МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Кемеровский государственный университет»**

**Институт фундаментальных наук**

**Кафедра ЮНЕСКО по ИВТ**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**ПО ДИСЦИПЛИНЕ “Автоматизация процессов развертывания и инсталляции информационных систем”**

студента 2 курса

Башкеева Степана Дмитриевича

Направление 09.03.03 – Прикладная информатика в экономике

Преподаватель:

И.Ю. Степанов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена:

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_202\_г.

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_

Кемерово 2022

**Содержание**

[Цель работы 2](#_Toc2071650587)

[Практическая часть 2](#_Toc1652626438)

[Вывод 6](#_Toc1609541928)

[Список литературы 6](#_Toc370945784)

[Приложение 6](#_Toc491686971)

# **Цель работы**

Написать бота для Telegram на python и разместить его на облачной платформе pythonanywhere.

# **Практическая часть**

1. Написать бота, который будет выводить ссылки на популярные сферы IT.

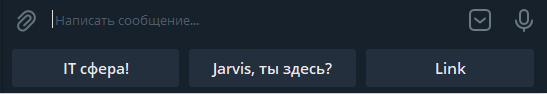


Рисунок 1. По нажатию кнопки “IT сфера” происходит переход на следующее меню.

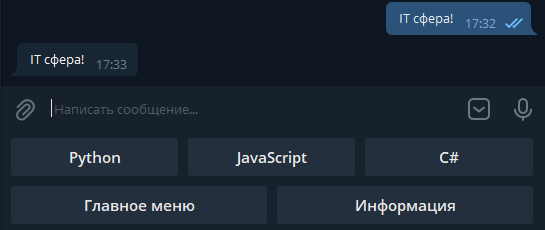


Рисунок 2. Меню с кнопками IT сферы.

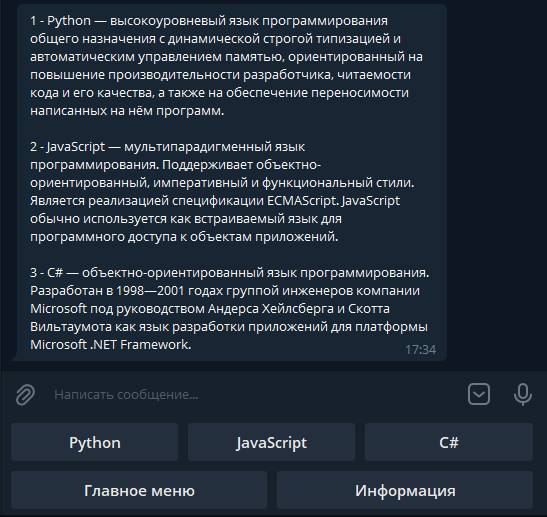


Рисунок 3. По нажатию кнопки “Информация” выводится общая информация об сайтах.



Рисунок 4. По нажатию кнопки “Python” выводится ссылка на сайт Python.org, по нажатию кнопки “JavaScript” выводится ссылка на сайт learn.JavaScript.ru.

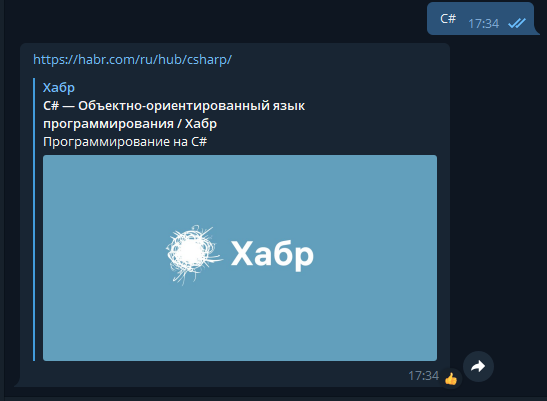


Рисунок 5. По нажатию кнопки “C#” выводится ссылка на сайт Csharp.

1. Написать бота, который будет реагировать на голосовые сообщения.

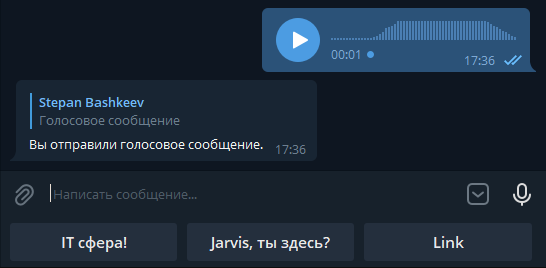


Рисунок 6. При отправке боту голосового сообщения он выводит сообщение “Вы отправили голосовое сообщение.”

1. Написать бота, который выведет введенное ему сообщение.

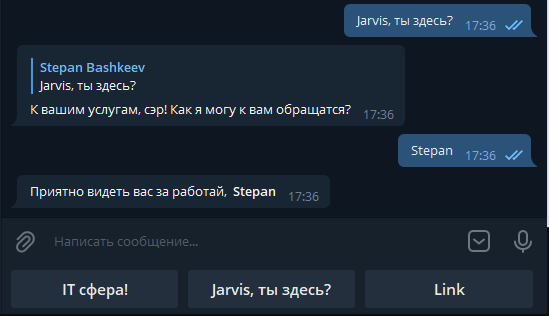


Рисунок 7. По нажатию кнопки “Jarvis, ты здесь?”, бот запросит и выведет ваше имя (или то, что вы введёте).

1. Написать команду “/close”, которая будет закрывать клавиатуру.



Рисунок 8. При вводе команды “/close” бот закрывает клавиатуру.

1. Написать бота, который отправит ответ сервера.

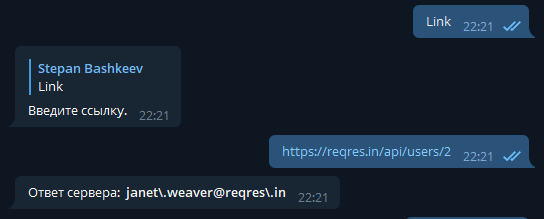


Рисунок 9. По нажатию кнопки “Link” бот запросит ссылку, после того как вы ввели ссылку бот выводит ответ сервера (в ином случаи бот выведет “Ошибка”).

1. Размещение бота на pythonanywhere.
2. Регистрируемся в pythonanywhere.
3. Заходим в пункт Files, создаем новую папку (например: bot)
4. Загружаем все файлы, связанные с ботом.
5. Переходим в пункт Consoles, открываем “Bash console”
6. В открывшейся консоле нужно установить все библиотеки, которые были использованы при создании бота (например: pip install aiogram)
7. В конце запускаем основной файл через консоль или открываем файл с ботом и запускаем нажав на кнопку “Run”.
8. Бот запущен.

# **Вывод**

В результате работы был создан бот для Telegram и размещен на облачной платформе pythonanywhere.

# Список литературы

1. Документация библиотеки aiogram - https://docs.aiogram.dev/en/latest/

# Приложение

from aiogram.dispatcher import FSMContext  
from aiogram.dispatcher.filters import Text  
from aiogram.contrib.fsm\_storage.memory import MemoryStorage  
from aiogram.types import ParseMode  
from aiogram.utils import executor  
  
import config  
import logging  
import buttons as btn  
import aiogram.utils.markdown as md  
import requests as req  
import aiohttp  
import asyncio  
  
from aiogram import Bot, Dispatcher, executor, types  
  
from aiogram.dispatcher.filters.state import State, StatesGroup  
  
from filters import IsAdminFilter  
#import infomations text  
from info import info\_lang, py\_link, js\_link, c\_link  
  
#log level  
logging.basicConfig(level=logging.INFO)  
  
#bot init  
bot = Bot(token=config.TOKEN)  
dp = Dispatcher(bot, storage=MemoryStorage())  
storage = MemoryStorage()  
  
#activate filters  
dp.filters\_factory.bind(IsAdminFilter)  
  
#class State  
class Mudialog(StatesGroup):  
 res\_user = State()  
  
#start bot message  
@dp.message\_handler(commands=['start'])  
async def command\_start(message: types.Message):  
 await bot.send\_message(message.from\_user.id, 'Hello {0.first\_name}'.format(message.from\_user), reply\_markup=btn.mainMenu)  
  
#remove keyboard  
@dp.message\_handler(commands=['close'])  
async def command\_start(message: types.Message):  
 await bot.send\_message(message.from\_user.id, 'Клавиатура закрыта!', reply\_markup=types.ReplyKeyboardRemove())  
  
  
class Form(StatesGroup):  
 name = State()  
 link = State()  
  
  
  
#buttons  
@dp.message\_handler()  
async def bot\_message(message: types.Message):  
 if message.text == 'IT сфера!':  
 await bot.send\_message(message.from\_user.id, 'IT сфера!', reply\_markup=btn.otherMenu)  
  
 elif message.text == 'Главное меню':  
 await bot.send\_message(message.from\_user.id, 'Главное меню', reply\_markup=btn.mainMenu)  
  
 elif message.text == 'Python':  
 await bot.send\_message(message.from\_user.id, py\_link)  
  
 elif message.text == 'JavaScript':  
 await bot.send\_message(message.from\_user.id, js\_link)  
  
 elif message.text == 'C#':  
 await bot.send\_message(message.from\_user.id, c\_link)  
  
 elif message.text == 'Информация':  
 await bot.send\_message(message.from\_user.id, info\_lang)  
  
 elif message.text == 'Jarvis, ты здесь?':  
 await Form.name.set()  
 await message.reply("К вашим услугам, сэр! Как я могу к вам обращатся?")  
  
 elif message.text == 'Link':  
 await Form.link.set()  
 await message.reply('Введите ссылку.')  
  
  
@dp.message\_handler(state='\*', commands='cancel')  
@dp.message\_handler(Text(equals='отмена', ignore\_case=True), state='\*')  
async def cancel\_handler(message: types.Message, state: FSMContext):  
 current\_state = await state.get\_state()  
 if current\_state is None:  
 return  
  
 await state.finish()  
 await message.reply('ОК')  
  
#input name  
@dp.message\_handler(state=Form.name)  
async def process\_name(message: types.Message, state: FSMContext):  
 async with state.proxy() as data:  
 data['name'] = message.text  
  
 async with state.proxy() as data:  
  
 await bot.send\_message(  
 message.chat.id,  
 md.text(  
 md.text('Приятно видеть вас за работай, ', md.bold(data['name'])),  
 sep='\n',  
 ),  
 parse\_mode=ParseMode.MARKDOWN,  
 )  
  
 await state.finish()  
  
#outlink  
@dp.message\_handler(state=Form.link)  
async def process\_link(message: types.Message, state: FSMContext):  
 async with state.proxy() as data:  
 try:  
 lk = req.get(message.text)  
 data['link'] = lk.json()['data']['email']  
 except:  
 data['link'] = 'Ошибка'  
  
 async with state.proxy() as data:  
  
 await bot.send\_message(  
 message.chat.id,  
 md.text(  
 md.text('Ответ сервера: ', md.bold(data['link'])),  
 sep='\n',  
 ),  
 parse\_mode=ParseMode.MARKDOWN,  
 )  
  
 await state.finish()  
#voice message  
@dp.message\_handler(content\_types=['voice'])  
async def message\_voice\_handler(message: types.Message):  
 await message.reply("Вы отправили голосовое сообщение.")  
  
#run long-polling  
if \_\_name\_\_ == "\_\_main\_\_":  
 executor.start\_polling(dp, skip\_updates=True)