**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

**Журнал практики**

Студента *Петренко Дмитрия Евгеньевича*

##### Факультет Ступинский филиал МАИ

###### Кафедра *«Моделирование систем и информационные технологии»*

##### Учебная группа ТСО-105Б-22

Направление подготовки (специальность) *09.03.01*

*(шифр)*

*«Информатика и вычислительная техника»*

*(название направления, специальности)*

Вид практики *учебная*

*(учебной, производственной, преддипломной или другой вид практики)*

Руководитель практики от МАИ

*Мамонов Игорь Михайлович*  \_\_\_\_\_\_\_\_\_\_\_\_\_

*(фамилия, имя, отчество) (подпись)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*Петренко Д.Е.*/ “12” июля 2023 г.

*(подпись студента)* *(дата)*

Москва 2023

**1.Место и сроки проведения практики**

*Сроки проведения практики:*

*-дата начала практики 29.06.2023 г.*

*-дата окончания практики 12.07.2023 г.*

*Наименование предприятия Ступинский филиал МАИ*

*Название структурного подразделения (отдел, лаборатория) кафедра «Моделирование систем и информационные технологии»*

1. **Инструктаж по технике безопасности**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Беспалов В.В./ “29” июня 2023 г.

*(подпись проводившего)* *(дата проведения)*

1. **Индивидуальное задание студенту**

Разработка анонимного чат бота в сети Telegram на основе ЯП python

**4.План выполнения индивидуального задания**

1. Изучить возможности данной социальной сети.

2. Выполнить установку и настройку чат-бота.

3. Подготовить реферат.

*Руководитель практики от МАИ*: */Мамонов И.М./*

*Руководитель от предприятия*:  */ /*

*/Петренко Д.Е./* “29” июня 2023 г.

*(подпись студента)* *(дата)*

**5.Отзыв руководителя практики от предприятия**

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

*Руководитель от предприятия*: / /

*(подпись) (фамилия, имя, отчество)*

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

М.П. (печать)

**6.Отчет студента о практике**

Целью учебной практики являлось создание анонимного чат бота в сети Telegram на базе ЯП Python.

### Установка и настройка среды программирования:

Скачиваем и устанавливаем среду разработки PyCharm

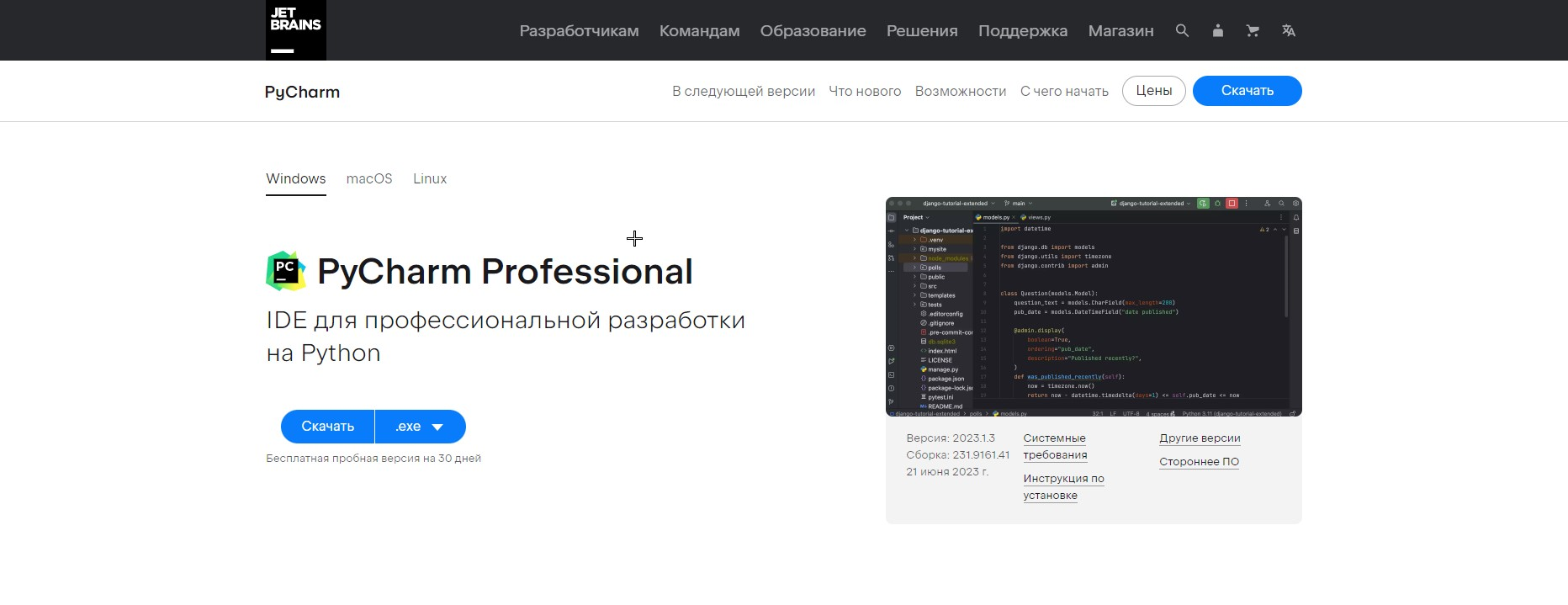


Рис. 1. Установка PyCharm.

Создаем папку, в которой будет храниться наш проект, и создаем 3 файла

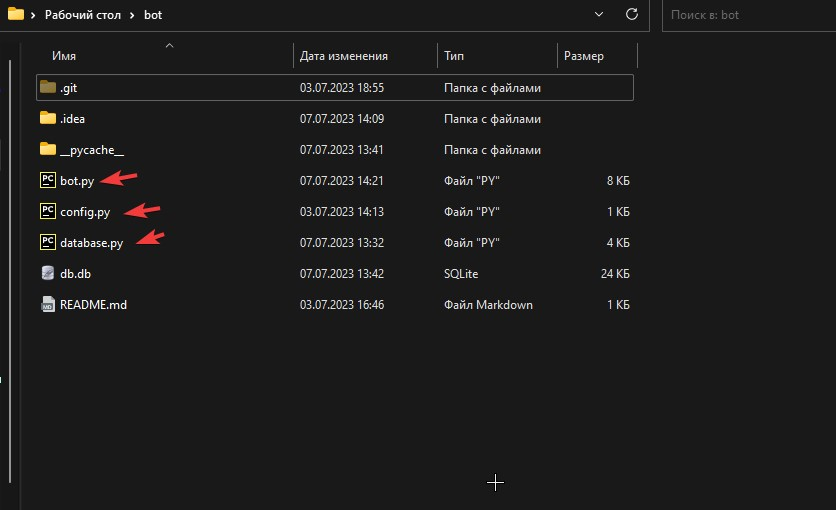


Рис. 2. Создание папки проекта.

Далее устанавливаем расширения Python для работы с ботом

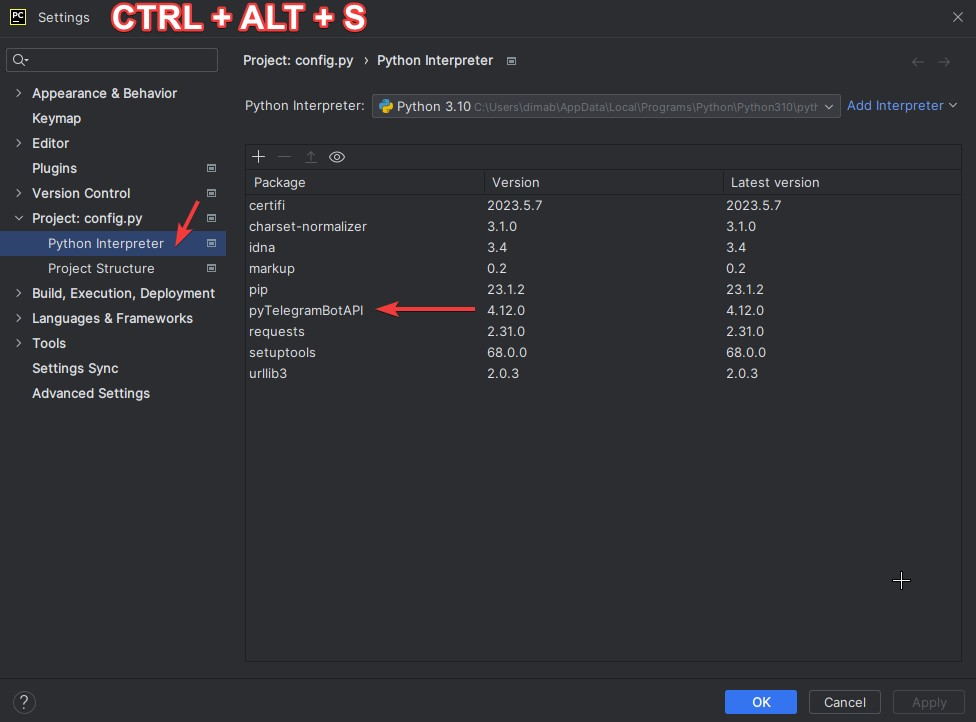


Рис. 3. Установка расширений.

Устанавливаем SQLiteStudio, для будущей базы данных

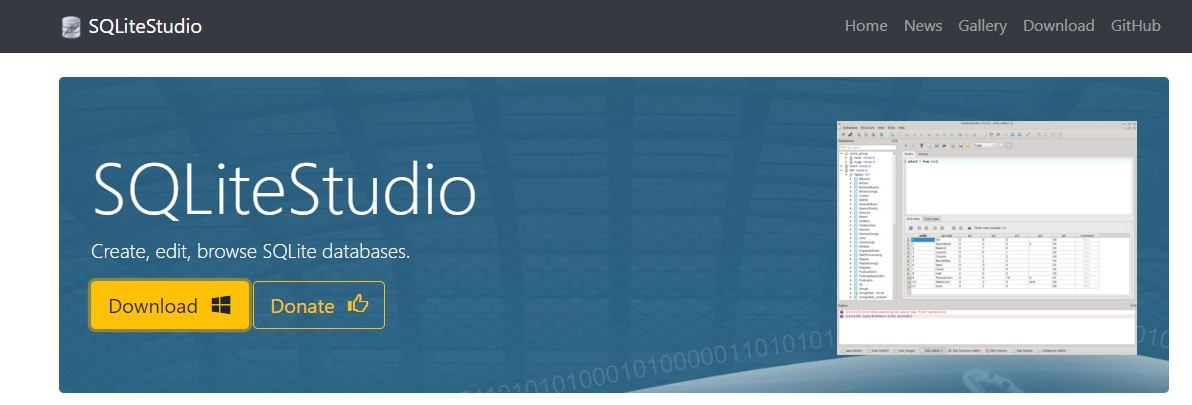


Рис. 4. Установка SQLiteStudio

Установка и настройка закончена

Переходим к написанию кода:

Импортируем установленные библиотеки

import config  
import telebot  
from telebot import types  
from database import Database

После этого нам нужно зарегистрировать нашего бота в BotFather в Telegram командой /newbot и получить токен (индикационный номер)

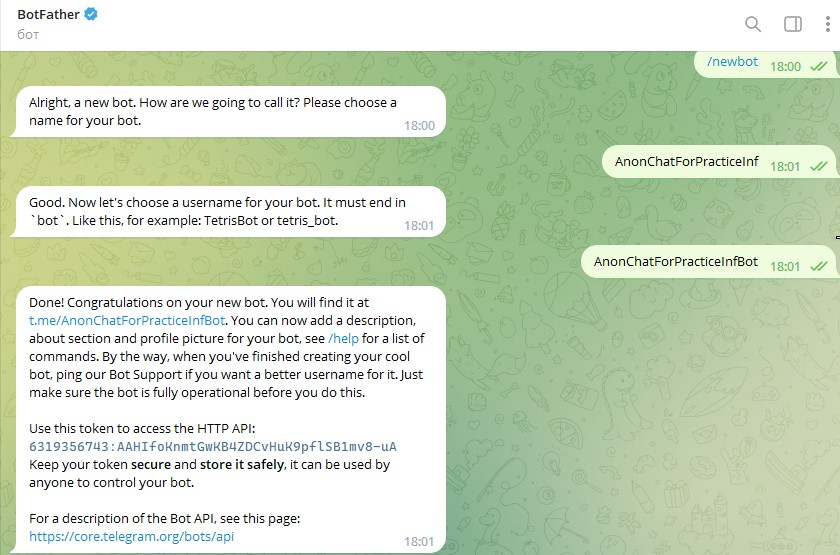


Рис. 5. Создание бота, получение токена.

Прописываем наш токен в файле config.py для удобства

TOKEN = "6319356743:AAHIfoKnmtGwKB4ZDCvHuK9pflSB1mv8-uA"

После этого создаем объекты бота в python. token представляет собой секретный ключ, который используется для аутентификации бота. Объект bot создается для получения токена из файла config.py. Он используется для управления распределением сообщений между пользователем и ботом. Также подключаем файл с базой данных.

db = Database('db.db')  
bot = telebot.TeleBot(config.TOKEN)

Далее создаем базу данных в SQLiteStudio

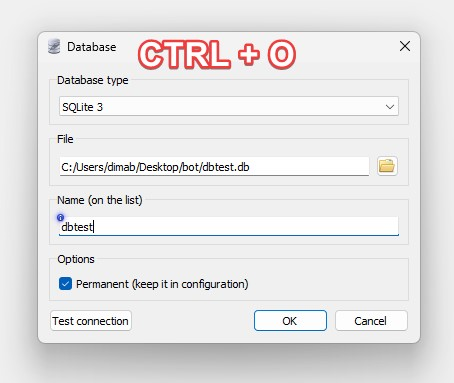


Рис. 6. Создание базы данных.

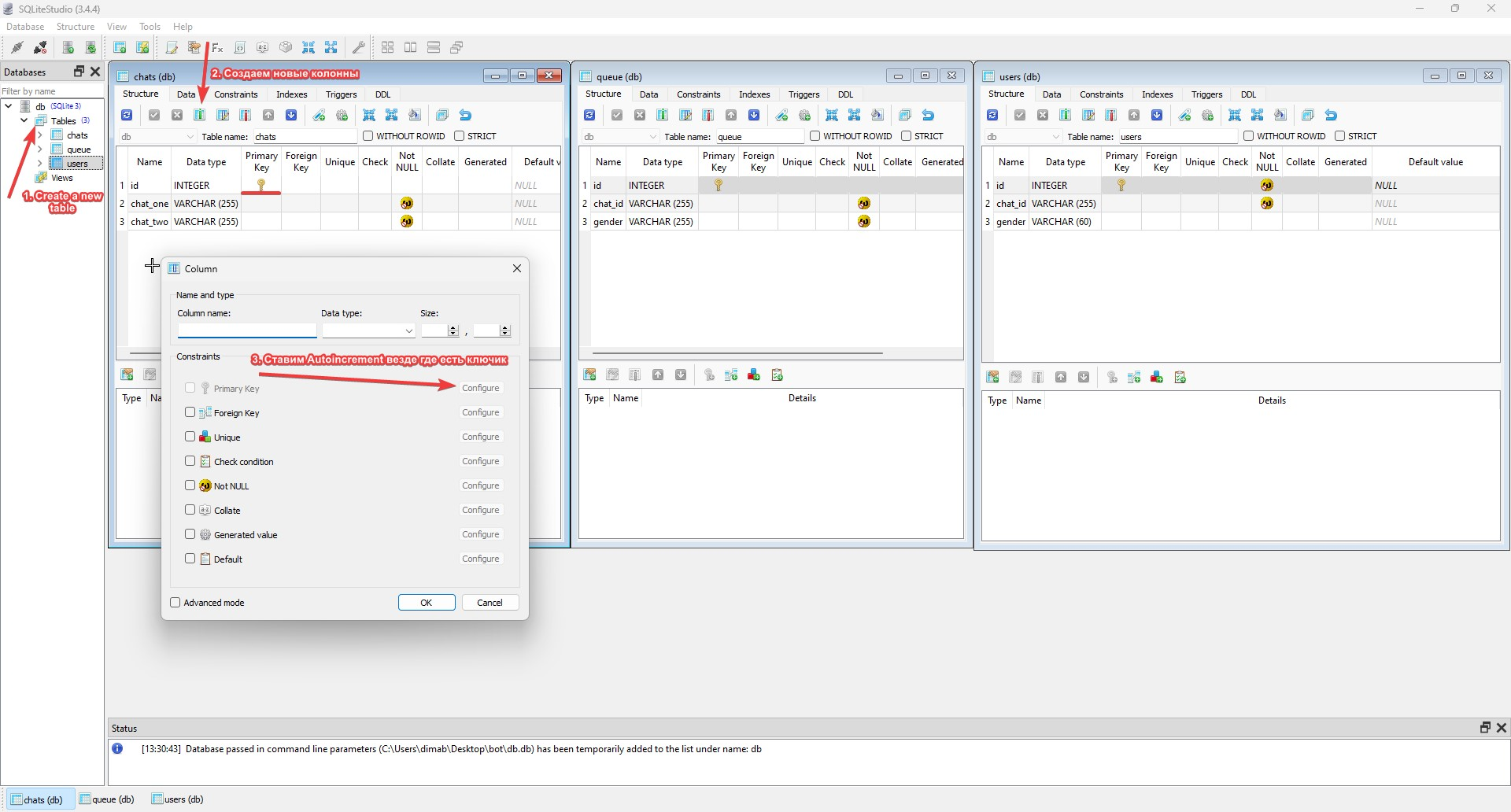


Рис. 7. Создаем таблицы базы данных.

Прописываем в коде основные функции кнопок, которые используются всегда

def main\_menu():  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 item1 = types.KeyboardButton('🔍Поиск собеседника')  
 markup.add(item1)  
 return markup  
  
def stop\_dialog():  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 item1 = types.KeyboardButton('🗣Сказать свой профиль')  
 item2 = types.KeyboardButton('/stop')  
 markup.add(item1, item2)  
 return markup  
  
def stop\_search():  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 item1 = types.KeyboardButton('❌Остановить поиск')  
 markup.add(item1)  
 return markup

Прописываем основные команды для бота

@bot.message\_handler(commands=['start'])  
def start(message):  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 item1 = types.KeyboardButton('Я парень🙍‍♂️')  
 item2 = types.KeyboardButton('Я девушка🙍‍♀️')  
 markup.add(item1, item2)  
  
 bot.send\_message(message.chat.id, 'Привет, {0.first\_name}! Добро пожаловать в анонимный чат! Укажите ваш пол'.format(message.from\_user), reply\_markup=markup)  
  
@bot.message\_handler(commands=['menu'])  
def menu(message):  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 item1 = types.KeyboardButton('🔍Поиск собеседника')  
 markup.add(item1)  
  
 bot.send\_message(message.chat.id, 'Меню'.format(message.from\_user), reply\_markup=markup)  
  
@bot.message\_handler(commands=['stop'])  
def stop(message):  
 chat\_info = db.get\_active\_chat(message.chat.id)  
 if chat\_info:  
 db.delete\_chat(chat\_info[0])  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 item1 = types.KeyboardButton('Следующий диалог➡️')  
 item2 = types.KeyboardButton('/menu')  
 markup.add(item1, item2)  
  
 bot.send\_message(chat\_info[1], 'Собеседник покинул чат😔', reply\_markup=markup)  
 bot.send\_message(message.chat.id, 'Вы вышли из чата', reply\_markup=markup)

Далее переходим в файл database.py, прописываем фунцкии базы данных

import sqlite3  
  
class Database:  
 def \_\_init\_\_(self, database\_file):  
 self.connection = sqlite3.connect(database\_file, check\_same\_thread=False)  
 self.cursor = self.connection.cursor()  
  
 def add\_queue(self, chat\_id, gender):  
 with self.connection:  
 return self.cursor.execute("INSERT INTO `queue` (`chat\_id`, `gender`) VALUES (?,?)", (chat\_id, gender))  
  
 def delete\_queue(self, chat\_id):  
 with self.connection:  
 return self.cursor.execute("DELETE FROM `queue` WHERE `chat\_id` = ?", (chat\_id,))  
  
 def delete\_chat(self, id\_chat):  
 with self.connection:  
 return self.cursor.execute("DELETE FROM `chats` WHERE `id` = ?", (id\_chat,))  
  
 def set\_gender(self, chat\_id, gender):  
 with self.connection:  
 user = self.cursor.execute("SELECT \* FROM `users` WHERE `chat\_id` = ?", (chat\_id,)).fetchmany(1)  
 if not bool(len(user)):  
 self.cursor.execute("INSERT INTO `users` (`chat\_id`, `gender`) VALUES (?,?)", (chat\_id, gender))  
 return True  
 else:  
 return False  
  
 def get\_gender(self, chat\_id):  
 with self.connection:  
 user = self.cursor.execute("SELECT \* FROM `users` WHERE `chat\_id` = ?", (chat\_id,)).fetchmany(1)  
 if bool(len(user)):  
 for row in user:  
 return row[2]  
 else:  
 return False  
  
 def get\_gender\_chat(self, gender):  
 with self.connection:  
 chat = self.cursor.execute("SELECT \* FROM `queue` WHERE `gender` = ?", (gender,)).fetchmany(1)  
 if bool(len(chat)):  
 for row in chat:  
 user\_info = [row[1], row[2]]  
 return user\_info  
 else:  
 return [0]  
  
 def get\_chat(self):  
 with self.connection:  
 chat = self.cursor.execute("SELECT \* FROM `queue`", ()).fetchmany(1)  
 if bool(len(chat)):  
 for row in chat:  
 user\_info = [row[1], row[2]]  
 return user\_info  
 else:  
 return [0]  
  
 def create\_chat(self, chat\_one, chat\_two):  
 with self.connection:  
 if chat\_two != 0:  
 self.cursor.execute("DELETE FROM `queue` WHERE `chat\_id` = ?", (chat\_two,))  
 self.cursor.execute("INSERT INTO `chats` (`chat\_one`, `chat\_two`) VALUES (?,?)", (chat\_one, chat\_two,))  
 return True  
 else:  
  
 return False  
  
 def get\_active\_chat(self, chat\_id):  
 with self.connection:  
 chat = self.cursor.execute("SELECT \* FROM `chats` WHERE `chat\_one` = ?", (chat\_id,))  
 id\_chat = 0  
 for row in chat:  
 id\_chat = row[0]  
 chat\_info = [row[0], row[2]]  
  
 if id\_chat == 0:  
 chat = self.cursor.execute("SELECT \* FROM `chats` WHERE `chat\_two` = ?", (chat\_id,))  
 for row in chat:  
 id\_chat = row[0]  
 chat\_info = [row[0], row[1]]  
 if id\_chat == 0:  
 return False  
 else:  
 return chat\_info  
 else:  
 return chat\_info

Прописываем основную часть кода в bot.py, для отправки сообщений и выбора пола собеседника:

@bot.message\_handler(content\_types=['text'])  
def bot\_message(message):  
 if message.chat.type == 'private':  
 if message.text == '🔍Поиск собеседника' or message.text == 'Следующий диалог➡️':  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 item1 = types.KeyboardButton('Парень🙍‍♂️‍')  
 item2 = types.KeyboardButton('Девушка🙍‍♀️')  
 item3 = types.KeyboardButton('Рандом👫')  
 markup.add(item1, item2, item3)  
  
 bot.send\_message(message.chat.id, 'Кого искать?🕵️‍♂️', reply\_markup=markup)  
  
 elif message.text == '❌Остановить поиск':  
 db.delete\_queue(message.chat.id)  
 bot.send\_message(message.chat.id, 'Поиск остановлен, напишите /menu', reply\_markup=main\_menu())  
  
 elif message.text == 'Парень🙍‍♂️‍':  
 user\_info = db.get\_gender\_chat('male')  
 chat\_two = user\_info[0]  
 if not db.create\_chat(message.chat.id, chat\_two):  
 db.add\_queue(message.chat.id, db.get\_gender(message.chat.id))  
 bot.send\_message(message.chat.id, '💣Поиск собеседника', reply\_markup=stop\_search())  
 else:  
 mess = 'Собеседник найден! Чтобы остановить диалог, напишите /stop'  
  
 bot.send\_message(message.chat.id, mess, reply\_markup=stop\_dialog())  
 bot.send\_message(chat\_two, mess, reply\_markup=stop\_dialog())  
  
 elif message.text == 'Девушка🙍‍♀️':  
 user\_info = db.get\_gender\_chat('female')  
 chat\_two = user\_info[0]  
 if not db.create\_chat(message.chat.id, chat\_two):  
 db.add\_queue(message.chat.id, db.get\_gender(message.chat.id))  
 bot.send\_message(message.chat.id, '💣Поиск собеседника', reply\_markup=stop\_search())  
 else:  
 mess = 'Собеседник найден! Чтобы остановить диалог, напишите /stop'  
  
 bot.send\_message(message.chat.id, mess, reply\_markup=stop\_dialog())  
 bot.send\_message(chat\_two, mess, reply\_markup=stop\_dialog())  
  
 elif message.text == 'Рандом👫':  
 user\_info = db.get\_chat()  
 chat\_two = user\_info[0]  
  
 if not db.create\_chat(message.chat.id, chat\_two):  
 db.add\_queue(message.chat.id, db.get\_gender(message.chat.id))  
 bot.send\_message(message.chat.id, '💣Поиск собеседника', reply\_markup=stop\_search())  
 else:  
 mess = 'Собеседник найден! Чтобы остановить диалог, напишите /stop'  
  
 bot.send\_message(message.chat.id, mess, reply\_markup=stop\_dialog())  
 bot.send\_message(chat\_two, mess, reply\_markup=stop\_dialog())  
 elif message.text == '🗣Сказать свой профиль':  
 chat\_info = db.get\_active\_chat(message.chat.id)  
 if chat\_info:  
 if message.from\_user.username:  
 bot.send\_message(chat\_info[1], 'Профиль собеседника @' + message.from\_user.username)  
 bot.send\_message(message.chat.id, 'Вы сказали свой профиль')  
 else:  
 bot.send\_message(chat\_info[1], 'Вы не указали свой username')  
  
 elif message.text == 'Я парень🙍‍♂️':  
 if db.set\_gender(message.chat.id, 'male'):  
 bot.send\_message(message.chat.id, 'Ваш пол уже успешно добавлен', reply\_markup=main\_menu())  
 else:  
 bot.send\_message(message.chat.id, 'Вы уже указали ваш пол. Обратитесь в поддержку @ybicanatinkere')  
  
 elif message.text == 'Я девушка🙍‍♀️':  
 if db.set\_gender(message.chat.id, 'female'):  
 bot.send\_message(message.chat.id, 'Ваш пол уже успешно добавлен', reply\_markup=main\_menu())  
 else:  
 bot.send\_message(message.chat.id, 'Вы уже указали ваш пол. Обратитесь в поддержку @ybicanatinkere')  
  
 else:  
 if db.get\_active\_chat(message.chat.id):  
 chat\_info = db.get\_active\_chat(message.chat.id)  
 bot.send\_message(chat\_info[1], message.text)  
  
  
@bot.message\_handler(content\_types=['sticker'])  
def bot\_stickers(message):  
 if message.chat.type == 'private':  
 chat\_info = db.get\_active\_chat(message.chat.id)  
 if chat\_info:  
 bot.send\_sticker(chat\_info[1], message.sticker.file\_id)  
 else:  
 bot.send\_message(message.chat.id, '❌ Вы не начали диалог')  
  
@bot.message\_handler(content\_types=['voice'])  
def bot\_voice(message):  
 if message.chat.type == 'private':  
 chat\_info = db.get\_active\_chat(message.chat.id)  
 if chat\_info:  
 bot.send\_voice(chat\_info[1], message.voice.file\_id)  
 else:  
 bot.send\_message(message.chat.id, '❌ Вы не начали диалог')  
  
@bot.message\_handler(content\_types=['photo'])  
def bot\_photo(message):  
 if message.chat.type == 'private':  
 chat\_info = db.get\_active\_chat(message.chat.id)  
 if chat\_info:  
 bot.send\_photo(chat\_info[1], message.photo[-1].file\_id)  
 else:  
 bot.send\_message(message.chat.id, '❌ Вы не начали диалог!')  
  
@bot.message\_handler(content\_types=['video'])  
def bot\_video(message):  
 if message.chat.type == 'private':  
 chat\_info = db.get\_active\_chat(message.chat.id)  
 if chat\_info:  
 bot.send\_video(chat\_info[1], message.video.file\_id)  
 else:  
 bot.send\_message(message.chat.id, '❌ Вы не начали диалог!')  
  
@bot.message\_handler(content\_types=['video\_note'])  
def bot\_video\_note(message):  
 if message.chat.type == 'private':  
 chat\_info = db.get\_active\_chat(message.chat.id)  
 if chat\_info:  
 bot.send\_video\_note(chat\_info[1], message.video\_note.file\_id)  
 else:  
 bot.send\_message(message.chat.id, '❌ Вы не начали диалог!')  
  
  
bot.polling(none\_stop=True)

Протестируем нашего бота:

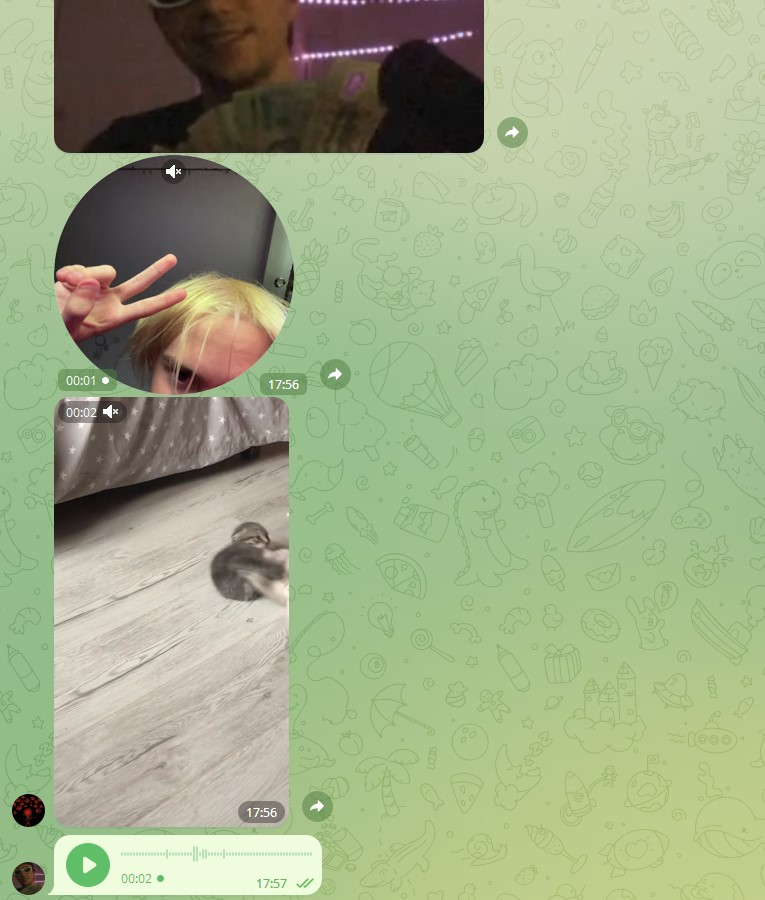


Рис. 7-8. Тестирование бота.

В итоге, был запрограммирован и настроен анонимный чат-бот, в котором можно общаться с собеседником, не раскрывая свою личность (если вы сами этого не захотите).

В результате прохождения учебной практики были изучены и выполнены следующие задачи:

* изучена технология создания ботов на основе библиотеки pyTelegramBot;
* изучено создание базы данных SQLiteStudio и подключение ее к коду;
* создан Анонимный чат-бот в Telegram;
* выполнено тестирование;
* подготовлен реферат.