Задание

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

Код программы.

handlers.py

```python
from aiogram import html, F, Router
from aiogram.fsm.context import FSMContext
from aiogram.fsm.state import default_state
from aiogram.filters import CommandStart, Command, StateFilter
from aiogram.types import Message, CallbackQuery

from app.keyboards import Kb

router = Router()

@router.message(CommandStart())
async def cmd_start(message: Message):
    await message.answer(f"Hello, {html.bold(message.from_user.full_name)}!",
reply_markup=Kb.main)

@router.message(Command("help"))
async def cmd_help(message: Message):
    await message.reply("команды:\n/help\n/start\n/covid\n/smile")

@router.message(StateFilter(None), Command(commands=["cancel"]))
@router.message(default_state, F.text.lower() == "отмена")
async def cmd_cancel_no_state(message: Message, state: FSMContext):
    await state.set_data({})
    await message.answer(text="Нечего отменять")

@router.message(Command(commands=["cancel"]))
@router.message(F.text.lower() == "отмена")
async def cmd_cancel(message: Message, state: FSMContext):
    await state.clear()
    await message.answer(text="Действие отменено")

@router.message(F.text == "ggg")
async def msg_catch(message: Message):
    await message.answer("fff")

@router.message(F.text == "1")
async def rkb_one(message: Message):
    await message.answer("Choose", reply_markup=Kb.ctlg)

@router.callback_query(F.data == "ib9")
async def ikb_nine(callback: CallbackQuery):
    await callback.answer("accept")
    await callback.message.answer("its 9")

@router.message()
async def echo_handler(message: Message):
    try: await message.send_copy(chat_id=message.chat.id)
    except TypeError: await message.answer("Nice try!")
```

keyboards.py

```python
from aiogram import html, F, Router
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton,
InlineKeyboardMarkup, InlineKeyboardButton

class Kb:
    main = ReplyKeyboardMarkup(keyboard=[[KeyboardButton(text="1")],
                                          [KeyboardButton(text="2")],
                                          [KeyboardButton(text="3"),
KeyboardButton(text="4")]],
                               resize_keyboard=True,
input_field_placeholder="choose...")

    ctlg =
InlineKeyboardMarkup(inline_keyboard=[[InlineKeyboardButton(text="9",
callback_data="ib9")],

[InlineKeyboardButton(text="8", callback_data="ib8")],

[InlineKeyboardButton(text="7", callback_data="ib7")]])

    type = ReplyKeyboardMarkup(keyboard=[[KeyboardButton(text="user"),
KeyboardButton(text="root")]],
                               resize_keyboard=True,
input_field_placeholder="choose role")
```

states.py

```
from aiogram import Router, F
from aiogram.filters import Command
from aiogram.fsm.context import FSMContext
from aiogram.fsm.state import StatesGroup, State
from aiogram.types import Message, ReplyKeyboardRemove

from app.keyboards import Kb

routerr = Router()

class Registration(StatesGroup):
    type = State()
    login = State()
    password = State()


@routerr.message(Command("reg"))
async def reg_start(message: Message, state: FSMContext):
    try: state.clear()
    except: pass
    await state.update_data(user_id=message.from_user.id)
    await message.answer(text="choose root or user:", reply_markup=Kb.type)
    await state.set_state(Registration.type)

@routerr.message(Command("my_data"))
async def output_data(message: Message, state: FSMContext):
    user_data = await state.get_data()
    if len(user_data.keys())==0: await message.answer(text="no data")
    else: await message.answer(text=f"login: {user_data['user_id']}\nlogin:
{user_data['type']}\n"
                                    f"login: {user_data['login']}\npassword:
{user_data['password']}.")

@routerr.message(F.text, Registration.type)
async def rooting(message: Message, state: FSMContext):
    if message.text not in ["user", "root"]:
        await message.answer(text="error, choose", reply_markup=Kb.type)
        return
    await state.update_data(type=message.text)
    await message.answer(text="Enter login:",
reply_markup=ReplyKeyboardRemove())
    await state.set_state(Registration.login)

@routerr.message(F.text, Registration.login)
async def logining(message: Message, state: FSMContext):
    await state.update_data(login=message.text)
    await message.answer(text="Enter password:")
    await state.set_state(Registration.password)

@routerr.message(F.text, Registration.password)
async def passwording(message: Message, state: FSMContext):
    await state.update_data(password=message.text)
    await message.answer(text=f"auntification success")
    await state.set_state(None)
```

main.py

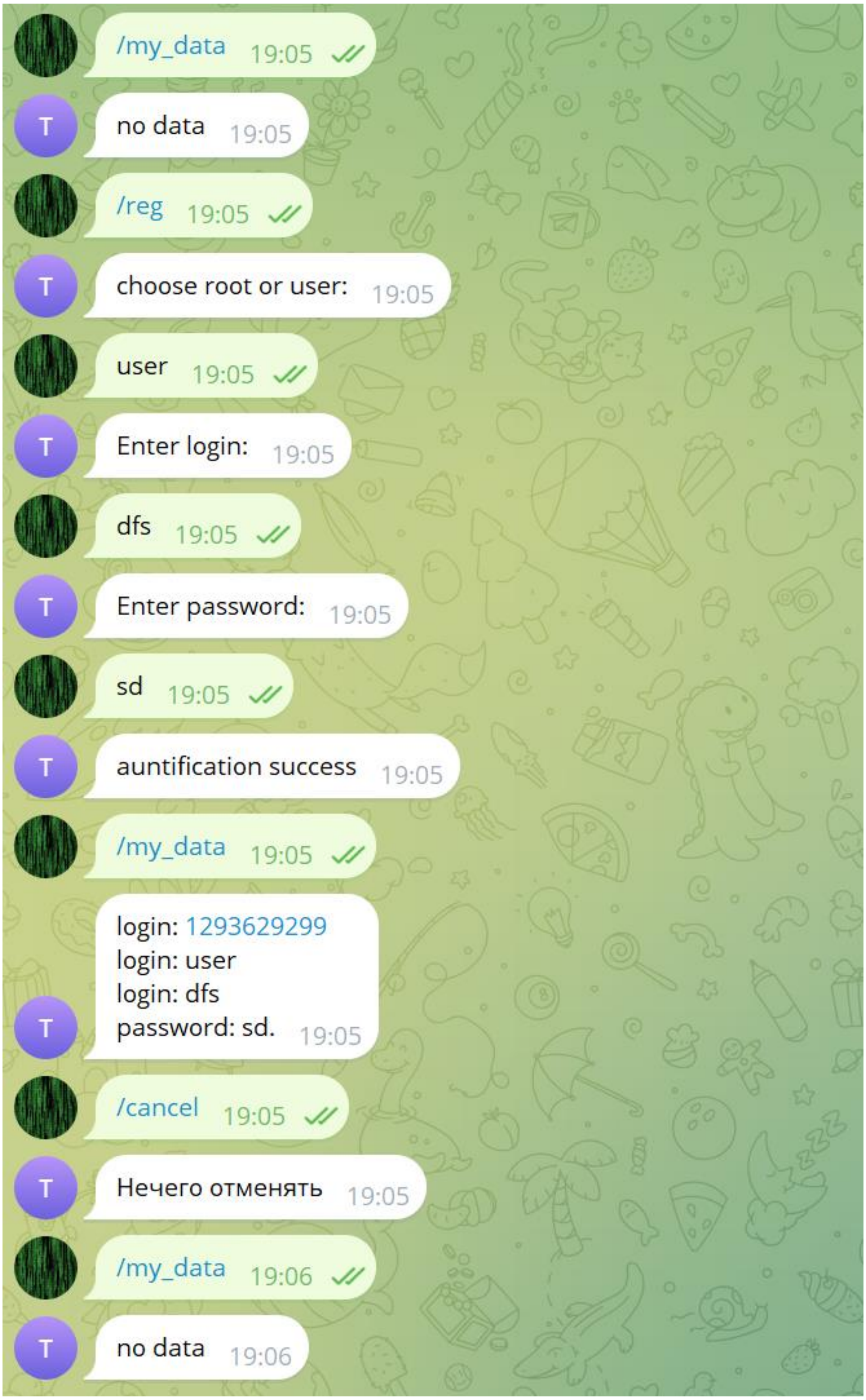
```
import asyncio
from aiogram import Bot, Dispatcher

from app.handlers import router

async def main():
    TOKEN = "token"
    bot = Bot(token=TOKEN)
    dp = Dispatcher()
    dp.include_router(router)
    await dp.start_polling(bot)

if __name__ == "__main__":
    try: asyncio.run(main())
    except KeyboardInterrupt: print("end")
```

Результаты.



```
/my_data 19:05
no data 19:05
/reg 19:05
choose root or user: 19:05
user 19:05
Enter login: 19:05
dfs 19:05
Enter password: 19:05
sd 19:05
auntification success 19:05
/my_data 19:05
login: 1293629299
login: user
login: dfs
password: sd. 19:05
/cancel 19:05
Нечего отменять 19:05
/my_data 19:06
no data 19:06
```