



Московский государственный технический университет

им. Н.Э. Баумана

(МГТУ им. Н.Э. Баумана)

Кафедра «Системы обработки информации и управления» (ИУ5)

Лабораторная работа № 4

По дисциплине: «Парадигмы и конструкции языков программирования»

«ТЕЛЕГРАМ-БОТ»

Выполнила: Вешторт Е. С.,

студентка группы ИУ5-32Б

Проверил: Нардид А. Н.

г. Москва 2023 г.

Задание: создать телеграм-бота с реализацией кнопок на основе конечного автомата.

В рамках лабораторной работы я создала бота-администратора для записи клиентов на тату и их консультации по разным вопросам. Кроме того, по ID аккаунта администратора доступны две дополнительных кнопки для редактирования базы данных фотографий (портфолио). При перезагрузке бота состояния сохраняются в той же базе данных.

```
import sqlite3
import telebot
from telebot import types
import secret_token

bot = telebot.TeleBot(secret_token.TOKEN)

@bot.message_handler(commands = ['start'])
def start(message):
    connection = sqlite3.connect('bot_database.db')
    cursor = connection.cursor()
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS Users (
            user_id INTEGER,
            step TEXT
        )
    ''')
    connection.commit()
    user_step = cursor.execute('SELECT step FROM Users WHERE user_id =?',
(message.from_user.id,)).fetchone()
    if user_step is None:
        cursor.execute('INSERT INTO Users (user_id, step) VALUES(?, ?)',
(message.from_user.id, 'menu'))
        connection.commit()
    else:
        cursor.execute('UPDATE Users SET step = ? WHERE user_id = ?', ('menu',
message.from_user.id))
        connection.commit()

    connection.close()
    markup = types.ReplyKeyboardMarkup()
    btn1 = types.KeyboardButton('Узнать больше о мастере и процессе')
    btn2 = types.KeyboardButton('Рассчитать стоимость тату')
    btn3 = types.KeyboardButton('Группа ВК')
    btn4 = types.KeyboardButton('Страница в инстаграм')
    admin_btn1 = types.KeyboardButton('Добавить фото в портфолио')
    admin_btn2 = types.KeyboardButton('Очистить портфолио')
    markup.row(btn1)
    markup.row(btn2)
    markup.row(btn3, btn4)
    if message.from_user.id == secret_token.ADMIN_ID:
        markup.row(admin_btn1, admin_btn2)
    bot.send_message(message.chat.id, f'<b>Меню</b>', parse_mode = 'html',
reply_markup = markup)
    bot.register_next_step_handler(message, menu)

def menu(message):
    if message.text == 'Узнать больше о мастере и процессе':
        connection = sqlite3.connect('bot_database.db')
        cursor = connection.cursor()
        cursor.execute('UPDATE Users SET step = ? WHERE user_id = ?', ('info',
message.from_user.id))
```

```

connection.commit()
connection.close()
markup = types.ReplyKeyboardMarkup()
markup.row(types.KeyboardButton('Хочу посмотреть работы'))
markup.row(types.KeyboardButton('Где проходят сеансы?'))
markup.row(types.KeyboardButton('Какие материалы ты используешь?'))
markup.row(types.KeyboardButton('А это точно безопасно?'))
markup.row(types.KeyboardButton('Что меня ждёт?'))
markup.row(types.KeyboardButton('Как выбрать эскиз?'))
markup.row(types.KeyboardButton('У меня другой вопрос'))
markup.row(types.KeyboardButton('Назад'))
bot.send_message(message.chat.id, 'Что бы ты хотел узнать?', reply_markup =
markup)
bot.register_next_step_handler(message, info)

elif message.text == 'Рассчитать стоимость тату':
    if message.from_user.id != secret_token.ADMIN_ID:
        bot.send_message(secret_token.ADMIN_ID, f'@{message.from_user.username}
начал(а) рассчитывать стоимость тату')

    elif message.text == 'Группа ВК':
        bot.send_message(message.chat.id, f'<a
href="https://vk.com/lostearink">Ссылка на мою группу</a>', parse_mode = 'html')
        start(message)

    elif message.text == 'Страница в инстаграм':
        bot.send_message(message.chat.id, f'<a
href="https://instagram.com/the_lost_ear_ink?igshid=NzZl0DBkYWE4Ng==">Ссылка на мою
инсту</a>', parse_mode = 'html')
        start(message)

    elif message.text == 'Добавить фото в портфолио' and message.from_user.id ==
secret_token.ADMIN_ID:
        connection = sqlite3.connect('bot_database.db')
        cursor = connection.cursor()
        cursor.execute('UPDATE Users SET step = ? WHERE user_id = ?', ('portfolio',
(message.from_user.id)))
        connection.commit()
        connection.close()
        bot.send_message(message.chat.id, 'Пришли фото')
        bot.register_next_step_handler(message, portfolio)

    elif message.text == 'Очистить портфолио' and message.from_user.id ==
secret_token.ADMIN_ID:
        connection = sqlite3.connect('bot_database.db')
        cursor = connection.cursor()

        cursor.execute('''
DROP TABLE IF EXISTS Portfolio
''')
        connection.commit()
        connection.close()
        bot.send_message(message.chat.id, 'Портфолио успешно очищено')
        start(message)

    else:
        bot.send_message(message.chat.id, 'Я тебя не понимаю, пожалуйста, выбери
одну из опций')
        bot.register_next_step_handler(message, menu)

def info(message):
    if message.text == 'Хочу посмотреть работы':
        connection = sqlite3.connect('bot_database.db')
        cursor = connection.cursor()

```

```

try:
    cursor.execute('SELECT file_id FROM Portfolio')
    photos = cursor.fetchall()
    for photo in photos:
        bot.send_photo(message.chat.id, photo[0])
    connection.close()

except:
    bot.send_message(message.chat.id, 'Портфолио пока пусто, приходи позже')
    message.text = 'Узнать больше о мастере и процессе'
    menu(message)

elif message.text == 'Где проходят сеансы?':
    bot.send_message(message.chat.id, '')
    message.text = 'Узнать больше о мастере и процессе'
    menu(message)

elif message.text == 'Назад':
    start(message)

else:
    bot.send_message(message.chat.id, 'Я тебя не понимаю, пожалуйста, выбери
одну из опций')
    bot.register_next_step_handler(message, info)

def portfolio(message):
    try:
        connection = sqlite3.connect('bot_database.db')
        cursor = connection.cursor()
        cursor.execute('''
CREATE TABLE IF NOT EXISTS Portfolio (
id INTEGER PRIMARY KEY,
file_id TEXT
)
''')
        connection.commit()
        cursor.execute('INSERT INTO Portfolio (file_id) VALUES(?)', (message.photo[-
1].file_id, ))
        connection.commit()
        connection.close()
        bot.send_message(message.chat.id, 'Фото успешно добавлено')
        start(message)
    except:
        bot.send_message(message.chat.id, 'Нужно прислать именно фото')
        bot.register_next_step_handler(message, portfolio)

@bot.message_handler(content_types=['text', 'photo'])
def checking_status(message):
    connection = sqlite3.connect('bot_database.db')
    cursor = connection.cursor()
    user_step = cursor.execute('SELECT step FROM Users WHERE user_id =?',
(message.from_user.id,)).fetchone()
    if user_step[0] == 'menu':
        menu(message)
    elif user_step[0] == 'info':
        info(message)
    elif user_step[0] == 'portfolio':
        portfolio(message)

bot.polling(none_stop=True)

```





