import telebot

import logging

import re

*# Замените 'YOUR\_BOT\_TOKEN' на токен вашего бота*

BOT\_TOKEN = '7318292371:AAFR19d8hjgSwzW-XMxip\_xZ2pDYQCPH0YQ'

bot = telebot.TeleBot(BOT\_TOKEN)

*# Настройка логирования (опционально)*

logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')

*# Информация о колледже (словарь)*

college\_info = {

    "name": "Приемной комиссии Слободского колледжа педагогики и социальных отношений",

    "address": "Российская Федерация, Кировская область, г. Слободской, ул. Рождественская, д.69",

    "phone": "+7 (83362) 4-71-54",

    "website": "https://slobkoll.ru",

    "email" : "Priem@slobkoll.ru",

    "specialities": {

        "Специальности":"",

        "44.02.01 Дошкольное образование": "Воспитатель детей дошкольного возраста. Бюджет. 3 года 10 мес.",

        "44.02.02 Преподавание в начальных классах": "Учитель начальных классов. Бюджет. 3 года 10 мес.",

        "44.02.03 Педагогика дополнительного образования": "Педагодополнительногообразования в области социально педагогической деятельности. Бюджет. 3 года 10 мес.",

        "44.02.04 Специальное Дошкольное образование": "Воспитатель детей дошкольного возраста с отклонениями в развитии и с сохранным развитием. Бюджет. 3 года 10 мес.",

        "49.02.01 Физическая культура": "Учитель физической культуры. Бюджет. 3 года 10 мес.",

        "09.02.07 Информационные системы и программирование": "Программист. Бюджет. 3 года 10 мес.",

        "40.02.04 Юриспруденция": "Юрист. Бюджет. 2 года 10 мес.",

    },

    "admission\_info": "График работы приемной комиссии: \n ПН–ПТ с 9:00 до 16:00 \n СБ, ВС – выходные дни",

    "faq": {

        "\nДолжен ли родитель присутствовать при подаче докуменов абитуриентом?": "Да, конечно. Если на момент подачи документов, абитуренту нет 18 лет",

        "\nНужно ли оставлять оригинал аттестата сразу?": "Нет, но если к концу отбора его не будет, то абитуриент автоматически не проходит",

        "\nНужно ли будет проходить вступительные испытания?": "Для педагогических специальностей, да!. Для физической культуры также обязательно, но при наличии значков ГТО абитуриент освобождается от испытаний. Для программистов вступительные испытания не предусмотрены.",

        "\nКакие документы нужны для поступления?": "Паспорт, аттестат, медицинская справка, снилс."

    },

    "contacts": {

        "Приемная комиссия": {"phone": "+7 (83362) 4-71-54", "email": "Priem.slobkoll.ru"},

        "Директор": {"phone": "8(83362) 4-18-80, 8953-698-01-00"},

        "Очное отделение" :{"phone": "+79536951234", "email":"zavoo@slobkoll.ru"},

        "Заочное отделение" :{"phone": "+79536918444", "email":"zavzao@slobkoll.ru"},

    }

}

*# --- Обработчики команд ---*

@bot.message\_handler(commands=['start'])

**def** start\_command(message):

    keyboard = telebot.types.ReplyKeyboardMarkup(resize\_keyboard=True)

    buttons = ['/Информация', '/График работы', '/Специальности', '/Вопросы', '/Контакты']

    keyboard.add(\*[telebot.types.KeyboardButton(button) for button in buttons])

    bot.send\_message(message.chat.id, **f**"Привет! Я бот {college\_info['name']}. Чем могу помочь?", reply\_markup=keyboard)

@bot.message\_handler(commands=['Информация'])

**def** info\_command(message):

    info\_text = **f**"\*{college\_info['name']}\*\n" \

**f**"Адрес: {college\_info['address']}\n" \

**f**"Телефон: {college\_info['phone']}\n" \

**f**"Почта: {college\_info['email']}\n"  \

**f**"Сайт: [{college\_info['website']}]({college\_info['website']})\n"

    bot.send\_message(message.chat.id, info\_text, parse\_mode="Markdown")

@bot.message\_handler(commands=['График'])

**def** admission\_command(message):

    bot.send\_message(message.chat.id, college\_info['admission\_info'], parse\_mode="Markdown")

@bot.message\_handler(commands=['Специальности'])

**def** specialities\_command(message):

    specialities\_text = "\n".join([**f**"\*{name}\*: {description}" for name, description in college\_info['specialities'].items()])

    bot.send\_message(message.chat.id, specialities\_text, parse\_mode="Markdown")

@bot.message\_handler(commands=['Вопросы'])

**def** faq\_command(message):

    faq\_text = "\n".join([**f**"\*{question}\*: {answer}" for question, answer in college\_info['faq'].items()])

    bot.send\_message(message.chat.id, faq\_text, parse\_mode="Markdown")

@bot.message\_handler(commands=['Контакты'])

**def** contacts\_command(message):

    contacts\_text = **f**"\*Приемная комиссия:\*\nТелефон: {college\_info['contacts']['Приемная комиссия']['phone']}\nEmail: {college\_info['contacts']['Приемная комиссия']['email']}\n\n\*Директор:\*\nТелефон: {college\_info['contacts']['Директор']['phone']}\n\n\*Учебная часть (Очное отделение):\*\nТелефон: {college\_info['contacts']['Очное отделение']['phone']}\nEmail: {college\_info['contacts']['Очное отделение']['email']}\n\n\*Учебная часть (Заочное отделение):\*\nТелефон: {college\_info['contacts']['Заочное отделение']['phone']}\nEmail: {college\_info['contacts']['Заочное отделение']['email']}"

    bot.send\_message(message.chat.id, contacts\_text, parse\_mode="Markdown")

*# Обработчик текстовых сообщений*

@bot.message\_handler(func=**lambda** message: True)

**def** handle\_text(message):

    question = message.text.lower()

*# Регулярные выражения для поиска ключевых слов*

    patterns = {

**r**"\b(привет)": "Привет! Я бот Приемной комиссии от Слободского коллелджа педагогики и социальных отношений, чем могу помочь?",

**r**"\b(специальност(и|ей|ями)?|специальность)\b(?:\s+\b\w+\b){0,5}": "\n".join([**f**"\*{name}\*: {description}" for name, description in college\_info['specialities'].items()]),

**r**"\b(адрес(а|у)?|адрес)\b(?:\s+\b\w+\b){0,5}": college\_info['address'],

**r**"\b(телефон(а|у)?|телефон|номер)\b(?:\s+\b\w+\b){0,5}": college\_info['phone'],

**r**"\b(сайт(а)?|сайт|веб-сайт)\b(?:\s+\b\w+\b){0,5}": **f**"[{college\_info['website']}]({college\_info['website']})",

**r**"\b(процесс|поступлени[еяь]|поступа[тьт]|поступить|какие|экзамен(ы|а|ов)?)\b(?:\s+\b\w+\b){0,5}": "Подать документы можно несколькими способами.\n Самый простой способ, прийти очно в приемную комиссию и подать документы лично.\n Также предоставить документы возможно и через Портал Госуслуги, но в любом случае оригиналы документов нужно будет предоставить до конца работы приемной комиссии, такое возможно через МФЦ, Почту России или принести лично",

**r**"\b(проживани[ея]?|студент(ов|а|ы)?|общежити[ея]?)\b(?:\s+\b\w+\b){0,5}": "Нуждающимся иногородним студентам очного отделения предоставляется общежитие. Общежитие расчитано на 165 мест, 5 этажей",

**r**"\b(документ(ы|а|ов)?|зачислен(ия|ий|ию)?|необходимые\s+документы)\b(?:\s+\b\w+\b){0,5}": "Список документов для зачисления: Паспорт РФ, аттестат (2 страницы), медицинская справка(086-У), снилс.",

**r**"\b(спортивн(ые|ой|ого|ыми|ым)?|секци(и|й|ей|ями|ям)|кружк(и|а|ов|е|ой|у)?|спорт|секции|кружки)\b(?:\s+\b\w+\b){0,5}": "Информация о спортивных секциях и кружках: Волейбол, Баскетбол, Теннис(настольный) Танцы, Фитнес.",

**r**"\b(трудоустройств(о|а|у)|выпускник(ов|а|и|е|ой|у)?|работа|трудоустройство|карьера)\b(?:\s+\b\w+\b){0,5}": "Колледжом учрежден Центр карьеры (Центр содействия трудоустройству выпускников).",

**r**"\b(перевод|из|друг(ого|их|ое|ую|их)|учебн(ого|ых|ое|ую|ыми|ым)?|заведени[еяь]?|перевестись)\b(?:\s+\b\w+\b){0,5}": "При наличии вакантных бюджетных мест зачисление осуществляется на первый курс на базе основного общего образования и переводом на Индивидуальный учебный план (второй курс) на основании документов, подтверждающих соответствующий уровень образования.",

**r**"\b(инфраструктур(а|ы|е|ой|у)|библиотека|лаборатор(ия|ии|ий|иям|иями)|аудитор(ия|ии|ий|иям|иями)?|здание|кампус|оборудование)\b(?:\s+\b\w+\b){0,5}": "3 учебных здания, библиотека, столовая, спортзал. 4 мастерские",

**r**"\b(языков(ые|ого|ой|ыми|ым)?|курс(ы|а|ов)?|язык|английский|немецкий)\b(?:\s+\b\w+\b){0,5}": "В колледже предусмотрено изучение 2 языков: английский и немецкий",

}

    for pattern, answer in patterns.items():

        if re.search(pattern, question):

            bot.send\_message(message.chat.id, answer, parse\_mode="Markdown")

            return  *# Прерываем цикл после нахождения совпадения*

*# Проверка вопросов из FAQ*

    for q, a in college\_info['faq'].items():

        if q.lower() in question:

            bot.send\_message(message.chat.id, a, parse\_mode="Markdown")

            return

*# Если ни один паттерн не совпал*

    bot.send\_message(message.chat.id, "Я пока не знаю ответа на этот вопрос. Попробуйте воспользоваться командами /Информация, /График, /Специальности, /Вопросы, или /Контакты")

*# Запуск бота*

bot.polling()