**SOURCE CODE:**

import os

import ssl

import nltk

import json

import streamlit as st

import random

import re

from collections import defaultdict

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.linear\_model import LogisticRegression

from nltk.tokenize import word\_tokenize

import string

ssl.\_create\_default\_https\_context = ssl.\_create\_unverified\_context

nltk.data.path.append(os.path.abspath("nltk\_data"))

nltk.download('punkt')

departments = {

    "cse": [

        "Dr. N. Bhalaji – Principal",

        "Dr. D. C. Joy Winnie Wise – Professor",

        "Dr. Lalitha R – Professor",

        "Dr. V. Anjana Devi – Professor",

        "Dr. R. Saravanan – Associate Professor",

        "Dr. Pandithurai O – Associate Professor",

        "Dr. T. Rajendran – Associate Professor",

        "Dr. Anwar Basha H. – Associate Professor",

        "Ashok M – Associate Professor",

        "C. Balaji – Associate Professor",

        "Dr. T. Nithya – Assistant Professor",

        "Dr. Ranjith Kumar M.V. – Assistant Professor",

        "C. Chairmakani – Assistant Professor",

        "AG. Noorul Julaiha – Assistant Professor",

        "B. Sriman – Assistant Professor",

        "R. Arunkumar – Assistant Professor",

        "Vijayalakshmi R. – Assistant Professor",

        "S. H. Annie Silviya – Assistant Professor",

        "S. Uma – Assistant Professor",

        "E. Venitha – Assistant Professor",

        "C.S. Somu – Assistant Professor",

        "G. Kavitha – Assistant Professor",

        "E. Pooja – Assistant Professor",

        "G. Sumathi – Assistant Professor",

        "R. Tamilselvan – Assistant Professor",

        "J. Praveen Kumar – Assistant Professor",

        "Dr. N. Indumathi – Assistant Professor",

        "Pugazhvendan I. – Professor of Practice",

        "Anantha Krishnan A. – Associate Professor of Practice",

        "Mugesh Hariharasudan – Assistant Professor of Practice"

    ],

    "ece": [

        "Dr. Sundar Rangarajan – Professor",

        "Dr. G. Nirmala Priya – Professor",

        "Dr. M. Malathi – Professor",

        "Dr. H. Sivaram – Associate Professor",

        "Dr. E. Sivanantham – Associate Professor",

        "Dr. M. Chitra – Associate Professor",

        "Dr. R. Sanmuga Sundaram – Associate Professor",

        "Dr. T. Roosefert Mohan – Associate Professor",

        "Chinnammal V – Assistant Professor",

        "Subashini V – Assistant Professor",

        "Kalyan Kumar G – Assistant Professor",

        "Kalaivani S – Assistant Professor",

        "Balaji A – Assistant Professor",

        "Malarvizhi C – Assistant Professor",

        "Vanathi A – Assistant Professor",

        "Charulatha Srinivasan – Assistant Professor",

        "Shofia Priyadharshini D. – Assistant Professor",

        "S. Sangeetha – Assistant Professor"

    ],

    "cce": [

        "Dr. C. Ganesh – Professor",

        "Dr. S. Ashok Kumar – Professor",

        "Dr. P. Sathish Kumar – Associate Professor",

        "Manimaran B. – Assistant Professor",

        "V. Sushmitha – Assistant Professor",

        "S. Bharath – Assistant Professor",

        "G. Saravanan – Assistant Professor",

        "N. Dharmaraj – Assistant Professor",

        "Vigneshvar D. – Associate Professor of Practice"

    ],

    "ee\_vlsi\_d&t": [

        "Dr. I. Chandra – Professor",

        "Franklin Telfer L – Assistant Professor",

        "Dr. Sheela S – Assistant Professor"

    ],

    "ec\_vlsi": [

        "Dr. S. Manjula – Professor",

        "Jayamani K – Assistant Professor",

        "Monica M. – Assistant Professor",

        "Monikapreethi S.K. – Assistant Professor"

    ],

    "csbs": [

        "Dr. K. Ramkumar – Professor",

        "Dr. Subha S – Associate Professor",

        "Dr. S. Sridhar – Associate Professor",

        "M. Babu – Associate Professor",

        "Loganayaki D – Assistant Professor",

        "S. Sathiyan – Assistant Professor",

        "K. Fouzia Sulthana – Assistant Professor",

        "T. Pandiarajan – Assistant Professor",

        "R. Deepak – Assistant Professor",

        "M. Baskar – Assistant Professor",

        "K. Jayashree – Assistant Professor",

        "J. Lakshmikanth – Assistant Professor",

        "Dr. P. Kalaivani – Assistant Professor",

        "Dr. J. Maria Arockia Dass – Assistant Professor",

        "P. Jyothy – Professor of Practice"

    ],

    "ai\_ds": [

        "Dr. N. Kanagavalli – Assistant Professor",

        "Dr. A. Arthi – Professor",

        "Dr. Srivenkateswaran C. – Professor",

        "Dr. M. Vivekanandan – Associate Professor",

        "Dr. B.N. Karthik – Associate Professor",

        "Dr. S. Niranjana – Assistant Professor",

        "R. Kennady – Assistant Professor",

        "S. Saranya – Assistant Professor",

        "S. Selvakumaran – Assistant Professor",

        "R. Saranya – Assistant Professor",

        "R. Kalaiyarasi – Assistant Professor",

        "V. Deepa – Assistant Professor",

        "B. Sasikala – Assistant Professor",

        "M. Bhavani – Assistant Professor",

        "S. Vaijayanthi – Assistant Professor",

        "S. Sahunthala – Assistant Professor",

        "T. Sam Paul – Assistant Professor",

        "G. Baby Saral – Assistant Professor",

        "M. Sneha – Assistant Professor",

        "Dr. Kalaiselvi S. – Assistant Professor",

        "A. Anbumani – Assistant Professor",

        "H. Hemal Babu – Assistant Professor",

        "V. Madhan – Assistant Professor",

        "Javis Jerald – Professor of Practice",

        "Nandhini – Assistant Professor of Practice",

        "K. Subashini – Assistant Professor of Practice",

        "Farzana B. – Assistant Professor of Practice"

    ],

    "cse\_ai\_ml": [

        "Dr. K. Regin Bose – Professor",

        "S. Shanthana – Assistant Professor",

        "F. Merlin Christo – Assistant Professor",

        "C. Gethara Gowri – Assistant Professor",

        "P. Somasundari – Assistant Professor",

        "K.G. Sara Rose – Professor of Practice"

    ],

    "mechanical": [

        "Dr. N. Pragadish – Professor",

        "Dr. Deepak Suresh – Professor",

        "Rajeswaran P. S. – Professor",

        "Dr. Rajesh Kanna S. K. – Professor",

        "Dr. M. Bakkiyaraj – Associate Professor",

        "Dr. Muthu G – Associate Professor",

        "Dr. Sai Krishnan G – Assistant Professor",

        "Dr. N. Sivashanmugam – Assistant Professor",

        "Dr. S. Bharani Kumar – Assistant Professor",

        "Srinivasan S. – Assistant Professor",

        "Vivek S – Assistant Professor"

    ]

}

member\_to\_departments = defaultdict(list)

for department, members in departments.items():

    for member in members:

        name = member.split("–")[0].strip().lower()

        member\_to\_departments[name].append(department)

first\_name\_to\_members = defaultdict(list)

for member\_full\_name in member\_to\_departments:

    first\_name = member\_full\_name.split()[0]

    first\_name\_to\_members[first\_name].append(member\_full\_name)

def load\_intents(file\_path):

    with open(file\_path, 'r', encoding='utf-8') as file:

        intents = json.load(file)

    return intents

intents\_file\_path = "D:\Mridula\stud chatbot\Final Code\intents.json"

intents = load\_intents(intents\_file\_path)

vectorizer = TfidfVectorizer()

clf = LogisticRegression(random\_state=0, max\_iter=10000)

tags = []

patterns = []

for intent in intents['intents']:

    for pattern in intent['patterns']:

        tags.append(intent['tag'])

        patterns.append(pattern)

x = vectorizer.fit\_transform(patterns)

y = tags

clf.fit(x, y)

def preprocess\_text(text):

    text = text.lower()

    text = text.translate(str.maketrans('', '', string.punctuation))

    return text

def extract\_name(tokens):

    if 'does' in tokens:

        idx = tokens.index('does')

        name\_tokens = tokens[idx + 1:]

        stopwords = {'belong', 'to', 'department'}

        name = ' '.join([token for token in name\_tokens if token not in stopwords])

        return name.strip()

    elif 'belong' in tokens:

        idx = tokens.index('belong')

        name\_tokens = tokens[idx + 1:]

        stopwords = {'to', 'department'}

        name = ' '.join([token for token in name\_tokens if token not in stopwords])

        return name.strip()

    elif 'is' in tokens:

        idx = tokens.index('is')

        name\_tokens = tokens[idx + 1:]

        stopwords = {'in', 'which', 'department'}

        name = ' '.join([token for token in name\_tokens if token not in stopwords])

        return name.strip()

    else:

        return tokens[-1] if tokens else ""

def get\_intent(user\_input):

    tokens = word\_tokenize(preprocess\_text(user\_input))

    if any(word in tokens for word in ["what", "list", "give", "show", "all"]):

        for department in departments.keys():

            department\_tokens = department.replace("\_", " ").split()

            if any(token in department\_tokens for token in tokens):

                return ("list\_members\_intent", department)

    if any(word in tokens for word in ["which", "does", "belong", "belongs", "is"]):

        name = extract\_name(tokens)

        if name:

            return ("find\_department\_intent", name)

    return ("unknown\_intent", None)

def format\_department\_name(department\_key):

    format\_mapping = {

        "cse": "CSE",

        "ece": "ECE",

        "cce": "CCE (Computer and Communication)",

        "ee\_vlsi\_d&t": "EE VLSI D&T",

        "ec\_vlsi": "EC VLSI",

        "csbs": "CSBS (Computer Science and Business Systems)",

        "ai\_ds": "AI&DS (Artificial Intelligence and Data Science)",

        "cse\_ai\_ml": "CSE (AI&ML)",

        "mechanical": "Mechanical"

    }

    return format\_mapping.get(department\_key, department\_key.upper())

def respond(intent, data):

    if intent == "list\_members\_intent":

        department = data

        department\_display = format\_department\_name(department)

        members = departments.get(department, [])

        if members:

            response = f"The members of the {department\_display} Department are:\n" + "\n".join(members)

        else:

            response = f"I don't have information about the {department\_display} Department."

        return response

    elif intent == "find\_department\_intent":

        name\_query = data.lower()

        matching\_departments = set()

        matching\_members = []

        for member\_full\_name, deps in member\_to\_departments.items():

            if name\_query in member\_full\_name:

                matching\_departments.update(deps)

                matching\_members.append(member\_full\_name)

        if matching\_departments:

            if len(matching\_departments) == 1:

                department\_display = format\_department\_name(next(iter(matching\_departments)))

                response = f"The name '{data}' belongs to the {department\_display} Department."

            else:

                departments\_display = ", ".join([format\_department\_name(dep) for dep in sorted(matching\_departments)])

                response = f"The name '{data}' belongs to the following departments: {departments\_display}."

            return response

        else:

            return f"Sorry, We still trying to find data for the question you rised. Please fill the form to token rise and try again later. Please fill this form to rise token https://forms.gle/vq7AgNXznn8rxrALA "

    else:

        return "I'm sorry, I didn't understand that. Could you please rephrase? if you are still not getting satisfied answer, please fill this form to rise the token https://forms.gle/vq7AgNXznn8rxrALA "

def chatbot\_logic(input\_text):

    input\_text\_lower = input\_text.lower().strip()

    matched\_tags = set()

    for intent in intents['intents']:

        for pattern in intent['patterns']:

            regex = re.compile(r'\b' + re.escape(pattern.lower()) + r'\b')

            if regex.search(input\_text\_lower):

                matched\_tags.add(intent['tag'])

    if matched\_tags:

        responses = []

        for tag in matched\_tags:

            for intent in intents['intents']:

                if intent['tag'] == tag:

                    responses.append(random.choice(intent['responses']))

                    break

        combined\_response = "\n\n".join(responses)

        return combined\_response

    intent, data = get\_intent(input\_text)

    response = respond(intent, data)

    if intent != "unknown\_intent":

        return response

    input\_vector = vectorizer.transform([input\_text])

    predicted\_tag = clf.predict(input\_vector)[0]

    for intent in intents['intents']:

        if intent['tag'] == predicted\_tag:

            return random.choice(intent['responses'])

    return "I'm sorry, I didn't understand that. Can you please rephrase?"

import streamlit as st

st.title("RIT Student Assistance Chatbot")

# user\_input = st.text\_input("The RIT Student Assistance Chatbot is designed to help students with quick answers to common inquiries about academics, campus services, and more. It provides 24/7 support for tasks like finding information on courses, events, and campus resources, making student life easier and more efficient.")

st.write("The RIT Student Assistance Chatbot is designed to help students with quick answers to common inquiries about academics, campus services, and more. It provides 24/7 support for tasks like finding information on courses, events, and campus resources, making student life easier and more efficient.")

st.write("")  # This adds a horizontal rule (break line)

user\_input = st.text\_input("Type your message here:")

if user\_input:

    response = chatbot\_logic(user\_input)

    st.write(response)