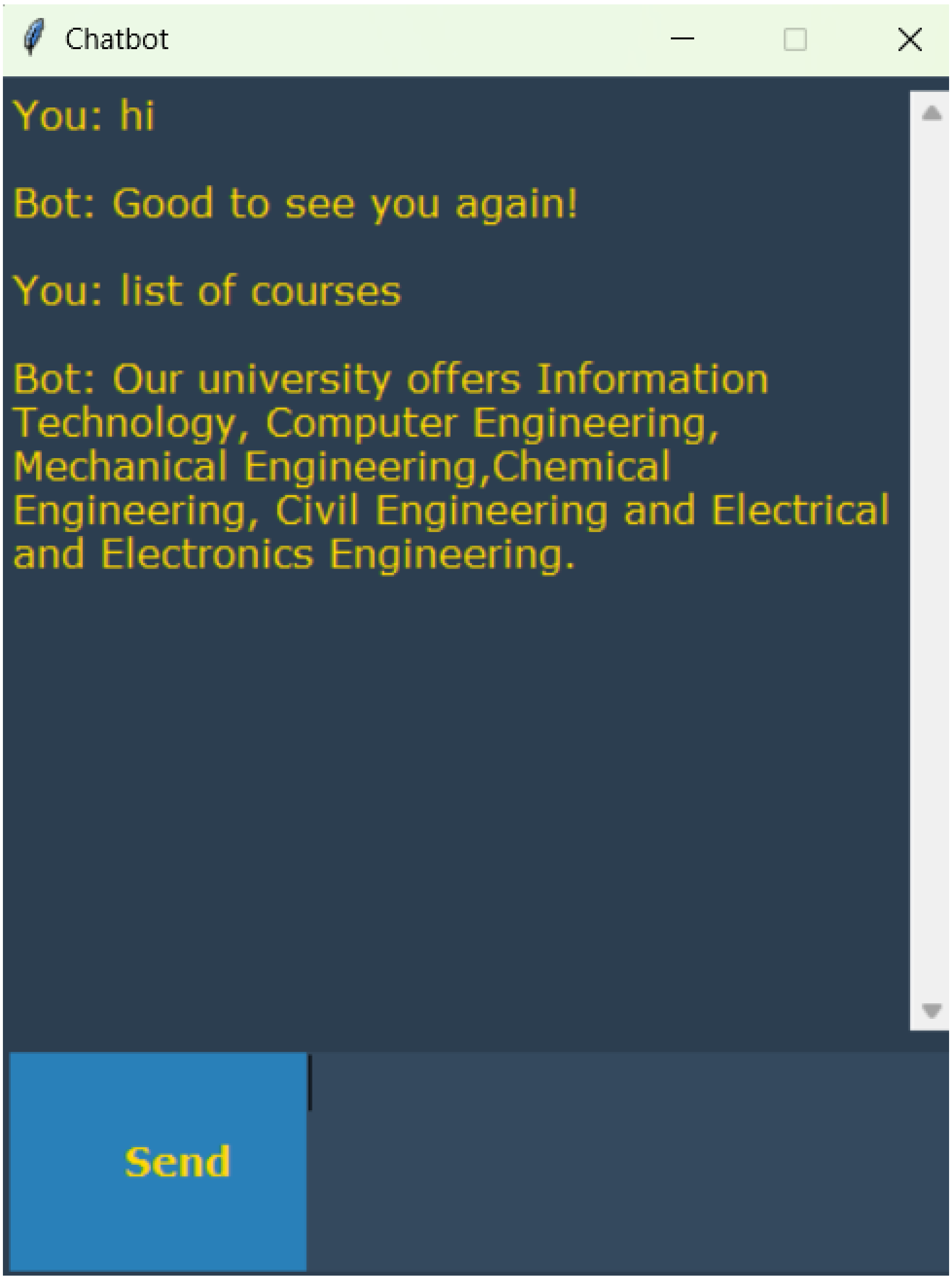
Source Code:-

```
import nltk
from nltk.stem import WordNetLemmatizer
import pickle
import numpy as np
from keras.models import load_model
import json
import random
import tkinter
from tkinter import *

# Initialize the lemmatizer
lemmatizer = WordNetLemmatizer()

# Load the model, intents, words, and classes
model = load_model('chatbot_model.h5')
intents = json.loads(open('intents.json').read())
words = pickle.load(open('words.pkl', 'rb'))
classes = pickle.load(open('classes.pkl', 'rb'))

def clean_up_sentence(sentence):
    sentence_words = nltk.word_tokenize(sentence)
    sentence_words = [lemmatizer.lemmatize(word.lower()) for word in sentence_words]
    return sentence_words

def bow(sentence, words, show_details=True):
    sentence_words = clean_up_sentence(sentence)
    bag = [0] * len(words)
    for s in sentence_words:
        for i, w in enumerate(words):
            if w == s:
                bag[i] = 1
                if show_details:
                    print("found in bag: %s" % w)
    return np.array(bag)

def predict_class(sentence, model):
    p = bow(sentence, words, show_details=False)
    res = model.predict(np.array([p]))[0]
    ERROR_THRESHOLD = 0.25
    results = [[i, r] for i, r in enumerate(res) if r > ERROR_THRESHOLD]
    return_list = []
    for r in results:
        return_list.append({"intent": classes[r[0]], "probability": str(r[1])})
    return return_list

def getResponse(ints, intents_json):
    tag = ints[0]['intent']
    list_of_intents = intents_json['intents']
    for i in list_of_intents:
        if i['tag'] == tag:
            result = random.choice(i['responses'])
            break
    return result

def chatbot_response(msg):
    ints = predict_class(msg, model)
    res = getResponse(ints, intents)
    return res
```

```
# GUI setup
def send():
    msg = EntryBox.get("1.0", 'end-1c').strip()
    EntryBox.delete("0.0", END)

    if msg != '':
        ChatLog.config(state=NORMAL)
        ChatLog.insert(END, "You: " + msg + '\n\n')
        ChatLog.config(foreground="#FFD700", font=("Verdana", 12))  # Brighter color for user messages

        res = chatbot_response(msg)
        ChatLog.insert(END, "Bot: " + res + '\n\n')

        ChatLog.config(state=DISABLED)
        ChatLog.yview(END)

base = Tk()
base.title("Chatbot")
base.geometry("400x500")
base.resizable(width=FALSE, height=FALSE)

# Colors
bg_color = "#2c3e50"
text_color = "#FFD700"  # Brighter yellow color for bot messages
button_color = "#2980b9"
entry_bg_color = "#34495e"
entry_text_color = "#ecf0f1"
button_active_color = "#3498db"

# Create chat window
```

```
ChatLog = Text(base, bd=0, bg=bg_color, fg=text_color, height="8", width="50",
font="Arial", wrap=WORD)
ChatLog.config(state=DISABLED)

# Bind scrollbar to chat window
scrollbar = Scrollbar(base, command=ChatLog.yview, cursor="heart")
ChatLog['yscrollcommand'] = scrollbar.set

# Create button to send message
SendButton = Button(base, font=("Verdana", 12, 'bold'), text="Send", width="12", height=5,
bd=0, bg=button_color, activebackground=button_active_color, fg=text_color,
command=send)

# Create the box to enter message
EntryBox = Text(base, bd=0, bg=entry_bg_color, fg=entry_text_color, width="29",
height="5", font="Arial", wrap=WORD)

# Place all components on the screen
scrollbar.place(x=376, y=6, height=386)
ChatLog.place(x=6, y=6, height=386, width=370)
EntryBox.place(x=128, y=401, height=90, width=265)
SendButton.place(x=6, y=401, height=90)

base.config(bg=bg_color)
scrollbar.config(bg=bg_color, troughcolor=bg_color, activebackground=button_active_color)

base.mainloop()
```

OUTPUT:-



```
You: hi

Bot: Good to see you again!

You: list of courses

Bot: Our university offers Information
Technology, Computer Engineering,
Mechanical Engineering,Chemical
Engineering, Civil Engineering and Electrical
and Electronics Engineering.
```