**1. Определение задач чат-бота:** Определите конкретные задачи, которые должен решать чат-бот (например, ответы на вопросы по учебному плану, помощь в поиске информации в учебных материалах, напоминания о сроках сдачи работ, предоставление информации о преподавателях).

**2. Выбор языковой модели:** Выберите подходящую языковую модель (например, RuGPT-3, BERT, Transformers) и API для работы с ней.

**3. Разработка базы знаний:** Создайте базу знаний для чат-бота, содержащую ответы на типичные вопросы учащихся и справочную информацию.

**4. Программирование чат-бота:** Разработайте код чат-бота на Python, используя выбранную языковую модель и базу знаний. Обеспечьте возможность интерактивного взаимодействия с пользователем.

**5. Тестирование и отладка:** Протестируйте чатбота на различных сценариях использования и отладьте его работу.

**6. Интеграция (опционально):** Интегрируйте чатбота в существующую образовательную платформу или мессенджер.

**Требования:** Четкое определение задач чат-бота и целевой аудитории. Выбор подходящей языковой модели и обоснование ее выбора. Создание базы знаний объемом не менее 50 вопросов/ответов. Рабочий прототип чат-бота на Python с возможностью интерактивного взаимодействия. Документация к разработанному коду. Отчет о тестировании и отладке чат-бота. Аналитический отчет, включающий описание всех этапов работы, использованные технологии и результаты тестирования.

import telebot

from telebot import types

from request import gpt\_reguest

def giga(message):

    bot.send\_message(message.chat.id,gpt\_reguest(message.text))

    return

bot = telebot.TeleBot(open('api.txt').read()) #Инициализация бота

btn1 = types.KeyboardButton('Расписание') #создание кнопки

btn2 = types.KeyboardButton('ДЗ')

btn3 = types.KeyboardButton('Информация о школе')

btn4 = types.KeyboardButton('Успеваемость')

btn5 = types.KeyboardButton('Фото')

btn6 = types.KeyboardButton('Вопрос GigaChat')

markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)#сетка для кнопок

markup.add(btn1,btn2,btn3,btn4,btn5,btn6)

@bot.message\_handler(commands=['start']) #определение реакции бота на /start

def send\_wecome(message): #функци реакции на /start

    bot.reply\_to(message, "Привет! Я учебный бот") #тело программы

    bot.send\_message(message.chat.id, "Выберите действие:",

                     reply\_markup=markup)

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

def handle\_buttons(message):

    if message.text == 'Расписание':

        ph=open('raspisanie\_23.jpg', 'rb') # путь к фото, тип чтения, rb - читать файл

        bot.send\_photo(message.chat.id,ph, 'Расписание') # отзыв бота на "Расписание", отправляет фото

        ur1 = 'https://school-5.ucoz.org/P3/11.png'

        bot.send\_photo(message.chat.id,ur1, 'Расписание')

        #bot.reply\_to(message, "Сейчас лето, занятий нет.")

        #inline\_markup = types.InlineKeyboardMarkup() #создание "шаблона" для инлайн копки

        #btn = types.InlineKeyboardButton( #текст и ссылка для кнопки и её инициализация

            #text="Летние активности",

            #url="https://leto.mos.ru/"

        #)

        #inline\_markup.add(btn) #добавление кнопки в "шаблон" для инлайн кнопок

        #bot.send\_message(message.chat.id, "Лучше посмотри летние активности",

                         #reply\_markup=inline\_markup) #отправка сообщения пользователю с кнопкой

    elif message.text == 'ДЗ':

        doc = open('Это домашнее задание.pdf','rb')

        bot.send\_document(message.chat.id, doc, caption='ДЗ',

                          visible\_file\_name='Файл ДЗ.pdf')

        #bot.reply\_to(message, "У вас каникулы, а всё ДЗ у учителей.")

    elif message.text == 'Информация о школе':

        bot.reply\_to(message, "https://sch1420uv.mskobr.ru/")

    elif message.text == 'Успеваемость':

        bot.reply\_to(message, "https://school.mos.ru/")

    elif str(message.text).lower() == "привет":

        bot.reply\_to(message, "Привет! Я учебный бот.") # тело программы

    elif message.text == 'Фото':

       try:

        ph = open('name\_file.jpg', 'rb')

        bot.send\_photo(message.chat.id,ph, 'Ваше фото')

       except BaseException:

        bot.reply\_to(message, "Фото отсутствует, отправьте новое!")

    elif message.text == 'Вопрос GigaChat':

       msg = bot.reply\_to(message, 'Напиши текст запроса для языковой модели')

       bot.register\_next\_step\_handler(msg, giga)

@bot.message\_handler(content\_types=['photo']) # получать фото от пользователя

def photo(message):

    file\_id = message.photo[-1].file\_id # из полученного сообщения берем фото

    # ИД хранится в последнем элементе с помощью обращения к нему мы получаем file id

    file\_info = bot.get\_file(file\_id) # получение информации о самом файле по его ID

    download\_file = bot.download\_file(file\_info.file\_path) # загрузка файла в оперативную память

    with open('name\_file.jpg', 'wb') as new\_f:

        new\_f.write(download\_file)

    bot.reply\_to(message, 'Фото сохранено') # отправка сообщения пользователю

bot.polling() #отправка "настроек" в бот и его активация.

# Без него бот неактивен