МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Благовещенский государственный педагогический университет»

Физико-математический факультет

Кафедра информатики и методики преподавания информатики

КУРСОВАЯ РАБОТА

на тему: Разработка многофункционального Telegram-бота

по дисциплине: Современные технологии программирования

Исполнитель: 25.10.2023 М.С. Христюк

студент группы 3 «Б» дата подпись ИОФ

Руководитель:

ст. преподаватель 25.10.2023 А.А. Ягелло

дата подпись ИОФ

Защита состоялась « » 202 г. Оценка

Благовещенск 2023

Оглавление

[ВВЕДЕНИЕ 3](#_Toc153231774)

[1 ОБЩИЕ СВЕДЕНИЯ 5](#_Toc153231775)

[1.1 История создания и развития Telegram-ботов 5](#_Toc153231776)

[1.2 О мессенджере Telegram 6](#_Toc153231777)

[1.3 О Telegram-ботах 7](#_Toc153231778)

[2 ВЫБОР ИНСТРУМЕНТОВ ДЛЯ РЕАЛИЗАЦИИ 9](#_Toc153231779)

[2.1 Преимущества языка программирования Python для разработки бота 9](#_Toc153231780)

[2.2 Обоснование выбора среды разработки 10](#_Toc153231781)

[3 РАЗРАБОТКА БОТА 11](#_Toc153231782)

[3.1 Установка и настройка среды разработки PyCharm 11](#_Toc153231783)

[3.2 Регистрация и настройка бота в Telegram 12](#_Toc153231784)

[3.3 Разработка Telegram-бота 14](#_Toc153231785)

[4 ТЕСТИРОВАНИЕ 19](#_Toc153231786)

[ЗАКЛЮЧЕНИЕ 23](#_Toc153231787)

[СПИСОК ЛИТЕРАТУРЫ 24](#_Toc153231788)

[ПРИЛОЖЕНИЕ 25](#_Toc153231789)

# ВВЕДЕНИЕ

Telegram-боты становятся все более популярными и востребованными в различных сферах деятельности, начиная от бизнеса и маркетинга, и заканчивая сферой образования и медицины. В связи с этим, разработка многофункционального Telegram-бота является актуальной и интересной задачей, которая позволяет автоматизировать рутинные задачи, упростить взаимодействие с пользователями и повысить эффективность работы.

Целью данной курсовой работы является разработка многофункционального Telegram-бота, способного выполнять различные задачи, такие как получение информации о товарах, отправка анекдотов, а также осуществление действий на основе команд. Для достижения поставленной цели необходимо решить следующие задачи:

* Изучение и анализ существующих решений по разработке Telegram-ботов и выбор подходящих технологий и инструментов для разработки.
* Проектирование архитектуры бота, определение основных функциональных и нефункциональных требований.
* Разработка и реализация функционала бота с использованием выбранных инструментов и библиотек.
* Тестирование и отладка разработанного бота, исправление ошибок и улучшение производительности.

В рамках данной курсовой работы будет использоваться язык программирования Python и библиотека aiogram. Они были выбраны из-за своей простоты использования, возможности расширения функционала и широкой поддержки сообщества разработчиков. В процессе работы будет осуществлен обмен данными с использованием Telegram Bot API, который предоставляет все необходимые инструменты для создания и взаимодействия с Telegram-ботами.  
Результатом работы будет полнофункциональный Telegram-бот, способный выполнять различные задачи и удовлетворять потребности пользователей. Это позволит упростить и автоматизировать множество повседневных действий, а также создать возможности для дальнейшего расширения и развития функционала бота.

# 1 ОБЩИЕ СВЕДЕНИЯ

1.1 История создания и развития Telegram-ботов

Telegram-боты являются одной из самых популярных и востребованных функций в мессенджере Telegram. Их создание и развитие имеют интересную историю, начиная с основания самого мессенджера.

Telegram был создан в 2013 году Павлом Дуровым, российским предпринимателем и соучредителем ВКонтакте. Целью создания Telegram было создание безопасной и защищенной платформы для обмена сообщениями и файлами. Благодаря своей шифрованной архитектуре и фокусу на конфиденциальности, Telegram быстро стал популярным среди пользователей со всего мира.

Однако, сам по себе Telegram предлагает много больше, чем просто переписку между людьми. В 2015 году Telegram запустил платформу ботов, которая позволила разработчикам создавать свои собственные приложения-боты внутри мессенджера. Это привело к революции в мире коммуникации и информации.

История развития Telegram-ботов началась с простых функций, таких как определение погоды, перевод текста и получение новостей. Со временем, благодаря дальнейшему развитию интерфейса ботов и возможностей API Telegram, боты стали более сложными и функциональными.

В настоящее время Telegram-боты могут выполнять широкий спектр задач. Они могут использоваться для автоматизации бизнес-процессов, обработки заказов, предоставления информации о товарах и услугах, игр, управления финансами и даже создания искусственного интеллекта.

Telegram-боты предоставляют удобный интерфейс для взаимодействия с пользователями. Они используют знакомые команды, кнопки и меню, что делает использование ботов простым даже для неопытных пользователей.

За последнее время сообщество разработчиков Telegram-ботов значительно увеличилось. Существуют различные онлайн-платформы и ресурсы, предлагающие готовые решения, шаблоны и инструменты для создания собственных ботов без необходимости программирования.

С учетом возросшей популярности Telegram-ботов, они все чаще используются как мощный инструмент для маркетинга, продвижения и взаимодействия с аудиторией. Они могут улучшить эффективность коммуникации бренда с клиентами и предоставлять персонализированный опыт обслуживания.

В целом, история создания и развития Telegram-ботов свидетельствует о постоянной эволюции и инновациях в области мессенджеров. Боты стали неотъемлемой частью нашего повседневного опыта в общении и получении информации, и в будущем их роль, вероятно, будет продолжать расти и развиваться.

1.2 О мессенджере Telegram

Telegram – один из самых популярных мессенджеров в мире, который предлагает своим пользователям множество удобных и защищенных возможностей коммуникации.

Одной из главных особенностей Telegram является его высокий уровень безопасности. Система end-to-end шифрования в Telegram защищает все передаваемые сообщения, а также гарантирует конфиденциальность личной переписки. Благодаря этой функции пользователи могут быть уверены в безопасности своих данных и личной информации.

Еще один преимущественный момент – это открытость платформы. Telegram предоставляет возможность разработчикам создавать собственные боты и приложения, расширяя функционал мессенджера под свои потребности. Это позволяет пользователям настраивать Telegram так, чтобы он отвечал их индивидуальным потребностям и предпочтениям.

Еще одной важной особенностью Telegram является его многофункциональность. Мессенджер предлагает возможность отправки текстовых сообщений, аудио- и видеозаписей, фотографий, файлов и даже голосовых сообщений. Благодаря этим возможностям пользователи могут комфортно общаться друг с другом и делиться информацией в различных форматах.

Telegram также предлагает широкий выбор групп и каналов, где пользователи могут общаться на различные темы, находить новостные и информационные ресурсы по интересующим их областям. Это делает Telegram не только средством личной коммуникации, но и платформой для обмена знаниями и информацией.

Особенно популярной функцией мессенджера Telegram является возможность создания и подключения к криптовалютным кошелькам. Это позволяет пользователям удобно и безопасно хранить и управлять своими криптовалютными активами, а также легко отправлять и получать криптовалютные переводы.

Таким образом, Telegram – это мощный и универсальный мессенджер, который сочетает в себе безопасность, многофункциональность и разнообразные возможности для коммуникации и обмена информацией. Он является незаменимым инструментом для тех, кто ценит свою конфиденциальность и ищет удобные решения для связи и обмена данными.

1.3 О Telegram-ботах

Telegram-боты – представляют собой небольшие программы внутри мессенджера Telegram, которые управляются текстовыми командами, это инновационное средство коммуникации, которое позволяет автоматизировать и упростить множество задач и процессов. Они стали особенно популярными в последние годы благодаря своей удобной и гибкой функциональности.

Telegram-боты предлагают широкий спектр возможностей, начиная от простых задач, таких как напоминания о событиях или получение новостей, до более сложных задач, таких как заказ продуктов или поддержка клиентов. Они также используются для автоматизации процессов бизнеса, помогая сократить расходы на персонал и улучшить общий опыт пользователей.

Один из самых популярных видов Telegram-ботов – это информационные боты. Они предоставляют быстрый доступ к последним новостям, погодным прогнозам, курсам обмена валют и другой полезной информации. Такие боты помогают пользователям быть в курсе событий и экономить время на поиске нужной информации.

Еще один важный вид Telegram-ботов – это коммерческие боты. Они предлагают возможность совершать покупки и заказы без необходимости посещать физический магазин или сайт. Благодаря интеграции с системами оплаты, коммерческие боты обеспечивают удобство и безопасность при покупках онлайн.

Telegram-боты также могут быть использованы в качестве инструментов маркетинга и продвижения бренда. Они могут предоставлять информацию о новых продуктах, проводить опросы или даже организовывать розыгрыши и акции. Такие боты позволяют взаимодействовать с аудиторией и увеличить ее вовлеченность.

Другое преимущество Telegram-ботов – это возможность разработки их собственных ботов с помощью Telegram API. Это позволяет предпринимателям и разработчикам создавать ботов, которые соответствуют их конкретным потребностям и требованиям. Обширная документация и руководства по API делают процесс разработки удобным и доступным.

В целом, Telegram-боты представляют собой мощный инструмент в современном мире коммуникации и бизнеса. Они обеспечивают удобство, экономию времени, автоматизацию процессов и множество других преимуществ для пользователей и организаций. Все это делает их неотъемлемой частью нашей современной цифровой жизни.

2 ВЫБОР ИНСТРУМЕНТОВ ДЛЯ РЕАЛИЗАЦИИ

2.1 Преимущества языка программирования Python для разработки бота

Python является одним из наиболее популярных языков программирования для разработки ботов и есть несколько преимуществ, которые делают его идеальным выбором для этой задачи:

* Простота изучения и использования: Python имеет простой и понятный синтаксис, который делает его очень доступным для новичков. Он использует читаемый английский язык, что помогает разработчикам писать код более естественным образом.
* Большое и активное сообщество: Python имеет огромное сообщество разработчиков, что делает его превосходным выбором для разработки ботов. Здесь вы найдете богатый набор библиотек и фреймворков, которые делают разработку бота быстрой и легкой.
* Возможности веб-разработки: Python имеет мощные фреймворки, такие как Django и Flask, которые упрощают создание веб-основанных ботов. Они предоставляют готовые инструменты для управления запросами по API, создания интерфейса пользователя и многое другое.
* Кроссплатформенность: Python является кроссплатформенным языком программирования, что значит, что ваш бот будет работать на различных операционных системах, таких как Windows, macOS и Linux. Это делает его универсальным выбором для разработчиков.
* Интеграция и интероперабельность: Python обладает богатым набором инструментов для интеграции с другими языками программирования. Это позволяет вам использовать функциональность, предоставляемую другими языками, в вашем боте.

В целом, Python предлагает множество преимуществ для разработчиков ботов, делая их создание быстрым, легким и эффективным. Его простота, мощные инструменты и широкая поддержка сообщества делают его идеальным выбором для начинающих и опытных разработчиков.

2.2 Обоснование выбора среды разработки

PyCharm - это отличная среда разработки для программирования на языке Python. Ее выбор основывается на нескольких причинах:

* PyCharm предлагает много полезных инструментов и функций, которые значительно упрощают процесс разработки. Например, он имеет удобный редактор кода с функцией автодополнения и подсветкой синтаксиса, что позволяет программисту писать код более эффективно и быстро.
* PyCharm обладает мощными инструментами для отладки и профилирования кода. Когда возникают ошибки, PyCharm позволяет легко отслеживать и исправлять их с помощью интуитивного интерфейса и функции построчного выполнения кода.
* Cреда разработки PyCharm предлагает обширный набор интегрированных инструментов для управления проектом. Он поддерживает системы контроля версий, такие как Git, и предоставляет возможность работать со сторонними инструментами и библиотеками.
* Преимуществом PyCharm является его расширяемость. PyCharm предоставляет API, позволяющий разработчикам создавать свои собственные плагины и интегрированные инструменты. Это позволяет программистам настроить среду разработки в соответствии со своими потребностями.
* PyCharm имеет активную и дружественную сообщество пользователей, где можно получить помощь, задать вопросы и найти интересные решения задач. Также существуют множество онлайн-ресурсов и учебных материалов, посвященных использованию PyCharm.

В целом, выбор PyCharm в качестве среды разработки для Python основан на его мощных инструментах, расширяемости, поддержке и активному сообществу. PyCharm значительно облегчает процесс программирования и позволяет создавать качественные и эффективные программы.

3 РАЗРАБОТКА БОТА

3.1 Установка и настройка среды разработки PyCharm

Перед началом разработки необходимо установить выбранную интегрированную среду разработки. Нами была выбрана среда PyCharm Commnity Edition 2023.

После запуска установленной среды был создан новый проект на базе языка программирования Python. Пример создания проекта представлен на рисунке 1.

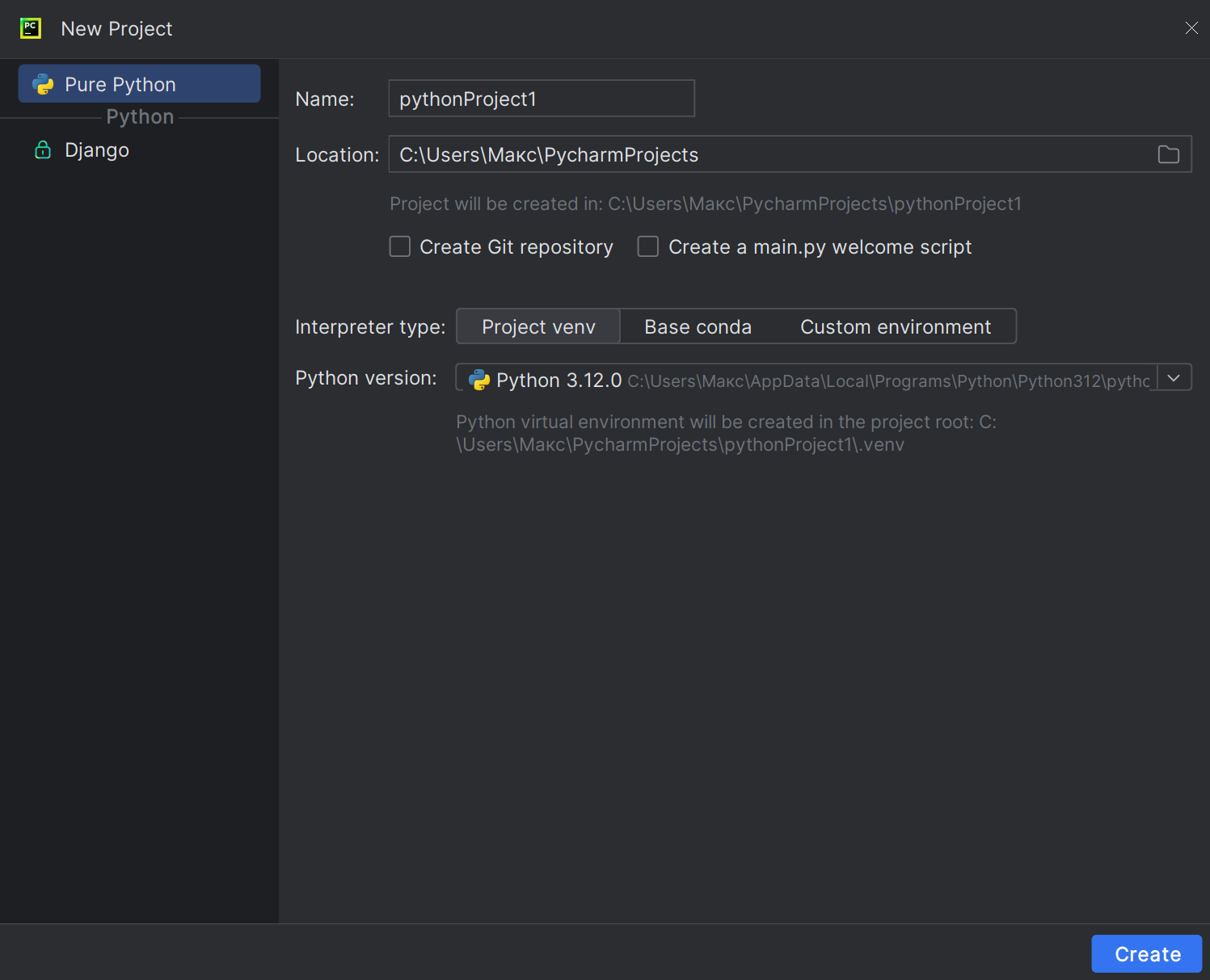


Рисунок 1 – создание проекта в PyCharm Commnity Edition 2023

После создания проекта были установлены следующие основные библиотеки: aiogram, requests, beautifulsoup4. Полный список библиотек представлен на рисунке 2.

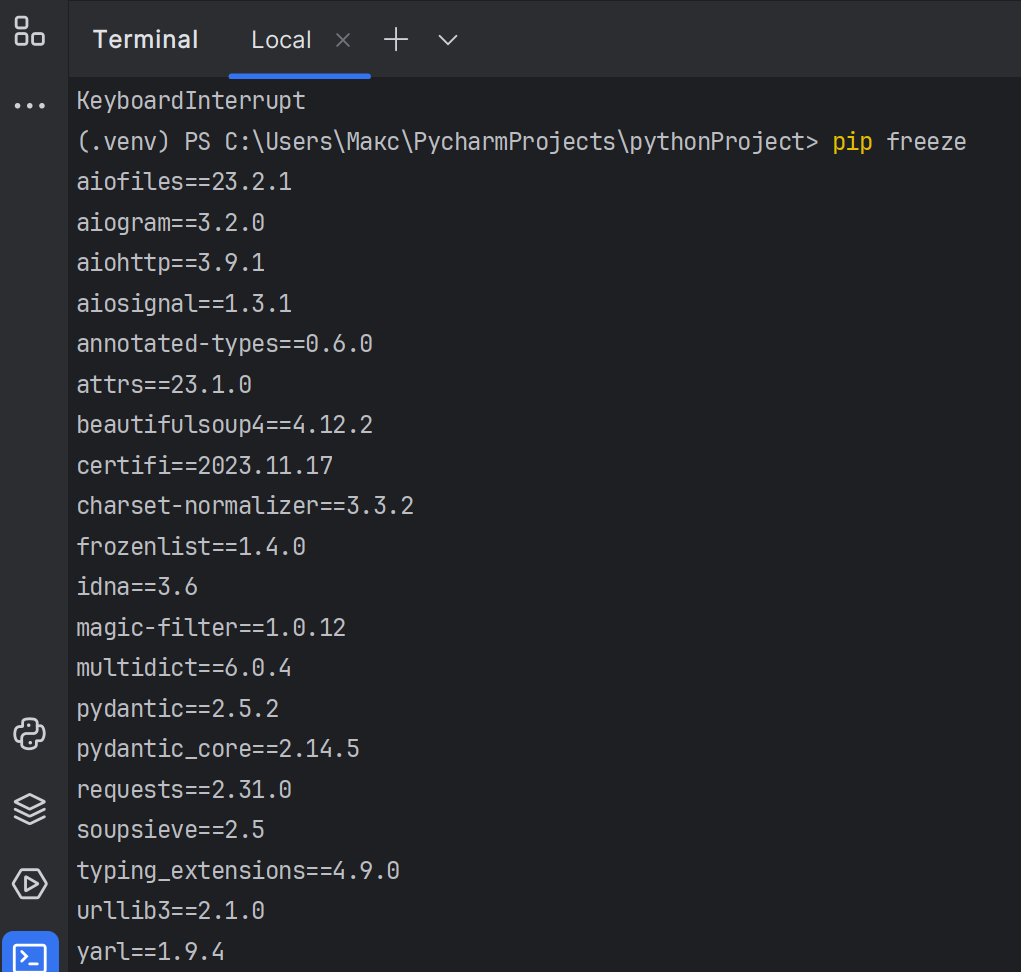


Рисунок 2 – список установленных библиотек проекта

## 3.2 Регистрация и настройка бота в Telegram

Важным шагом в разработке бота является регистрация у специального чат-бота «BotFather». Регистрация начинается с команды «/newbot», после чего предлагается ввести название бота в соответствии с условиями. При успешной регистрации «BotFather» выдает токен и URL-адрес для доступа к боту. Пример регистрации чат-бота представлен на рисунке 3.

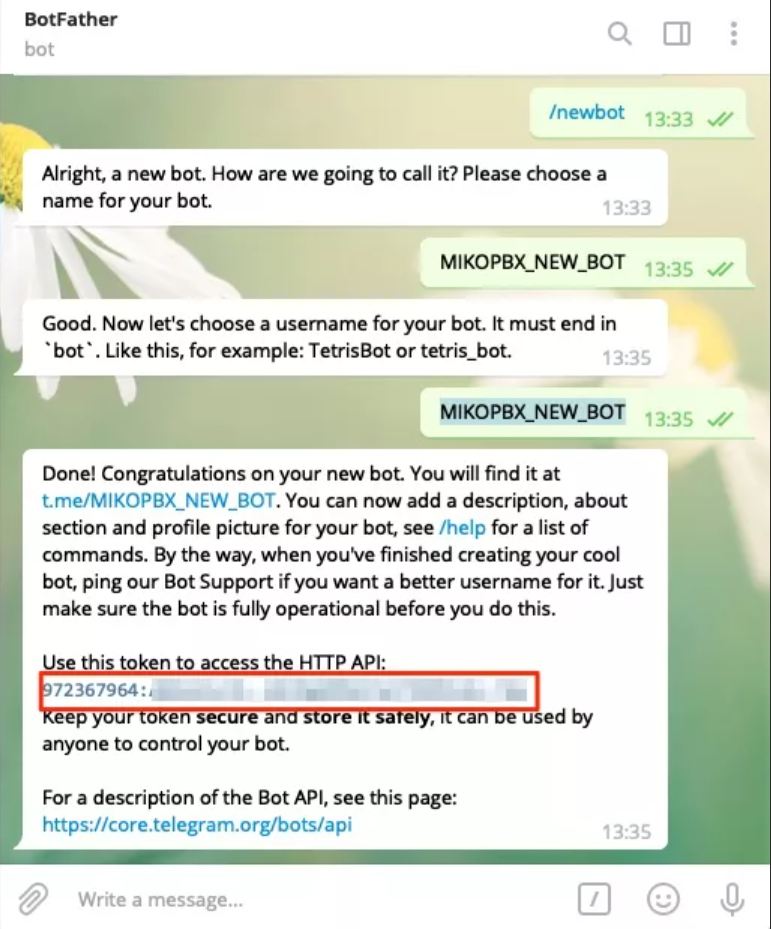


Рисунок 3 – Регистрация бота в Telegram

Для установки дополнительных параметров, существуют следующие команды (таблица 1)

Таблица 1 – Доступные команды для модификации бота

|  |  |
| --- | --- |
| Команда | Описание |
| /setname | Изменяет имя бота |
| /setdescription | Изменяет текст, отображающийся при первом открытии бота |
| /setabouttext | Изменяет текст в поле «О чат-боте» |
| /setuserpic | Изменяет фотографию профиля бота |
| /setcommands | Изменяет список команд бота |
| /deletebot | Удаляет бота |
| /token | Генерирует новый токен авторизации |
| /revoke | Отзывает токен доступа к боту |
| /setinline | Переключает встроенный режим возможности вызова бота из других чатов |
| /setinlinegeo | Переключает встроенные запросы местоположения |
| /setinlinefeedback | Изменение настроек обратной связи |
| /setjoingroups | Регулирует возможность добавления бота в другие чаты |
| /setprivacy | Переключение режима конфиденциальности в группах |

После успешной регистрации бота в Telegram и получении токена можно приступить к разработке программной части бота.

## 3.3 Разработка Telegram-бота

Для реализации Telegram-бота в проекте был создан файл BotMain.py, являющийся главным и единственным в проекте.

В созданном файле в начале реализуется подключение полученного API, необходимого для взаимодействия с Telegram-ботом. Реализация представлена в листинге 1.

Листинг 1 – Подключение API для взаимодействия с ботом

import asyncio

import logging

import requests

import random

from aiogram import Bot, Dispatcher, types

from aiogram.types import reply\_keyboard\_markup, keyboard\_button,reply\_keyboard\_remove

from aiogram.filters import CommandStart, Command

from bs4 import BeautifulSoup as bsoup

# Идентификатор для бота и ссылка на сайт с анекдотами

TOKEN\_API = "6842064304:AAFvCfPNL1SH9UVvLOZzoeHfoXbHXa1yBH0"

URL = 'https://www.anekdot.ru/last/good/'

# Передача токена боту и добавление диспетчера для дальнейшей обработки команд

bot = Bot(token=TOKEN\_API)

Далее реализуется первая команда бота «/start», с которой начинается весь процесс взаимодействия с ботом. В эту команду добавлена клавиатура для удобства написания последующих реализованных команд, а также приветствие пользователя и дальнейшие указания. Реализация представлена в листинге 2.

Листинг 2 – Реализация команды «/start»

# Функция реагирования на команду start

@dp.message(CommandStart())

async def handle\_start(message: types.Message):

kb = [

[

types.KeyboardButton(text="/help"),

types.KeyboardButton(text="/description"),

types.KeyboardButton(text="/joke"),

],

]

keyboard = types.ReplyKeyboardMarkup(

keyboard=kb,

resize\_keyboard=True,

input\_field\_placeholder="Для вывода списка команд введите /help."

)

await message.answer(text=f"Здравствуйте, {message.from\_user.full\_name}!\nДля вывода списка команд введите <b>/help</b>.",

parse\_mode="HTML",

reply\_markup=keyboard)

Далее реализуются ещё две команды: «/description», которая отображает краткую информацию о боте, и «/help», выводящая справку по всем командам бота. Реализация представлена в листинге 3.

Листинг 3 – Реализация команд «/description» и «/help»

# Функция реагирования на команду help

@dp.message(Command("help"))

async def help\_cmd(message: types.Message):

await bot.send\_message(chat\_id=message.from\_user.id,

text=HELP\_COMMAND,

parse\_mode="HTML",

disable\_web\_page\_preview=1)

# Функция реагирования на команду description

@dp.message(Command("description"))

async def desc\_cmd(message: types.Message):

await bot.send\_message(chat\_id=message.from\_user.id,

text=DESC\_COMMAND,

parse\_mode="HTML")

На этом этапе реализуется первый парсер для следующей команды бота «/joke». Парсер принимает ссылку на сайт с анекдотами, а возвращает список анекдотов с главной страницы сайта, после чего список перемешивается. Реализация парсера представлена в листинге 4.

Листинг 4 – реализация парсера для команды «/joke»

def parser(url):

r = requests.get(url)

soup = bsoup(r.text, "html.parser")

anekdots = soup.find\_all('div', class\_='text')

return [c.text for c in anekdots]

# Передача списка анекдотов и перемешивание их

jokes\_list = parser(URL)

random.shuffle(jokes\_list)

Далее реализуется парсер сайта с товарами, который по ключевому слову или фразе выдаёт 10 товаров из категории. Структура представлена в листинге 5.

Листинг 5 – Парсер сайта с товарами

# Функция-парсер для нахождения товаров по ключевому слову

@dp.message()

async def parser(message: types.message):

url = "https://n-katalog.ru/search?keyword=" + message.text

request = requests.get(url)

soup = bsoup(request.text, "html.parser")

all\_links = soup.find\_all('a', class\_='model-short-title')

for link in all\_links:

url = "https://n-katalog.ru" + link["href"]

request = requests.get(url)

soup = bsoup(request.text, "html.parser")

name = soup.find("div", class\_="page-title").text

img = soup.find("div", class\_="img200")

img = img.findChildren("img")[0]

img = "https://n-katalog.ru/" + img["src"]

await bot.send\_photo(message.chat.id, img,

caption="<b>" + name + "</b>\n<i>" + f"</i>\n<a href='{url}'>Ссылка на сайт</a>",

parse\_mode="html")

if all\_links.index(link) == 9:

break

Для того, чтобы бот в режиме реального времени реагировал на происходящие события, происходит постоянный опрос событий. Реализация опроса представлена в листинге 6.

Листинг 6 – Реализация опроса событий

# Постоянный опрос обновлений бота + поверхностное логирование

async def main():

logging.basicConfig(level=logging.INFO)

await dp.start\_polling(bot)

# Точка входа

if \_\_name\_\_ == '\_\_main\_\_':

asyncio.run(main())

4 ТЕСТИРОВАНИЕ

Для начала работы с ботом необходимо перейти по ссылке, где находится бот (t.me/ItsCloudyTheBot), и запустить его командой «/start». Данное взаимодействие показано на рисунке 4.

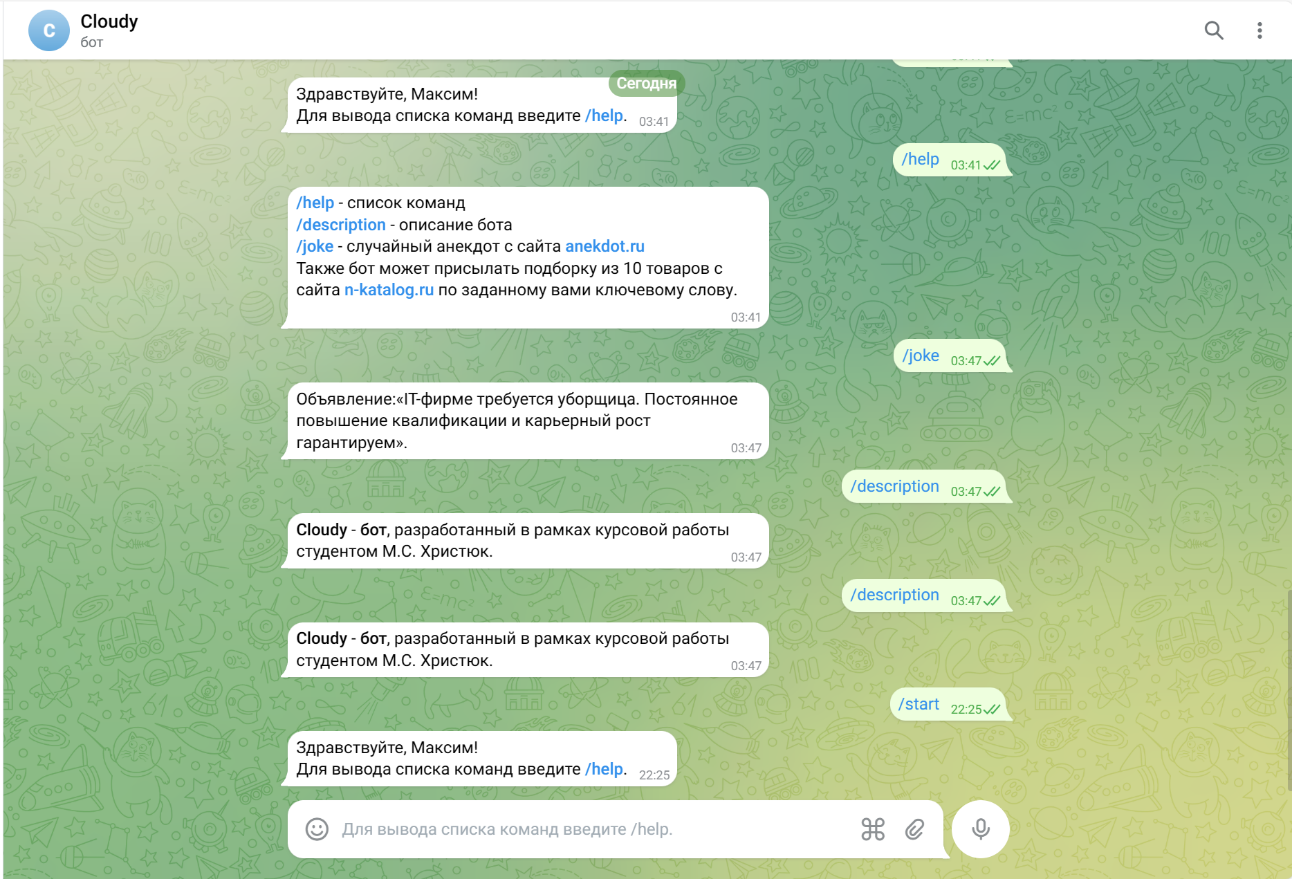


Рисунок 4 – Взаимодействие с ботом, команда «/start»

После запуска бота для пользователя становится доступна клавиатура с набором команд, представленная на рисунке 5.

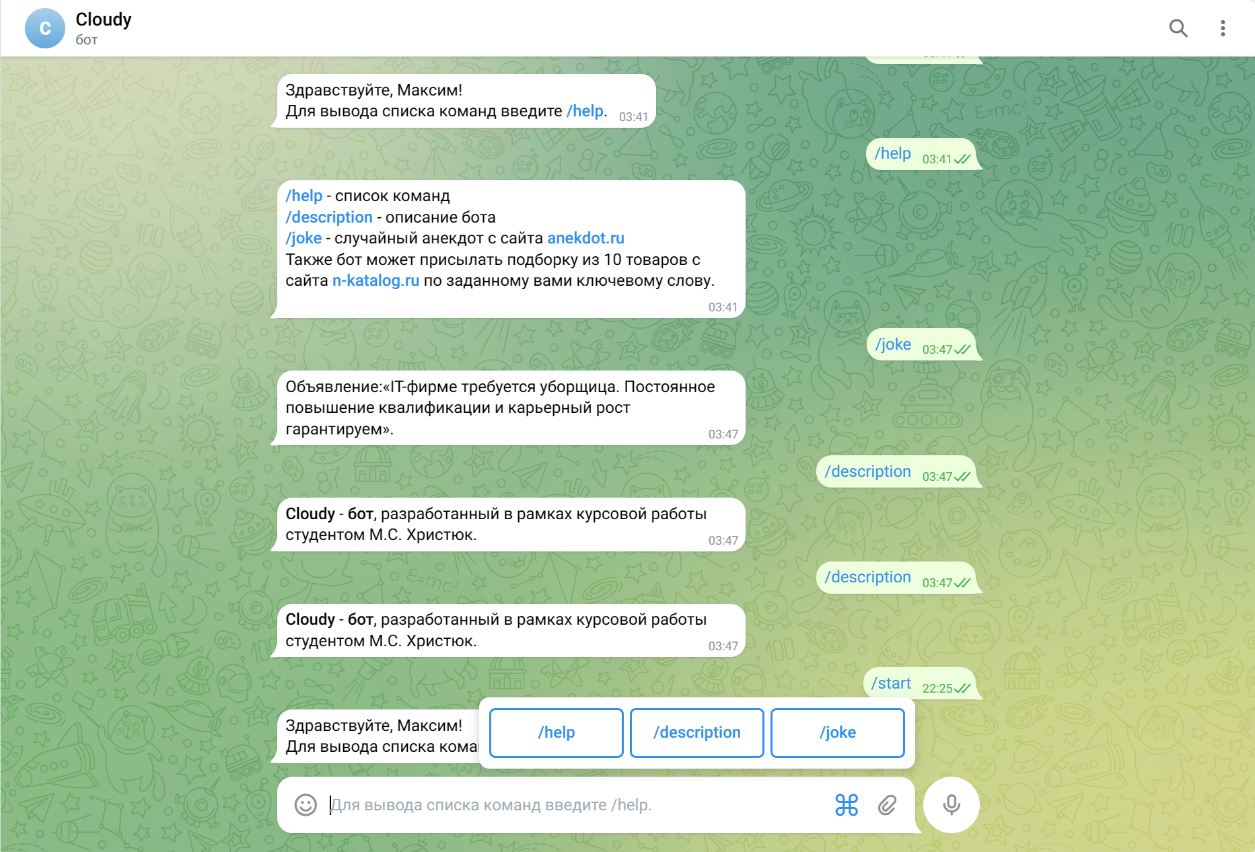


Рисунок 5 – Клавиатура бота

Далее пользователю рекомендуется ознакомиться со списком команд бота. На рисунке 6 продемонстрирована работа команд «/help» и «/description».

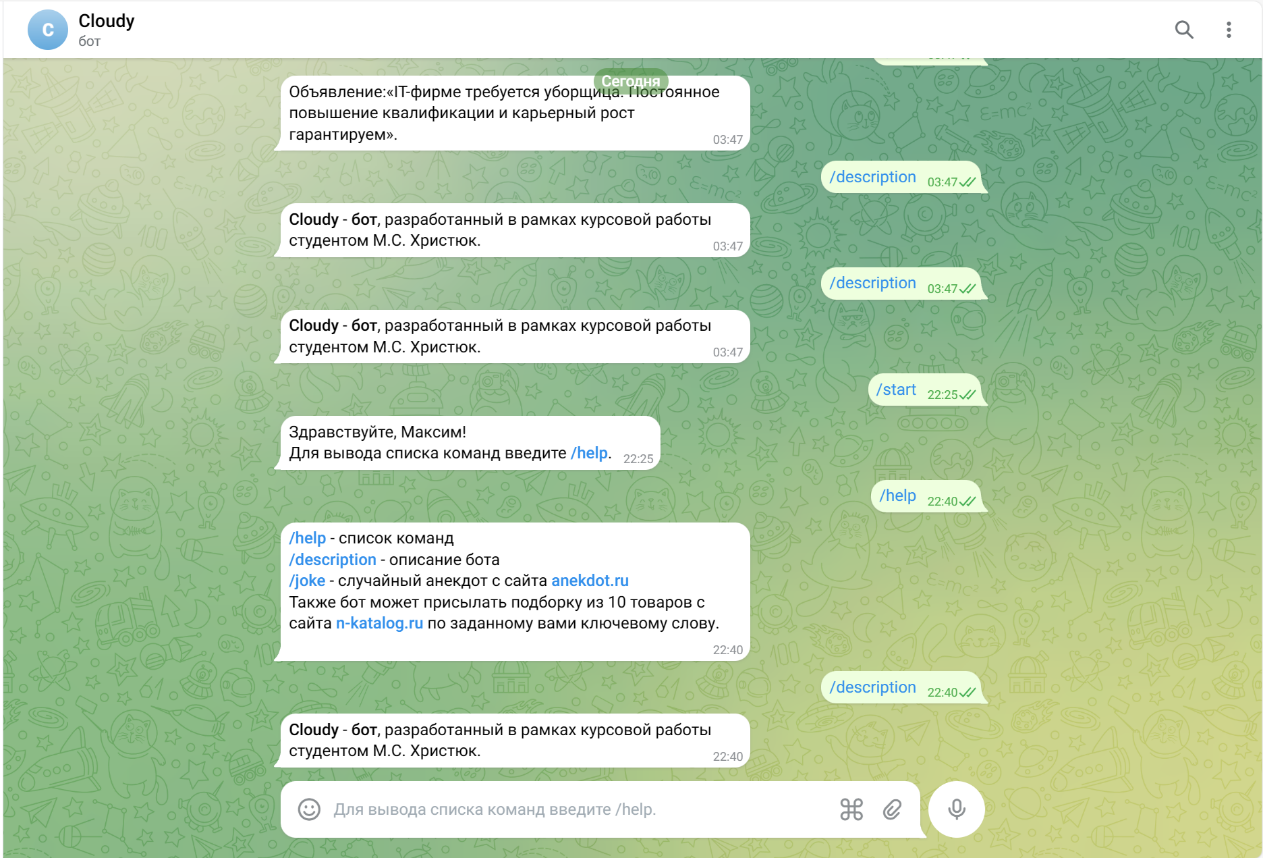


Рисунок 6 – Демонстрация команд «/help» и «/description»

Пользователю также доступна команда «/joke», которая отправляет в чат случайный анекдот с сайта. Демонстрация команды представлена на рисунке 7.

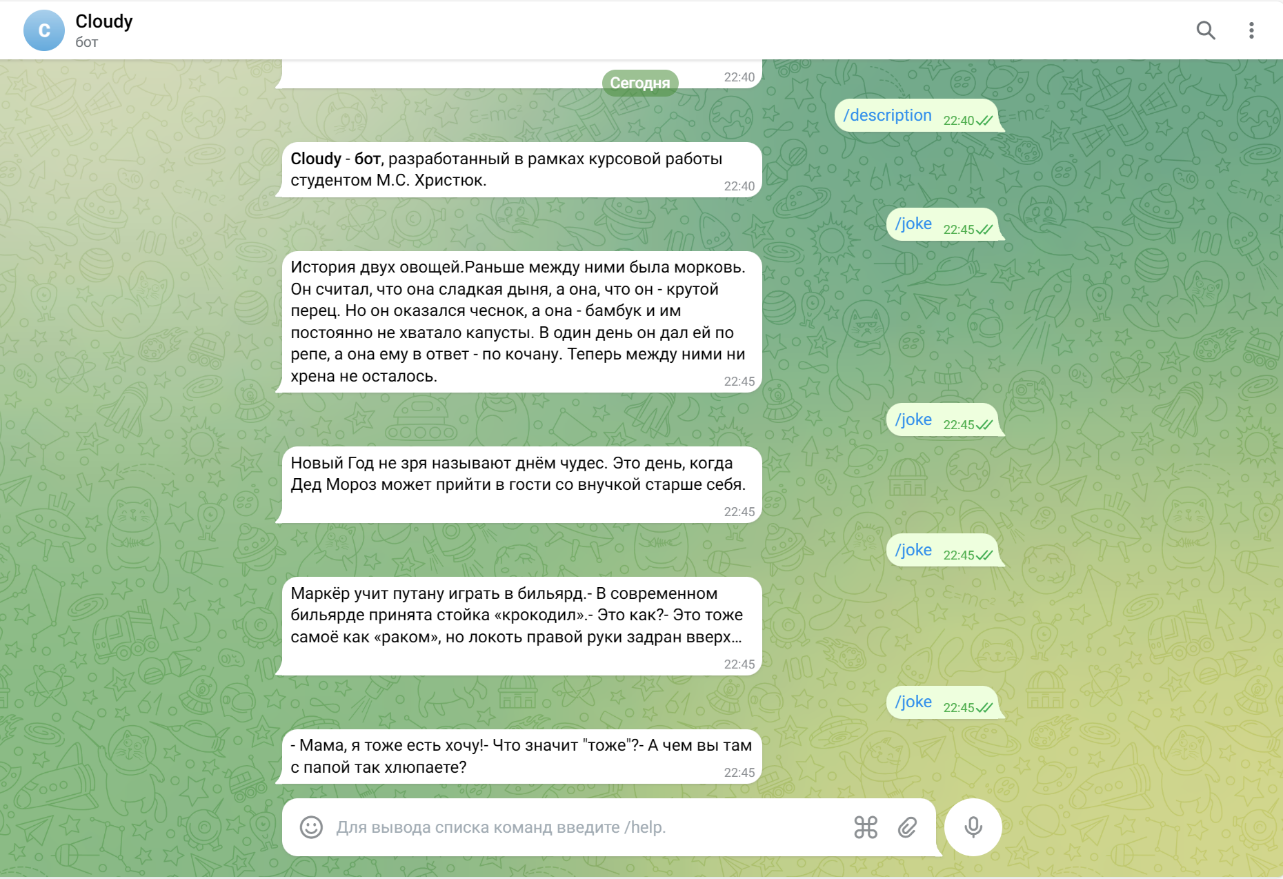


Рисунок 7 – Демонстрация команды «/joke»

Пользователь может отправить ключевое слово или фразу боту. В этом случае он найдёт на сайте товаров подборку из 10 товаров из категории. Работа данной функции продемонстрирована на рисунке 8.

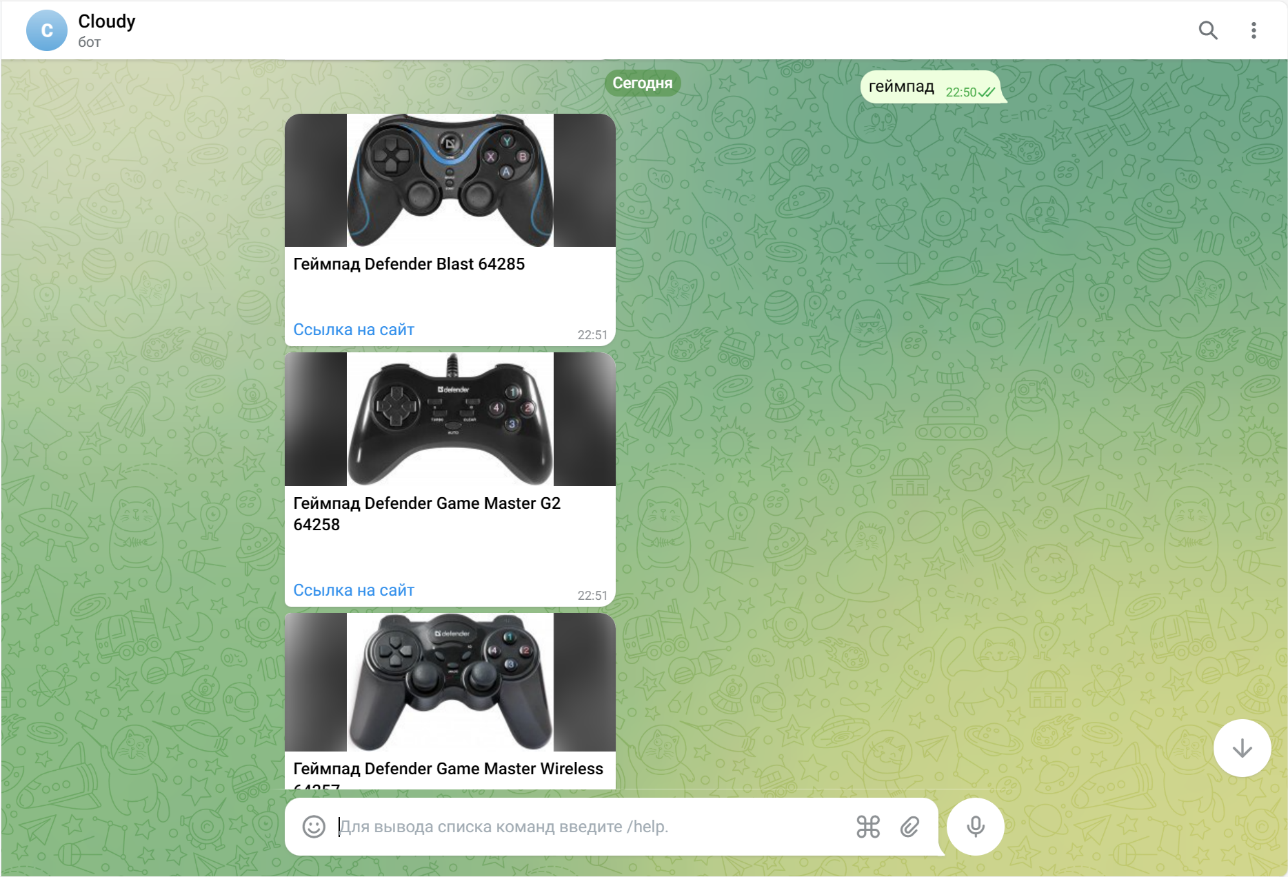


Рисунок 8 – Демонстрация функции поиска товаров

# ЗАКЛЮЧЕНИЕ

В ходе данной работы выполнена разработка многофункционального Telegram-бота.

Бот удовлетворяет следующим основным требованиям:

* Язык программирования: Python.
* Библиотека: aiogram.
* Среда разработки: PyCharm.
* Удобный пользовательский интерфейс.
* Получение данных от пользователя.
* Обработка полученных данных.
* Отправка результатов пользователю.

Были решены следующие задачи:

* Проведено изучение основ телеграмм ботов и принципов их разработки.
* Определены требования к разрабатываемому Telegram-боту, выбран язык и инструменты для реализации.
* Выполнена разработка и тестирование бота.

В дальнейшем можно доработать данного бота: добавить вывод прогноза погоды, создание опросов, улучшить работу парсеров и детализацию поиска. Также можно добавить мини-игры, например, крестики-нолики.

СПИСОК ЛИТЕРАТУРЫ

1. aiogram: сайт. – URL: https://aiogram.dev/ (дата обращения 9.12.2023)
2. Beautiful Soup Documentation: сайт. – URL: https://beautiful-soup-4.readthedocs.io/en/latest/ (дата обращения 10.12.2023)
3. Requests: HTTP for Humans: сайт. – URL: https://requests.readthedocs.io/en/latest/index.html (дата обращения 10.12.2023)
4. История появления телеграм ботов: сайт. – URL: https://treskamurmansk.ru/istoriya-poyavleniya-telegram-botov/?ysclid=lq0zwwuapb993162335 (дата обращения 11.12.2023)

ПРИЛОЖЕНИЕ

GitHub [сайт]. – URL: https://github.com/Paprika-RU/stash/tree/main/pythonProject