



**Министерство науки и высшего образования
Российской Федерации Федеральное государственное
бюджетное образовательное учреждение высшего
образования
«Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский университет
МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и
управления»**

Лабораторная работа №6
«Разработка бота на основе конечного автомата для
Telegram с использованием языка Python»
по предмету
«Базовые компоненты интернет-технологий»

Выполнил:
студент группы № ИУ5-33Б
Пермяков Дмитрий

Проверил:
Преподаватель кафедры ИУ-5
Гапанюк Юрий

2022 г.

Постановка задачи.

Разработайте бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

Код программы:

Main.py

```
# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>

from aiogram.utils import executor
from create_bot import dp
from aiogram import types

async def on_startup(_):
    print('The bot was included.')

async def handler_start(message: types.Message):
    await message.answer('Привет!')

if __name__ == '__main__':
    from handlers import operations

    operations.register_handlers(dp)
    executor.start_polling(dp, skip_updates=True, on_startup=on_startup)
```

Creat_bot.py

```
# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>
from aiogram import Bot, Dispatcher
from aiogram.contrib.fsm_storage.memory import MemoryStorage
from dotenv import load_dotenv
import os

load_dotenv()
TOKEN = os.getenv('TOKEN')

bot = Bot(TOKEN)
storage = MemoryStorage()
dp = Dispatcher(bot, storage=MemoryStorage())
```

Utils.db_management.py

```
# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>

import sqlite3
from random import choice

def update_name(id_user: int, new_name: str, filename: str = 'data/users.old.db') -> None:
    db = sqlite3.connect(filename)
    cursor = db.cursor()
    cursor.execute(f"UPDATE users SET name = '{new_name}' WHERE id_user = {id_user}")
    db.commit()
    db.close()
```

```

def get_user_data(id_user: int, filename: str = 'data/users_old.db') -> tuple
| None:
    db = sqlite3.connect(filename)
    cursor = db.cursor()
    cursor.execute(f"SELECT * FROM users WHERE id_user = {id_user}")
    res = cursor.fetchone()
    db.commit()
    db.close()
    return res

def get_random_user_form(id_user: int, filename: str = 'data/users_old.db')
-> tuple:
    db = sqlite3.connect(filename)
    cursor = db.cursor()
    cursor.execute(f"SELECT id_user FROM users WHERE id_user <> {id_user}")
    id_user: int = choice([el[0] for el in cursor.fetchall()])
    res = get_user_data(id_user, filename)
    db.commit()
    db.close()
    return res

def set_user_data(id_user: int, data: dict, filename: str = 'data/
users_old.db'):
    db = sqlite3.connect(filename)
    cursor = db.cursor()
    cursor.execute(
        "INSERT INTO users VALUES ({} , '{}', {}, '{}', '{}', '{}', '{}', '{}',
        '{}')".format(id_user, data['name'],
    data['age'],
    data['place'],
    data['university'],
    data['department'],
    data['description'],
    data['photo']))
    db.commit()
    db.close()

def update_user_data(id_user: int, data: dict, filename: str = 'data/
users_old.db'):
    db = sqlite3.connect(filename)
    cursor = db.cursor()
    cursor.execute(
        "UPDATE users SET name = '{}', age={}, place = '{}', university =
        '{}', department = '{}', description = '{}'",
        "photo = '{}' WHERE id_user = {}".format(data['name'], data['age'],
    data['place'], data['university'],
        data['department'],
        data['description'],
        data['photo'],
        id_user))

    db.commit()
    db.close()

```

```

def update_user_description(id_user: int, description: str, filename: str = 'data/users_old.db'):
    db = sqlite3.connect(filename)
    cursor = db.cursor()
    cursor.execute(f"UPDATE users SET description = '{description}' WHERE id_user = {id_user}")
    db.commit()
    db.close()

def update_user_photo(id_user: int, photo: str, filename: str = 'data/users_old.db'):
    db = sqlite3.connect(filename)
    cursor = db.cursor()
    cursor.execute(f"UPDATE users SET photo = '{photo}' WHERE id_user = {id_user}")
    db.commit()
    db.close()

def user_presents(id_user: int, filename: str = 'data/users_old.db') -> bool:
    """Checking, is user in data or is not.
    :param id_user: id of the checking user.
    :param filename: way and name for file.
    :return: user status True/False.
    """
    db = sqlite3.connect(filename)
    cursor = db.cursor()
    cursor.execute(f"SELECT id_user FROM users WHERE id_user = {id_user}")
    lst = cursor.fetchall()
    db.commit()
    db.close()
    return False if len(lst) == 0 else True

```

Keyabords.menu_bts.py

```

# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton

start_markup = ReplyKeyboardMarkup(resize_keyboard=True)
buttons = (
    'Заполнить анкету заново',
    'Изменить фото',
    'Изменить текст анкеты',
    'Искать друзей 🤝'
)

start_markup.row(*[el for el in buttons[:2]]).row(*[el for el in buttons[2:4]])

```

Keyabords.search_lib.py

```

# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>
from aiogram.types import InlineKeyboardButton, InlineKeyboardMarkup

search_markup = InlineKeyboardMarkup(resize_keyboard=True)
buttons = (
    ('👍', 'like'),
    ('📝', 'write'),
    ('👎', 'dislike'),
)

```

```

        ('zz', 'info')
    )
search_markup.row(*([InlineKeyboardButton(text, callback_data=data) for text,
data in buttons)])

```

Handlers.operations

```

# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>
from aiogram import Dispatcher, types
from aiogram.dispatcher.filters.state import State, StatesGroup
from aiogram.dispatcher import FSMContext
import aiogram.utils.markdown as md
from keyboards.menu_bts import start_markup
from keyboards.search_ib import search_markup
from utils.db_management import *
from create_bot import dp


class FSMUsers(StatesGroup):
    id_user = State()
    photo = State()
    name = State()
    age = State()
    place = State()
    university = State()
    department = State()
    description = State()


class FSMChangeUserData(StatesGroup):
    photo = State()


async def handler_start(message: types.Message):
    # Если пользователь уже зарегистрирован.
    user_data = get_user_data(message.chat.id)
    if user_data is not None:
        text = "Класс! Вы уже зарегистрированы!"
        await message.answer(text=text)
        await message.answer_photo(
            photo=user_data[7],
            caption=md.text(
                md.text('Так выглядит Ваша анкета:\n'),
                md.text(f'{user_data[1]}, {user_data[2]}, {user_data[3]}'),
                md.text(f'{user_data[4]}, {user_data[5]}'),
                md.text('О себе:'),
                md.text(user_data[6]),
                sep='\n',
            )
        )
        await message.answer('Выберите дальний сценарий:',
reply_markup=start_markup)

    else:
        await message.answer("О, новичок!\nДавайте Вас зарегистрируем!")
        await FSMUsers.photo.set()
        await message.answer("Отправьте ваше фото:")

async def catch_photo(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['photo'] = message.photo[0].file_id

    # print(len(data))

```

```

    await FSMUsers.next()
    await message.reply('Теперь введите Ваше имя:')

async def catch_name(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['name'] = message.text
    await FSMUsers.next()
    await message.reply('Введите ваш возраст:')

async def catch_age(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['age'] = message.text
    await FSMUsers.next()
    await message.reply('Введите ваш город:')

async def catch_place(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['place'] = message.text
    await FSMUsers.next()
    await message.reply('Введите название вашего университета:')

async def catch_university(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['university'] = message.text
    await FSMUsers.next()
    await message.reply('Введите название вашего факультета:')

async def catch_department(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['department'] = message.text
    await FSMUsers.next()
    await message.reply('Расскажите что-то о себе:')

async def catch_description(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['description'] = message.text
    async with state.proxy() as data:
        try:
            await message.answer_photo(
                photo=data['photo'],
                caption=md.text(
                    md.text('Так выглядит Ваша анкета:\n'),
                    md.text(f'{data["name"]}, {data["age"]},\n{data["place"]}'),
                    md.text(f'{data["university"]}, {data["department"]}\n'),
                    md.text('О себе:'),
                    md.text(data['description']),
                    sep='\n',
                )
            )
        # Добавляет в SQL таблицу.
        if user_presents(message.chat.id):
            update_user_data(message.chat.id, dict(data))
        else:
            set_user_data(message.chat.id, dict(data))
    except KeyError:
        temp_id = message.chat.id
        update_user_description(temp_id, message.text)
        new_data = get_user_data(temp_id)
        await message.answer_photo(

```

```

photo=new_data[7],
caption=md.text(
    md.text('Так выглядит Ваша анкета:\n'),
    md.text(f'{new_data[1]}, {new_data[2]}, {new_data[3]}'),
    md.text(f'{new_data[4]}, {new_data[5]}'),
    md.text('О себе:'),
    md.text(new_data[6]),
    sep='\n',
)
)
await state.finish()
await message.answer('Выберите новый сценарий:', reply_markup=start_markup)

async def all_msg_handler(message: types.Message):
    button_text = message.text

    if button_text == 'Заполнить анкету заново':
        await FSMUsers.photo.set()
        await message.answer("Отправьте ваше фото:", reply_markup=types.ReplyKeyboardRemove())

    elif button_text == 'Изменить фото':
        await FSMChangeUserData.photo.set()
        await message.reply("Отправьте новое фото:", reply_markup=types.ReplyKeyboardRemove())

    elif button_text == 'Изменить текст анкеты':
        await FSMUsers.description.set()
        await message.reply("Расскажите что-то о себе:", reply_markup=types.ReplyKeyboardRemove())

    elif button_text == 'Искать друзей 🤝':
        user_data = get_random_user_form(message.chat.id)
        await message.answer('Поиск друзей!', reply_markup=types.ReplyKeyboardRemove())
        await message.delete()
        await message.answer_photo(
            photo=user_data[7],
            caption=md.text(
                md.text(f'{user_data[1]}, {user_data[2]}, {user_data[3]}'),
                md.text(f'{user_data[4]}, {user_data[5]}'),
                md.text('О себе:'),
                md.text(user_data[6]),
                sep='\n',
            ),
            reply_markup=search_markup
        )
        # await message.reply(reply_text)

    else:
        reply_text = "Keep calm... Everything is fine, you just a silly"
        await message.reply(reply_text)
        await message.delete()

async def update_user_photo_fsm(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['photo'] = message.photo[0].file_id
    temp_id = message.chat.id
    update_user_photo(temp_id, data['photo'])
    new_data = get_user_data(temp_id)
    await message.answer_photo(
        photo=new_data[7],

```

```

caption=md.text(
    md.text('Так выглядит Ваша анкета:\n'),
    md.text(f'{new_data[1]}, {new_data[2]}, {new_data[3]}'),
    md.text(f'{new_data[4]}, {new_data[5]}'),
    md.text('О себе:'),
    md.text(new_data[6]),
    sep='\n',
)
)
await state.finish()
await message.answer('Выберите новый сценарий:',
reply_markup=start_markup)

@dp.callback_query_handler(text=['like', 'write', 'dislike', 'info'])
async def inline_kb_answer_callback_handler(query: types.CallbackQuery):
    answer_data = query.data

    if answer_data == 'like':
        text = 'В разработке!'
        await query.message.answer(text)

    elif answer_data == 'write':
        text = 'В разработке!'
        await query.message.answer(text)

    elif answer_data == 'dislike':
        user_data = get_random_user_form(query.message.chat.id)
        await query.message.answer_photo(
            photo=user_data[7],
            caption=md.text(
                md.text(f'{user_data[1]}, {user_data[2]}, {user_data[3]}'),
                md.text(f'{user_data[4]}, {user_data[5]}'),
                md.text('О себе:'),
                md.text(user_data[6]),
                sep='\n',
            ),
            reply_markup=search_markup
        )

    elif answer_data == 'info':
        user_data = get_user_data(query.message.chat.id)
        await query.message.answer_photo(
            photo=user_data[7],
            caption=md.text(
                md.text('Так выглядит Ваша анкета:\n'),
                md.text(f'{user_data[1]}, {user_data[2]}, {user_data[3]}'),
                md.text(f'{user_data[4]}, {user_data[5]}'),
                md.text('О себе:'),
                md.text(user_data[6]),
                sep='\n',
            )
        )
        await query.message.answer('Выберите дальнейший сценарий:',
reply_markup=start_markup)

    else:
        text = f'Unexpected callback data {answer_data}!'
        await query.message.answer(text)

def register_handlers(dp_main: Dispatcher):
    dp_main.register_message_handler(handler_start, commands=['start'])
    dp_main.register_message_handler(catch_photo, content_types=['photo'],
state=FSMUsers.photo)
    dp_main.register_message_handler(update_user_photo_fsm,

```

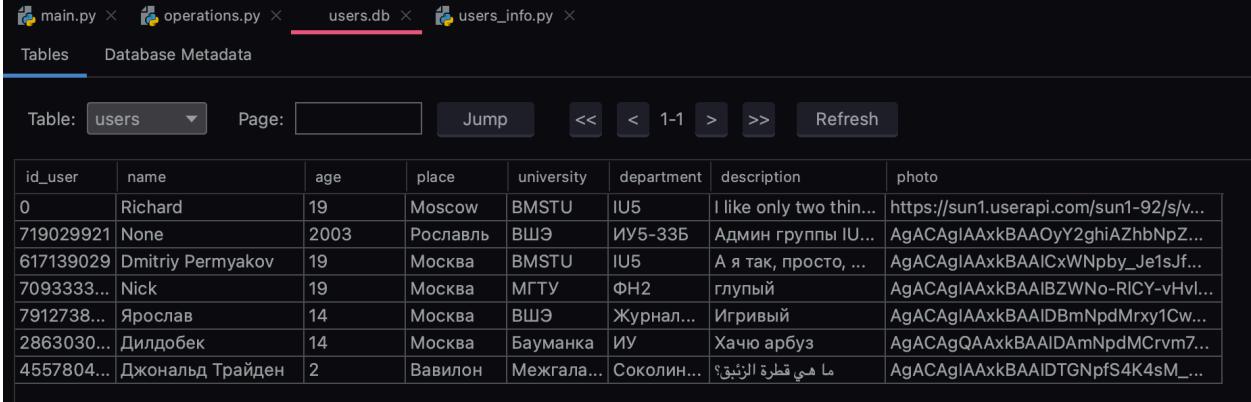
```

content_types=['photo'], state=FSMChangeUserData.photo)
dp_main.register_message_handler(catch_name, state=FSMUsers.name)
dp_main.register_message_handler(catch_age, lambda message:
message.text.isdigit(), state=FSMUsers.age)
dp_main.register_message_handler(catch_place, state=FSMUsers.place)
dp_main.register_message_handler(catch_university,
state=FSMUsers.university)
dp_main.register_message_handler(catch_department,
state=FSMUsers.department)
dp_main.register_message_handler(catch_description,
state=FSMUsers.description)
# dp_main.register_message_handler(inline_kb_answer_callback_handler,
text=['like', 'write', 'dislike', 'info'])
# Last point!! Important!
dp_main.register_message_handler(all_msg_handler)

```

Результат работы программы:

База данных на основе SQLite3



The screenshot shows a SQLite database interface with four tabs at the top: main.py, operations.py, users.db (which is selected and highlighted in red), and users_info.py. Below the tabs, there are two buttons: 'Tables' and 'Database Metadata'. Underneath, there's a search bar labeled 'Table: users' with a dropdown arrow, a page number input field 'Page: []', a 'Jump' button, and navigation buttons for 'Page 1-1' and 'Refresh'. The main area displays a table with the following data:

id_user	name	age	place	university	department	description	photo
0	Richard	19	Moscow	BMSTU	IU5	I like only two thin...	https://sun1.userapi.com/sun1-92/s/v...
719029921	None	2003	Рославль	ВШЭ	ИУ5-335	Админ группы IU...	AgACAgIAAxkBAAOyY2ghiAZhbNpZ...
617139029	Dmitriy Permyakov	19	Москва	BMSTU	IU5	А я так, просто, ...	AgACAgIAAxkBAAICxWNpby_Je1sJf...
7093333...	Nick	19	Москва	МГТУ	ФН2	глупый	AgACAgIAAxkBAAlBZWNo-RICY-vHvI...
7912738...	Ярослав	14	Москва	ВШЭ	Журнал...	Игривый	AgACAgIAAxkBAAIDBmNpdMrxy1Cw...
2863030...	Дилдобек	14	Москва	Бауманка	ИУ	Хачю арбуз	AgACAgQAAxkBAAIDAmNpdMCrvm7...
4557804...	Джональд Трайден	2	Вавилон	Межгала...	Соколин...	ما هي قطرة الزئبق؟	AgACAgIAAxkBAAIDTGNpfS4K4sM_...

Посмотрим результаты выполнения команд в самом телеграмме:

Telegram

Поиск

Все чаты IT 1 Football 3 Cinema

ЦК: Web-разработчик... 23:00
Софья
Понятно, я люблю выполнять... 36

Поток ИУ5 - 2 курс 20... 15:32
Yuriy Gapanyuk
Уважаемые студенты, возможн...

Цифровая кафедра МГТУ и... 23:00
Цифровая кафедра МГТУ им.
Н.Э.Баумана закрепил(а) «...

ISCRA | InfoSec Club "Ra" Вс
Web-безопасность
Существует множество уязвим...

ИУ5 7.11.22
#vk 🎉 Полностью обновили
Образовательный центр VK в Б...

Сибирь 2021 — часть II 11.05.22
Даниил Ляпунов
❗ Напоминаю, что Сибирь сег...

BMSTU Students | Richard бот

Dmitriy Permyakov /start 23:07

BMSTU Students | Richard 23:07
Класс! Вы уже зарегистрированы!



Так выглядит Ваша анкета:

Dmitriy Permyakov, 19, Москва
BMSTU, IU5
О себе:
А я так, просто, Желез.. Админ

BMSTU Students | Richard 23:07
Выберите дальнейший сценарий:

Сообщение...

Заполнить анкету заново Изменить фото

Изменить текст анкеты Искать друзей 🤝

Telegram

Поиск

Все чаты IT 1 Football 3 Cinema

ЦК: Web-разработчик... 23:00
Софья
Понятно, я люблю выполнять... 36

Поток ИУ5 - 2 курс 20... 15:32
Yuriy Gapanyuk
Уважаемые студенты, возможн...

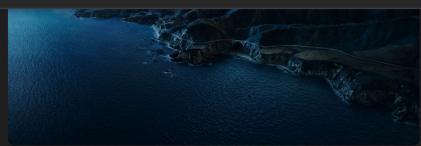
Цифровая кафедра МГТУ и... 23:00
Цифровая кафедра МГТУ им.
Н.Э.Баумана закрепил(а) «...

ISCRA | InfoSec Club "Ra" Вс
Web-безопасность
Существует множество уязвим...

ИУ5 7.11.22
#vk 🎉 Полностью обновили
Образовательный центр VK в Б...

Сибирь 2021 — часть II 11.05.22
Даниил Ляпунов

BMSTU Students | Richard бот



Так выглядит Ваша анкета:

Dmitriy Permyakov, 19, Москва
BMSTU, IU5
О себе:
А я так, просто, Желез.. Админ

BMSTU Students | Richard 23:08
Выберите дальнейший сценарий:

Dmitriy Permyakov 23:08
Изменить фото

BMSTU Students | Richard 23:08
Dmitriy Permyakov
Изменить фото

Отправьте ваше новое фото:

Сообщение...

Telegram

Поиск

Все чаты IT 1 Football 3 Cinema

ЦК: Web-разработчик... 23:00
София
Понятно, я люблю выполнять... 36

Поток ИУБ - 2 курс 20... 15:32
Yuriy Gapanyuk
Уважаемые студенты, возможн...

Цифровая кафедра МГТУ и... 15:32
Цифровая кафедра МГТУ им.
Н.Э.Баумана закрепил(а) «...

ISCRA | InfoSec Club "Ra" 15:32
Web-безопасность
Существует множество уязвим...

ИУБ 7.11.22
#vk 👩‍💻 Полностью обновили
Образовательный центр VK в Б...

Сибирь 2021 – часть II 11.05.22
Даниил Ляпунов
❗ Напоминаю, что Сибирь сег...

BMSTU Students | Richard
бот

Dmitriy Permyakov

I'm a git. Just a git.

- Dmitry student, UI-IT
- 19 y.o. (14.03.2003)
- 19 kg (100 lbs)
- 180 cm (5'11")
- 100% self-taught student
- 100% open source
- 100% Java
- 100% Love coffee.

My favorite projects:

- Educational IT work at BMSTU
- UI-BOT
- UI-IT
- UI-IT
- UI-IT
- UI-IT

My organizations:

BMSTU Students | Richard 23:09

I'm a git. Just a git.

- Dmitry student, UI-IT
- 19 y.o. (14.03.2003)
- 19 kg (100 lbs)
- 180 cm (5'11")
- 100% self-taught student
- 100% open source
- 100% Java
- 100% Love coffee.

My favorite projects:

- Educational IT work at BMSTU
- UI-BOT
- UI-IT
- UI-IT
- UI-IT
- UI-IT

My organizations:

Сообщение...

Заполнить анкету заново Изменить фото

Изменить текст анкеты Искать друзей 🌟

Telegram

Поиск

Все чаты IT 1 Football 3 Cinema

ЦК: Web-разработчик... 23:00
Софья
Понятно, я люблю выполнять... 36

Поток ИУБ - 2 курс 20... 15:32
Yuriy Gapanyuk
Уважаемые студенты, возможн...

Цифровая кафедра МГТУ и... Вт
Цифровая кафедра МГТУ им.
Н.Э.Баумана закрепил(а) «...

ISCRA | InfoSec Club "Ra" Вс
Web-безопасность
Существует множество уязвим...

ИУБ 7.11.22
#vk #Полностью обновили
Образовательный центр VK в Б...

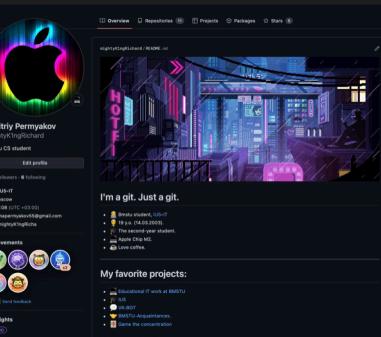
Сибирь 2021 – часть II 11.05.22
Даниил Ляпунов
! Напоминаю, что Сибирь сег...

BMSTU Students | Richard бот
Dmitriy Permyakov
Изменить текст анкеты 23:09

BMSTU Students | Richard
Dmitriy Permyakov
Изменить текст анкеты 23:09
Расскажите что-то о себе:

Dmitriy Permyakov
Новый текст анкеты для тестирования программы 23:09

BMSTU Students | Richard 23:09



Так выглядит Ваша анкета:

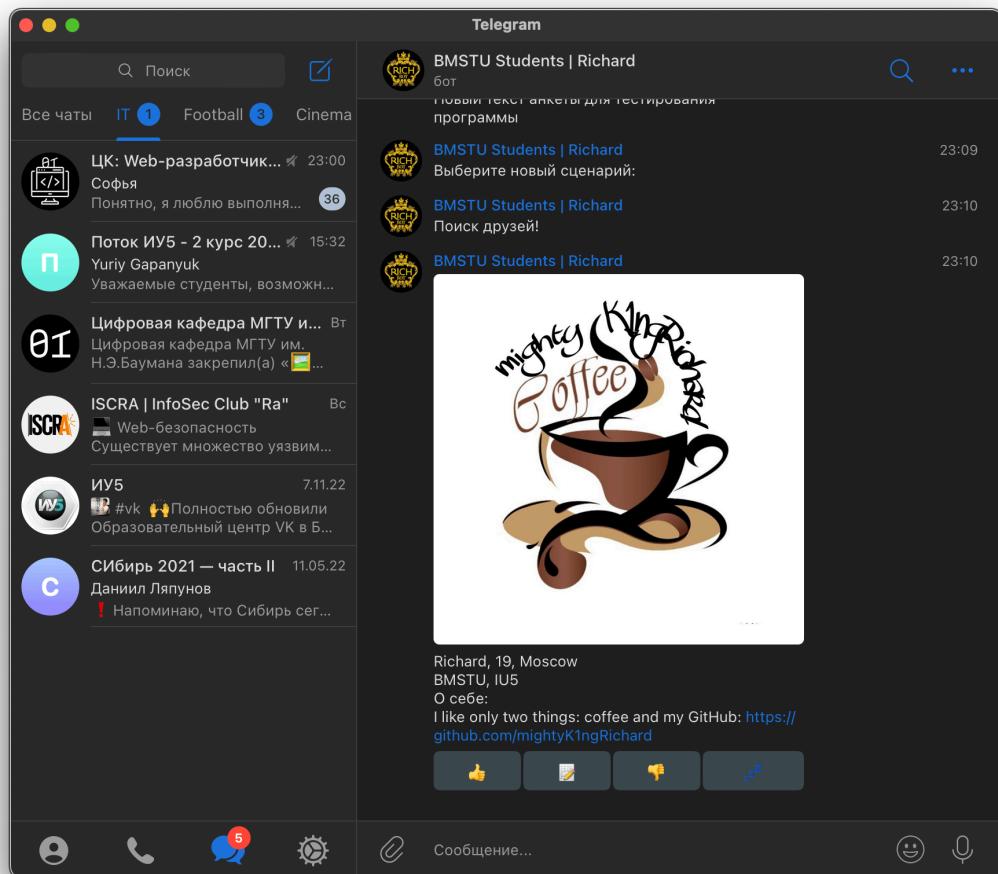
Dmitriy Permyakov, 19, Москва
BMSTU, IU5
О себе:
Новый текст анкеты для тестирования
программы

BMSTU Students | Richard 23:09
Выберите новый сценарий:

Сообщение...

Заполнить анкету заново Изменить фото

Изменить текст анкеты Искать друзей 🤝



Нажимаем 👎

Telegram

Поиск

IT 1 Football 3 Cinema

ЦК: Web-разработчик... 23:00
София
Понятно, я люблю выполнять... 36

Поток ИУБ - 2 курс 20... 15:32
Yuriy Gapanyuk
Уважаемые студенты, возможн...

Цифровая кафедра МГТУ и... 19:00
Цифровая кафедра МГТУ им.
Н.Э.Баумана закрепил(а) «...

ISCRA | InfoSec Club "Ra" Вс
Web-безопасность
Существует множество уязвим...

ИУБ 7.11.22
#vk #Полностью обновили
Образовательный центр VK в Б...

Сибирь 2021 – часть II 11.05.22
Даниил Ляпунов
! Напоминаю, что Сибирь сег...

BMSTU Students | Richard

Richard, 19, Moscow
BMSTU, IU5
О себе:
I like only two things: coffee and my GitHub: <https://github.com/mightyKingRichard>

23:10

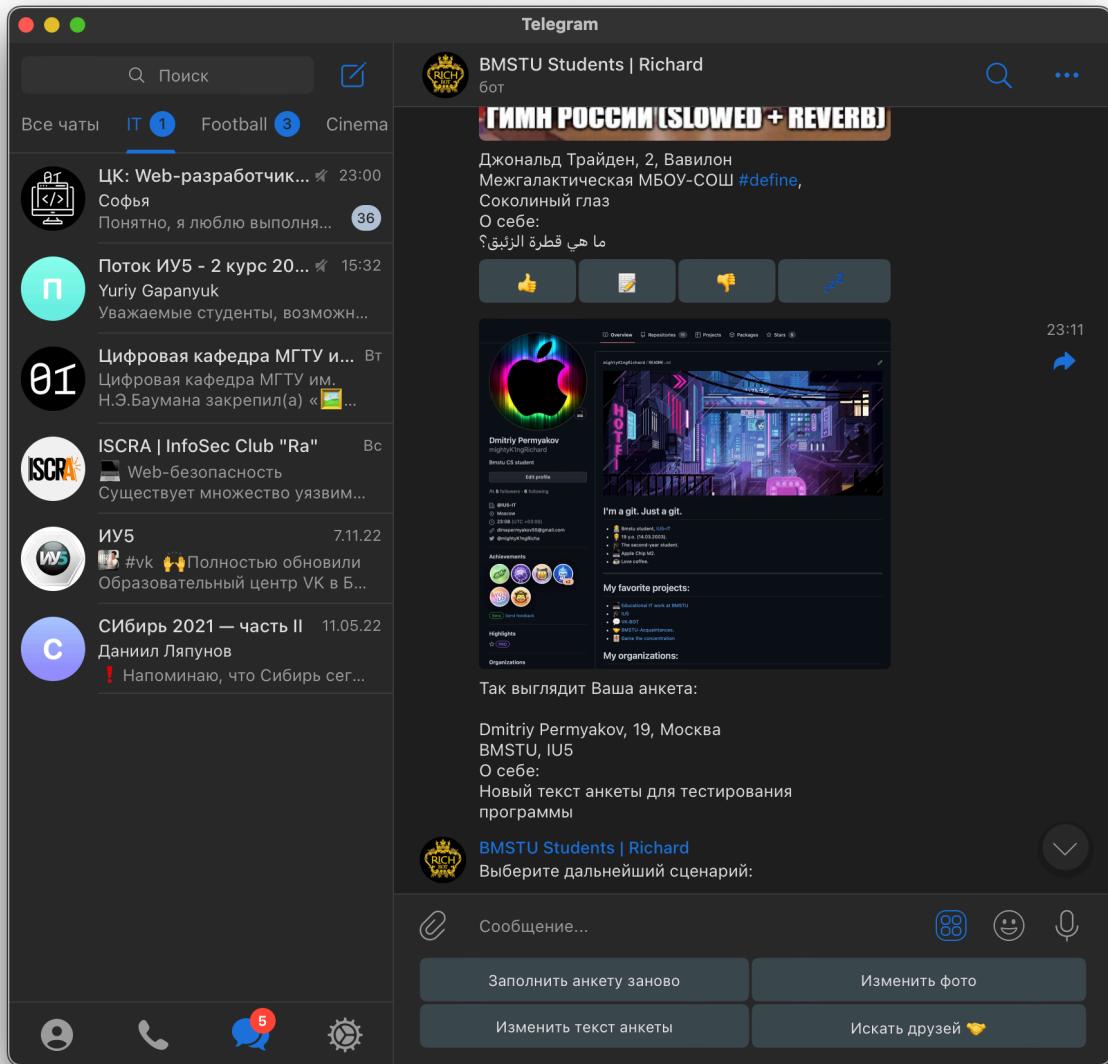
ДОРОГИЕ ДАМЫ

ПОПРОШУ ВСТАТЬ ИГРАЕТ
ГИМН РОССИИ (SLOWED + REVERB)

Джональд Трайден, 2, Вавилон
Межгалактическая МБОУ-СОШ #define,
Соколиный глаз
О себе:
ما هي قطرة الزيقان؟

Нажимаем **zzz**

Сообщение...



Ссылка на телеграм бот - [тут](#)