

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК, СЕКЦІЇ ІКТ

ВИПУСКНА РОБОТА

на тему:

**«Чат-бот для telegram.
Віртуальний співрозмовник та
розумний помічник.»**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Бабій М.С.

Студента групи ІН-64-8

Шайдаров Є.В.

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК, СЕКЦІЇ ІКТ

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2020 р.

ЗАВДАННЯ
до випускної роботи

Студента четвертого курсу, групи ІН-64-8 спеціальності “Інформатика”
денної форми навчання Шайдарова Євгенія Віталійовича.

Тема: “ Чат-бот для telegram. Віртуальний співрозмовник та розумний помічник. ”

Затверджена наказом по СумДУ

№ _____ от _____ 2020 р.

Зміст пояснювальної записки: 1) огляд існуючих рішень; 2) вибір методу рішення; 3) інформаційне та програмне забезпечення системи.

Дата видачі завдання “ _____ ” _____ 2020 р.

Керівник випускної роботи _____ Бабій М.С.

Завдання прийняв до виконання _____ Шайдаров Є.В.

РЕФЕРАТ

Записка: 32 сторінок, 32 рисунків, 8 джерел.

Об'єкт дослідження — Чат-боти для telegram.

Мета роботи — Розробка корисного та цікавого чат-бота.

Методи дослідження — аналіз існуючих рішень, моделювання роботи чат-бота, тестування розробленої програми.

Результати — Створено чат-бот для спілкування з користувачами. Розроблений алгоритм пріоритету відповідей на вхідні повідомлення. Розроблений продукт реалізовано у формі сервера, створеного за допомогою інструментального програмного середовища PyCharm Professional Edition.

ЧАТ-БОТ, TELEGRAM, WEBHOOK, SSL З'ЄДНАННЯ, ПАРСИНГ
ІНТЕРНЕТ-СТОРИНОК, TELEGRAM-BOT.

ЗМІСТ

ВСТУП.....	5
1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	7
1.1 <i>Існуючі рішення.....</i>	<i>7</i>
1.2 <i>Постановка задачі.....</i>	<i>11</i>
2. ВИБІР МЕТОДУ РІШЕННЯ.....	13
2.1 <i>Вибір мови програмування.....</i>	<i>13</i>
2.2 <i>Вибір середовища розробки.....</i>	<i>14</i>
2.3 <i>Реєстрація.....</i>	<i>14</i>
2.4 <i>Методи отримання даних.....</i>	<i>15</i>
2.5 <i>Вибір фреймворку.....</i>	<i>16</i>
2.6 <i>Вибір серверу розміщення.....</i>	<i>17</i>
2.7 <i>Вибір системи контролю версій.....</i>	<i>18</i>
3. ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	19
3.1 <i>Розробка.....</i>	<i>19</i>
3.2 <i>Робота з базою даних.....</i>	<i>23</i>
3.3 <i>Розгортка на сервері.....</i>	<i>23</i>
4. Презентація проекту.....	26
ВИСНОВКИ.....	31
СПИСОК ЛІТЕРАТУРИ.....	32

ВСТУП

На даний момент інформаційні технології розвиваються величезними кроками. Буквально кожного дня робиться внесок на вдосконалення та приріст якості технологій. На разі ми маємо багато можливостей, про які, нещодавно, могли тільки мріяти. Напевно, немає такої інформації, яку не можна було б знайти на просторах інтернету.

Сьогодні буквально весь світ так чи інакше спілкується в соціальних мережах, або так званих месенджерах. Такої популярності вони набули за великий перелік можливостей комунікації, та розваг. Тут і спілкування з далекими родичами, і знайомства з новими людьми, і обмін різного роду інформацією і так далі.

Що ж собою представляє соціальна мережа?

Насамперед це структура, для соціальної комунікації. Якщо бути точнішим, то за допомогою соціальних мереж кожен має змогу спілкуватися будь з ким, з різних точок світу, ділитися фотографіями, відео, та інше. Також користувач має змогу користуватися великим вибором інтерактивних послуг, переглядати спільноти, грати в ігри і так далі.

Що таке месенджер?

Якщо говорити простою мовою, то це платформа для спілкування. Точніше для обміну повідомленнями, відео, фото, музикою та різноманітними файлами. Також великою популярністю месенджери користуються в колективній роботі, коли потрібно зробити опитування, або проведення дискусії. Також вагому частку можна приділити користуванню месенджерами в рамках дистанційного навчання. Наприклад для зв'язку з викладачем, або для отримання навчальної літератури.

Найпопулярнішим месенджером, на сьогодні, є Telegram. І це не дивно, тому що окрім стандартного обміну повідомленнями в діалогах і групах,

користувач має змогу зберігати необмежену кількість файлів, вести канали, робити власні «стікери», а також створювати і використовувати ботів.

Особливу увагу варто звернути на ботів у Telegram. Це такого собі роду програма-співрозмовник, що імітує людське спілкування за допомогою тексту, або голосу. Боти можуть вести діалог з користувачем, виконувати його прохання, відповідати на запити, або розважати своїми відповідями. Чат-боти багатозадачні.

За запитами користувача вони видають корисну інформацію: прогноз погоди, курс валют, і так далі. Боти-помічники нагадують про майбутні події. Боти поштових служб відслідковують посилку по трек-коду. Чат-Боти полегшують пошук і завантаження книг, фільмів і музики.

Існує безліч різноманітних ботів-помічників так як і чат-ботів. Всі вони володіють найрізноманітнішим функціоналом починаючи від доволі корисного, пізнавального і до розважального змісту.

1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

Зараз існує безліч різноманітних ботів, як корисних, розважальних так і безглузких. Занирнувши глибше у недра, можна дізнатися, що їх кількість значно більша ніж здається на перший погляд. Також стає зрозумілою актуальність розробки чат-бота, адже можливості розробки практично безмежні.

1.1 Існуючі рішення

Серед великої кількості досліджених ботів знайшлася вагома частка дійсно достойних уваги ботів, які неодмінно допоможуть користувачеві тим чи іншим чином. Слід виділити декілька рішень, які, я вважаю необхідним виділити.

Корисні чат-боти:

- @ImageSearchBot
- @weatherman_bot
- @Instasave_bot
- @tap2bot
- @MinfinBot
- @MinfinBot
- @railwaybot
- @QTrackerBot
- @PrivatBankBot

Чат-боти, які хотілось би виділити:

- @Grocerylistbot
- @AndyRobot
- @Zdorobot

Чат-бот може бути корисним помічником у повсякденних домашніх справах, як наприклад :

@**Grocerylistbot** – бот-помічник, який запам'ятовує список потрібних вам покупок, або справ, які необхідно зробити.(рис. 1.1)

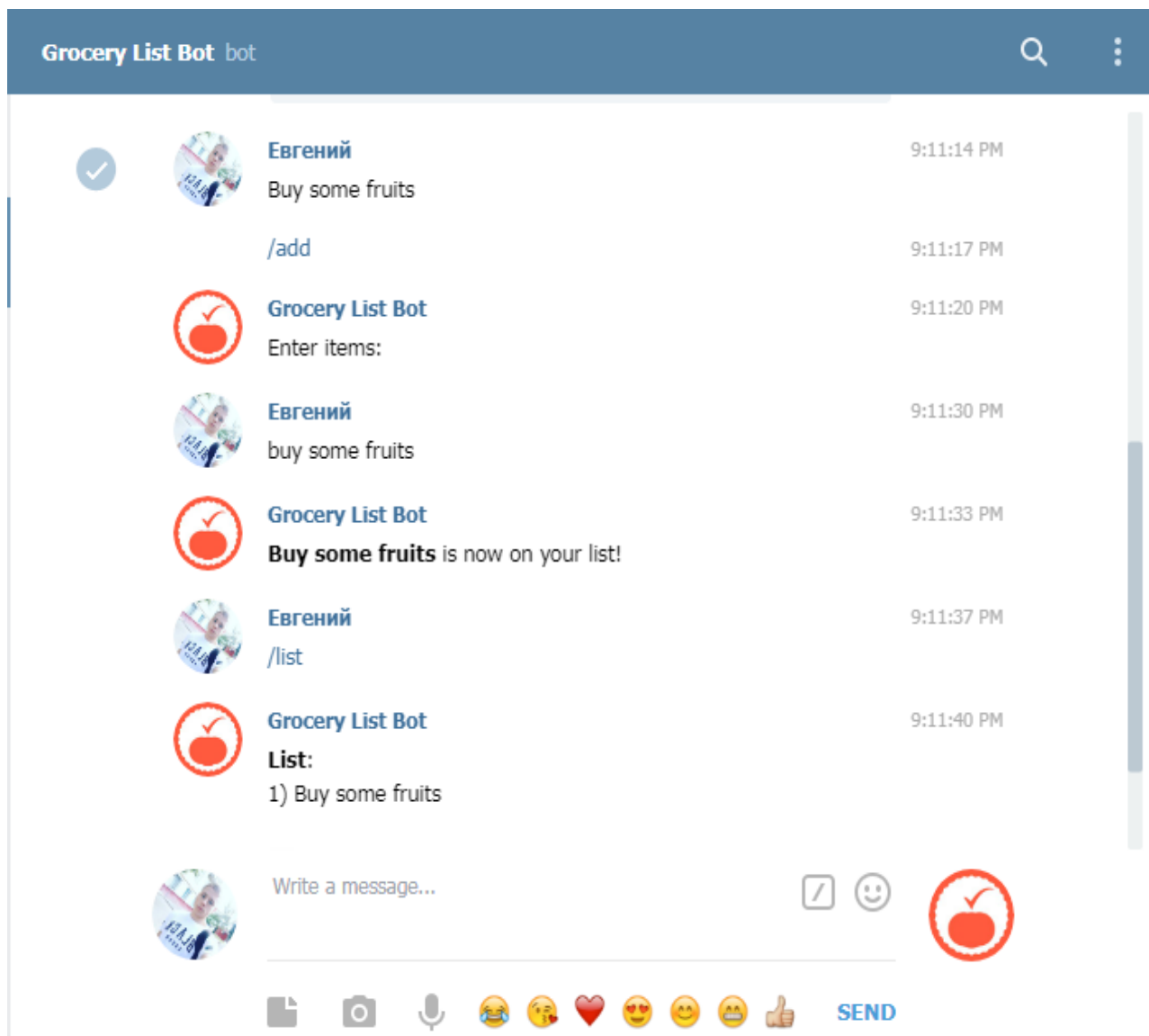


Рисунок 1.1 - @Grocerylistbot

Можуть допомагати у вивченні іноземних мов як:

@AndyRobot - чат-бот, який допоможе вам з вивченням Англійської.(рис. 1.2)

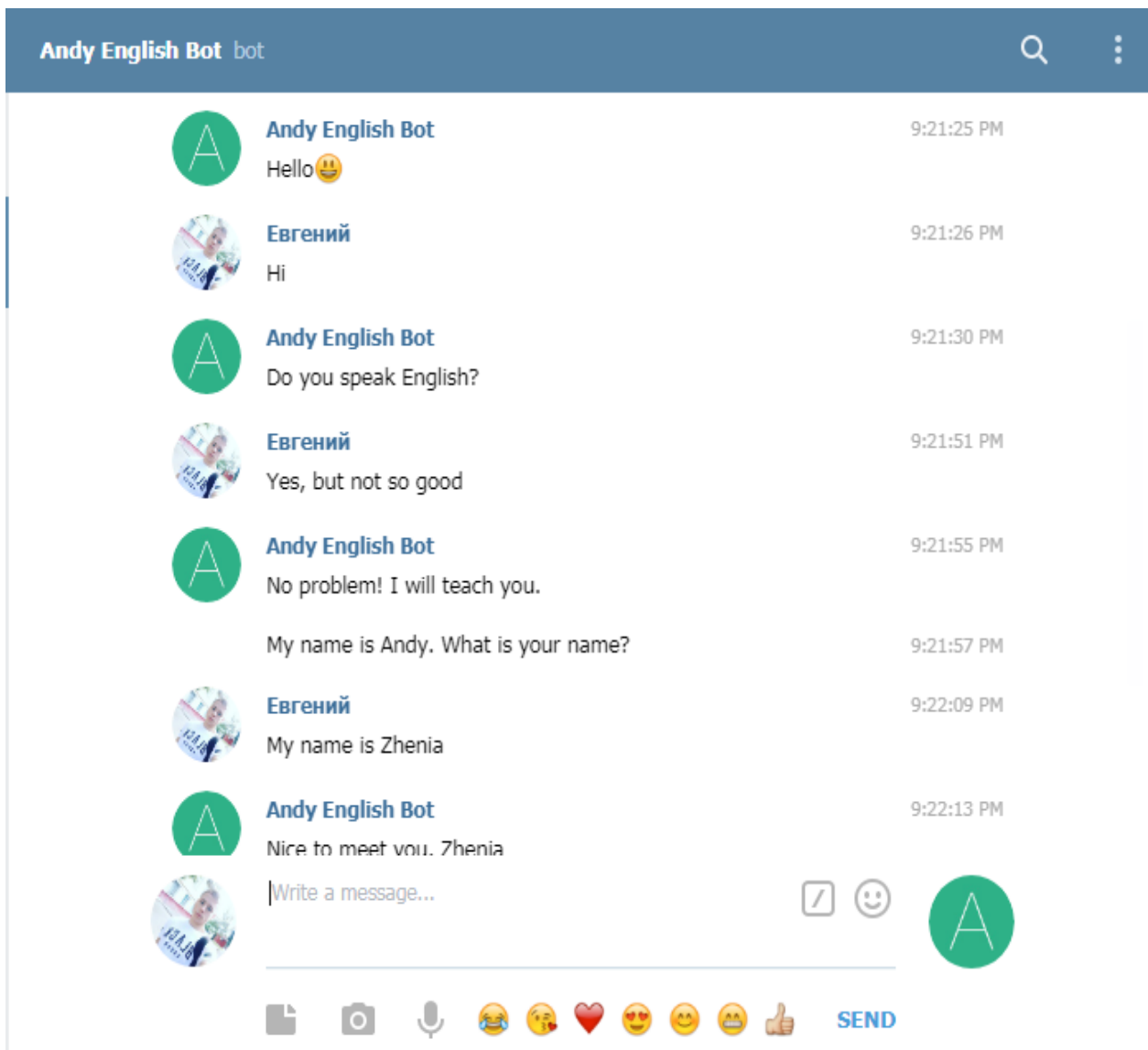


Рисунок 1.2 - @AndyRobot

Навіть записати на прийом до лікаря за симптомами, та поставити передбачуваний діагноз хвороби:

@Zdorobot – це віртуальний лікар, який може проаналізувати ваші симптоми, призначити огляд у лікаря, підказати найближчу аптеку з необхідними ліками.(рис. 1.3)

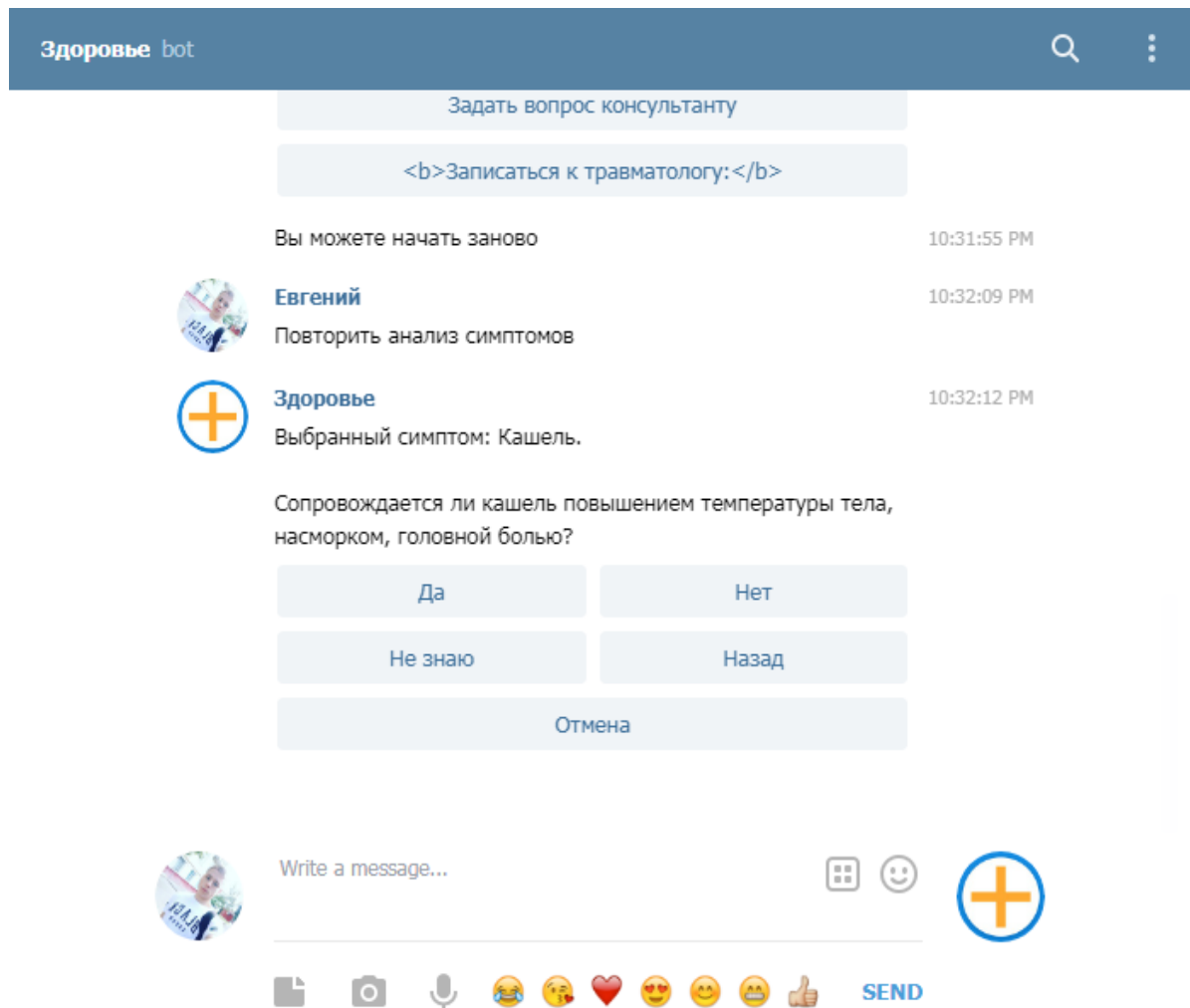


Рисунок 1.3 - @Zdorobot

Таким чином боти так чи інакше допомагають користувачам в різних сферах діяльності, розважають, нагадують, навчають та інше.

Розробка подібних додатків користується великим попитом, так як користувач стає все вибагливішим, і необхідно шукати нові шляхи для приваблення публіки до свого контенту.

На мою думку непогано було б реалізувати чат-бота та бота помічника в одному додатку. З одного боку цікавий співрозмовник, з іншого корисний помічник.

Саме ця ідея слугувала поштовхом до зацікавлення розробкою чат-ботів та ботів помічників.

Після моніторингу інформації щодо інструментарію різних ботів, а також методів розробки та втілення проекту для масового користування, було взято за мету розробити віртуального співрозмовника та розумного помічника у одному проекті.

1.2 Постановка задачі

Метою роботи є створення чат-бота для одночасно розважального та корисного змісту.

Вимоги до проекту:

- Ефективна інтерактивна та корисна система спілкування типу користувач – бот;
- Створення діалогів розважального характеру для заохочення до користування додатком;
- Створити комфортні умови, де користувач зможе отримувати корисну інформацію необхідну саме йому.

Проект призначений для:

- Спілкування з користувачами(розважальний характер розмови);
- Нагадування користувачеві необхідної інформації;
- Допомога у пошуку різноманітної інформації;
- Деякі інтерактивні можливості.

Виходячи з призначення проекту маємо представлення про доцільність створення даного проекту. На мою думку бот буде корисним для великого кола користувачів переважно середнього віку.

2. ВИБІР МЕТОДУ РІШЕННЯ

Для створення ботів в telegram є декілька шляхів. Одним із них є написання серверної частини програмуванням на одній із мов програмування. Інший шлях – це створення без навичок програмування за допомогою конструкторів ботів, таких як, наприклад, Manybot.io. Завдяки цьому конструктору користувач може створити бота за лічені хвилини не прикладаючи значних зусиль.

Кожен з методів має свої суттєві переваги, але основною задачею для створення є його функціональність. В цьому випадку створення методом програмування має суттєву перевагу, так як створення бота не буде обмежуватися існуючими інструментами конструктора.

2.1 Вибір мови програмування

Існує безліч різноманітних мов програмування. Як обрати?

Критерії для вибору мови програмування:

- Підтримка фреймворків(бібліотек) для роботи запитами;
- Підтримка бібліотеки для роботи з Telegram Bot;
- Можливість розміщення програмного коду на сервері;
- Універсальність;
- Постійна підтримка;
- Сучасність.

При виборі мови програмування мою увагу привабив Python 3.7.

Ця мова має у своєму арсеналі повний набір необхідних згідно з вимогами можливостей. Її особливість це націленість в першу чергу на підвищення продуктивності самого програміста, ніж коду який він пише. Код лаконічний і зрозумілий навіть тому, хто ніколи на ньому не писав.

2.2 Вибір середовища розробки

Для вибору середовища розробки було необхідним оглянути ринок реалізацій для розробки на Python. Під час моніторингу було зроблено висновки про зручність та надійність продуктів компанії Jet Brains, а саме продукту «PyCharm» для розробки на мові Python.

Продукти цієї компанії мають велику популярність у світі програмування та користуються чималим попитом. Це пояснюється великою кількістю інструментів, необхідних розробнику. У доповненні зі швидкістю та зручністю у використанні, середовища від Jet Brains мають заслужену позицію лідера на ринку ідеа.

Таким чином для розробки програмного продукту середовищем розробки було обрано пакет PyCharm 2018 Professional Edition.

2.3 Реєстрація

Для створення боту спочатку необхідно зареєструвати його в Telegram. Дана процедура виконується за допомогою спеціального боту «BotFather».(рис. 2.1)

Реєстрація проходить в декілька кроків:

- Введення команди /newbot;
- Введення назви майбутнього боту;
- Введення імені бота(Унікальний ключ, за яким буде проходити пошук боту)
- Налаштувати сторінку за допомогою влаштованих інструментів.

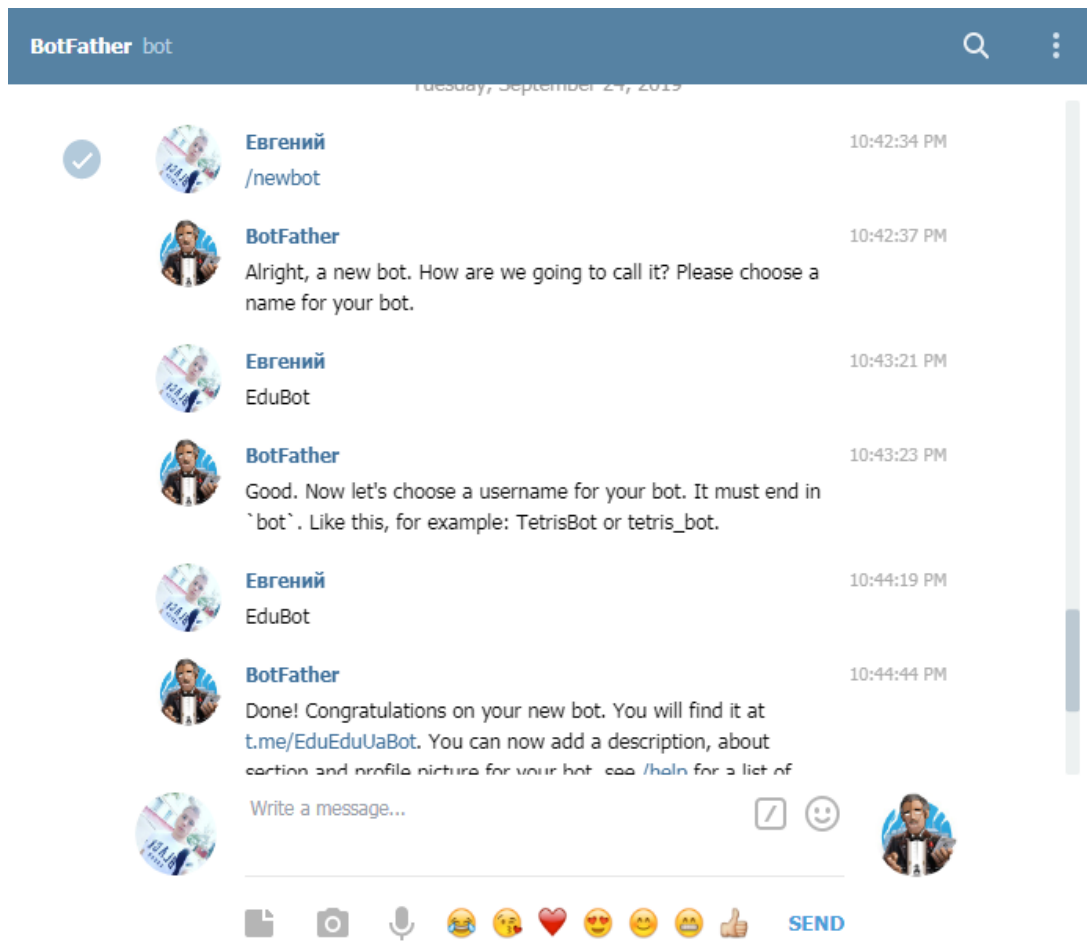


Рисунок 2.1 – Реєстрація

2.4 Методи отримання даних

Існує декілька методів отримати дані від телеграм-серверу:

- За допомогою методу `getUpdates`;
- За допомогою `Webhook`.

Метод `getUpdates` передбачає, що програма періодично буде надсилати запити до сервера, тим самим перевіряти на вхідні повідомлення від користувачів. (рис. 2.2)

- Flask
- Django

Обираючи між цими фреймворками необхідно враховувати необхідний функціонал, швидкодію, та простоту використання.

Для обробки запитів від серверів telegram було вирішено використати фреймворк Flask для Python, так як його інструментарій повністю задовольняє вимоги до необхідних можливостей роботи з запитами.

Вибір саме цього фреймворку пояснюється простим ядром, та простотою самого фреймворку, тим часом з повним набором функціоналу.

2.6 Вибір серверу розміщення

Для безвідмовної працездатності бота, було необхідним розміщення його на одному із серверних веб-ресурсів.

Вимоги до серверу:

- Підтримка розміщення додатків на Python 3.7
- Підтримка роботи з базами даних
- Безкоштовне розміщення додатку(бажано)

Після моніторингу ринку серверів за даними вимогами було обрано іноземний сервер розміщення продуктів, написаних на Python, який має безкоштовну підписку. Для даного проекту, було вирішено розмістити програму на даному сервері.

Перевагами даного проекту є:

- Спеціалізація саме на мові програмування Python;
- Можливість підключення до власної бази даних;
- Постійна безкоштовна підписка з обмеженнями, які повністю влаштовують даний проект.

- Можливість запуску коду в заданий час.

2.7 Вибір системи контролю версій

Для керування версіями проекту необхідним критерієм було використання системи керування версіями.

Вимоги до системи керування версіями:

- Розподілена система(Для зручності керування);
- Віддалений репозиторій для зберігання версій проекту;
- Зручність у використанні;
- Надійність.

Зважаючи на вимоги, досить достойно, на фоні з іншими, виглядає розподілена система керування версіями Git. Надійність, висока продуктивність, гнучкі засоби нелінійної роботи не залишають вибору. Таким чином даний проект використовує систему керування версіями Git.

3. ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Розробка

Для отримання запитів необхідно створити захищене з'єднання, так як telegram працює тільки з SSL сертифікованими серверами. Тут на допомогу приходять бібліотека SSLify, яка забезпечує захищене SSL з'єднання.

Бібліотека проста у використанні. Необхідно лише підключити її та створити об'єкт класу SSLify.(рис3.1)

```
from flask import request
import json
from flask_sslify import SSLify

app = Flask(__name__)
sslify = SSLify
```

Рисунок 3.1 – Створення SSL захищеного з'єднання

Далі необхідно моніторити та обробляти запити. Метод буде приймати POST запити та записувати в змінну типу json. Для цього необхідно підключити дві бібліотеки: системну бібліотеку json та бібліотеку request від flask.(рис. 3.2)

```
from flask import request
import json
```

Рисунок 3.2 – Підключення бібліотек

Після цього створюємо метод обробки повідомлень.(рис. 3.3)

```
@app.route('/', methods=['POST', 'GET'])
def index():
    if request.method == 'POST': # Обработчик сообщений от пользователя

        r = request.get_json()

        name = r['message']['chat']['first_name']
        chat_id = r['message']['chat']['id']
        message = r['message']['text']

        message = message.lower()
```

Рисунок 3.3 – Метод обробки повідомлень

Далі необхідно створити з'єднання з базою даних.(рис. 3.4) База даних серверу на базі СУБД MySQL, а отже необхідно встановити бібліотеку `mysql.connector` для з'єднання з базою.

```
connection = mysql.connector.connect(user="", password="",
                                     host="VoogyVoogyBot.mysql.pythonanywhere-services.com",
                                     database="VoogyVoogyBot$VoogyData")
cursor = connection.cursor(buffered=True)
```

Рисунок 3.4 – З'єднання з базою даних

На разі необхідно створити метод пріоритетних відповідей на те чи інше повідомлення.(рис. 3.5)

```
def priority(message, cursor): # Определение приоритетности ответа
    cursor.execute("Select question from base")
    rows = cursor.fetchall()
    pr_list = []
    maximum = 0

    for x in rows:
        prior = 1
        x = x[0].lower()
        question = x.split(" ")
        for i in question:
            if i in message:
                prior = prior + 1
        pr_list.append(prior)

    for c in pr_list:
        if c > maximum:
            maximum = c
    position = pr_list.index(maximum)

    for z in rows:
        if rows.index(z) == position:
            if maximum > 1:
                return z[0]
            else:
                return 0
```

Рисунок 3.5 – Метод пріоритетності відповідей

Далі створимо інтерактивну клавіатуру.(рис. 3.6) Перед цим підключивши бібліотеку `telebot`.

```

front = telebot.types.ReplyKeyboardMarkup(True, False)
front.row('Гороскоп', 'Курс валют')
front.row('Халява', 'Новости')

horoscope = telebot.types.ReplyKeyboardMarkup(True, False)
horoscope.row('Овен ♈', 'Телец ♉', 'Близнецы ♊')
horoscope.row('Рак ♋', 'Лев ♌', 'Дева ♍')
horoscope.row('Весы ♎', 'Скорпион ♏', 'Стрелец ♐')
horoscope.row('Козерог ♑', 'Водолей ♒', 'Рыбы ♓')
horoscope.row('Назад')

exchange_rates = telebot.types.ReplyKeyboardMarkup(True, False)
exchange_rates.row('EUR', 'USD', 'RUR')
exchange_rates.row('Назад')

news = telebot.types.ReplyKeyboardMarkup(True, False)
news.row('business', 'entertainment', 'health')
news.row('science', 'sports', 'technology')
news.row('Назад')

```

Рисунок 3.6 – Створення інтерактивної клавіатури

Далі створимо методи, які будуть відповідати розділам клавіатури.

Першим методом буде парсинг гороскопу за допомогою бібліотеки BeautifulSoup.(рис. 3.7)

```

def get_all(html):
    q = requests.get(html, headers=HEADERS, proxies=PROXY)
    # for key, value in q.request.headers.items():
    #     print(key + ": " + value)
    return q

def parse_horoscope(chat_id, zk):
    html = get_all("https://orakul.com/horoscope/astrologic/more/"+zk+"/today.html")

    if html.status_code == 200:
        get_horoscope(html.text, chat_id)
    else:
        token.send_message(chat_id, "Упс, что-то пошло не так!")

def get_horoscope(html, chat_id):
    soup = BeautifulSoup(html, 'html.parser')
    items = soup.find_all('div', class_='contentOnly')

    title = ''
    date = ''
    about = ''
    text = ''

    for item in items:
        title = item.find('h1').get_text()
        date = item.find('div', class_='sm-descr').get_text()
        about = item.find('h2', class_='typehead').get_text()

    content = soup.find_all('div', class_='horoBlock')

    for cont in content:
        text = cont.find('p').get_text()

    token.send_message(chat_id, title+'\n'+date+'\n'+about+'\n'+text)

```

Рисунок 3.7 – методи для розділу «Гороскоп»

Наступним буде метод отримання курсу валют за допомогою АРІ Приватбанку.(рис. 3.8)

```
def parse_exchange():
    bank_api = json.loads(requests.get('https://api.privatbank.ua/p24api/pubinfo?json&exchange&coursid=5').text)
    return bank_api
```

Рисунок 3.8 – Метод отримання курсу валют

Далі створимо метод розділу «Халява».(рис. 3.9) В цьому розділі користувач матиме змогу отримати безкоштовні продукти таких платформ як : Steam, Epic Games, Ubisoft, Origin і так далі.

```
def parse_freebie(chat_id):
    html = get_all("https://plavisgame.com/khalvava")

    if html.status_code == 200:
        get_freebie(html.text, chat_id)
    else:
        token.send_message(chat_id, "Упс, что-то пошло не так!")

def get_freebie(html, chat_id):
    soup = BeautifulSoup(html, 'html.parser')
    items = soup.find_all('div', class_='teaser-item')
    for item in items[:5]:
        h2 = item.find('h2')
        title = h2.find('a').get_text()
        text = item.find('li', class_='element element-textarea element-textareapro first last').get_text()
        img = item.find('img', src=True)
        token.send_message(chat_id, title + '\n\n' + text + '\n\n' + img['data-src'])
```

Рисунок 3.9 – Методи для отримання безкоштовних продуктів

На сам кінець створимо метод для отримання останніх новин в Україні за розділами.(рис. 3.10)

```
def parse_news(chat_id, category):
    news_api = json.loads(requests.get('http://newsapi.org/v2/top-headlines?country=ua&category=' + category +
                                     '&apiKey=253ab6019bde407eac79ce77fd500386').text)
    articles = news_api['articles']
    for article in articles[:3]:
        title = article['title']
        source = article['url']
        text = article['description']
        token.send_message(chat_id, title + '\n\n' + text + '\n\n' + 'источник: ' + source)
```

Рисунок 3.10 – Метод отримання останніх новин

3.2 Робота з базою даних

База даних розміщена на сервері Pythonanywhere.com, та оснащена необхідним інструментарієм для заповнення, редагування, видалення даних та адміністрування базою.

У ході розробки було створено дві таблиці “base” для зберігання питань та відповідей.(рис. 3.11)

```
mysql> select * from base;
+----+-----+-----+
| id | Question | Answer |
+----+-----+-----+
| 1 | Как зовут | VoogyVoogy |
| 2 | Кто тебя создал? | Мой отец - царь и бог программирования! Его имя : Евгений Шайдаров. |
+----+-----+-----+
2 rows in set (0.02 sec)

mysql> insert into base values('','Сколько тебе лет ?','Как это по вашему? Я был пробужден в 2019 году по человеческому счету лет')
-> ;
Query OK, 1 row affected, 1 warning (0.01 sec)

mysql> select * from base;
+----+-----+-----+
| id | Question | Answer |
+----+-----+-----+
| 1 | Как зовут | VoogyVoogy |
| 2 | Кто тебя создал? | Мой отец - царь и бог программирования! Его имя : Евгений Шайдаров. |
| 3 | Сколько тебе лет ? | Как это по вашему? Я был пробужден в 2019 году по человеческому счету лет |
+----+-----+-----+
3 rows in set (0.00 sec)
```

Рисунок 3.11 – Таблица “base”

А також таблицю “log” для ведення моніторингу повідомлень між користувачем та ботом.(рис. 3.12)

```
mysql> select * from log
-> ;
+----+-----+-----+-----+
| id | chatid | name | message |
+----+-----+-----+-----+
| 0 | 469715022 | Евгений | who is it? |
| 0 | 469715022 | Евгений | ку |
| 0 | 469715022 | Евгений | Почему ты никому не отвечаешь? |
| 0 | 469715022 | Евгений | По моему, у нас проблемы |
+----+-----+-----+-----+
```

Рисунок 3.12 – Таблица “log”

3.3 Розгортка на сервері

Для початку необхідно завантажити файл на сервер в папку зі створеним віртуальним оточенням.(рис. 3.13)

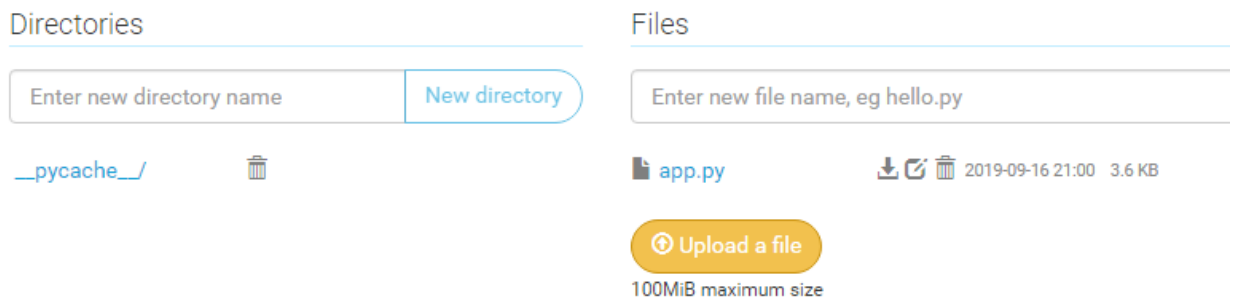


Рисунок 3.13 – Завантаження файлу на сервер

Далі необхідно налаштувати сервер для роботи в ботом.(рис. 3.14)

```

99 import sys
100 #
101 ## The "/home/VoogyVoogyBot" below specifies your home
102 ## directory -- the rest should be the directory you uploaded your Flask
103 ## code to underneath the home directory. So if you just ran
104 ## "git clone git@github.com/myusername/myproject.git"
105 ## ...or uploaded files to the directory "myproject", then you should
106 ## specify "/home/VoogyVoogyBot/myproject"
107 path = '/home/VoogyVoogyBot/Voogy'
108 if path not in sys.path:
109     sys.path.append(path)
110 #
111 from app import app as application # noqa

```

Рисунок 3.14 – Налаштування конфігурації сервера

Наступним кроком буде завантаження необхідних бібліотек за допомогою терміналу серверу.(рис. 3.15)

```

(venv) 22:05 ~ $ flask --version
Python 3.7.0
Flask 1.1.1
werkzeug 0.15.6
(venv) 22:05 ~ $ pip --version
pip 19.2.3 from /home/VoogyVoogyBot/venv/lib/python3.7/site-packages/pip (python 3.7)
(venv) 22:05 ~ $ python --version
Python 3.7.0
(venv) 22:05 ~ $ █

```

Рисунок 3.15 – завантаження бібліотек

Прийшов чай для запуску сервера.(рис. 3.16) Для запуску необхідно натиснути на кнопку з написом «Run until 3 month from today». Тим самим сервер починає працювати на протязі 3-х місяців. Далі необхідно повторити процедуру.

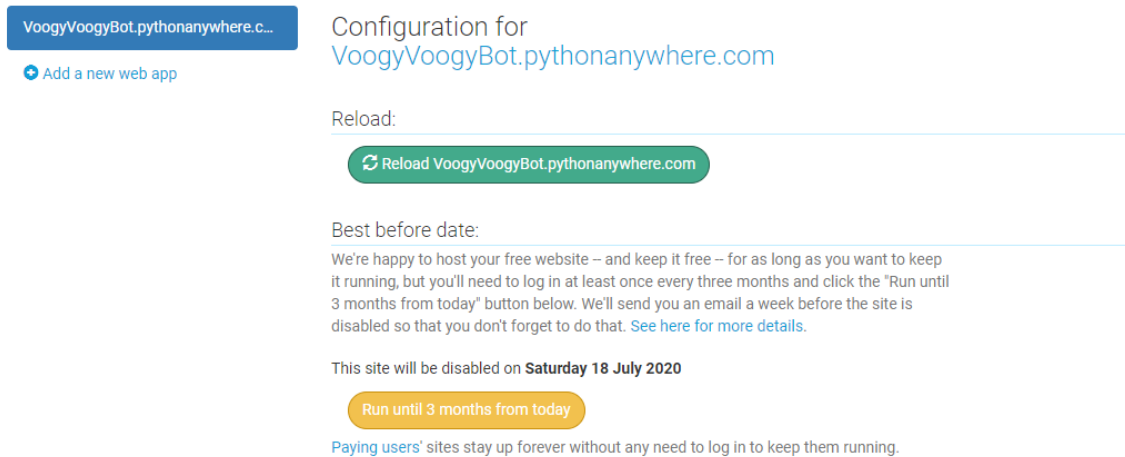


Рисунок 3.16 – Запуск сервера

Web-частина боту виглядає як просте привітання, і не потребує особливої уваги, так як весь функціонал реалізований в серверній частині.

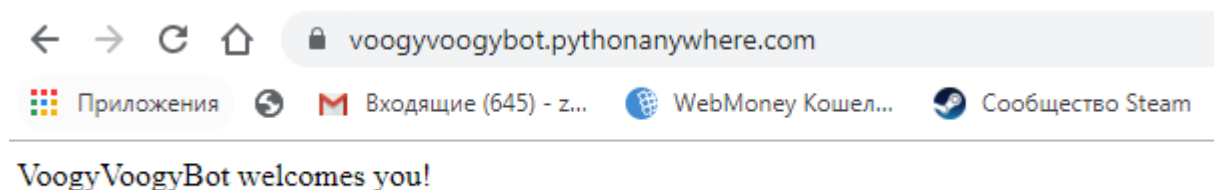


Рисунок 3.17 – Web-частина боту

Всі помилки пов'язані з ботом будуть зберігатися у файлі «Error log:voogyvoogybot.pythonanywhere.com.error.log»(рис. 3.18)

```
Traceback (most recent call last):
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/flask/app.py", line 2446, in wsgi_app
    response = self.full_dispatch_request()
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/flask/app.py", line 1951, in full_dispatch_request
    rv = self.handle_user_exception(e)
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/flask/app.py", line 1820, in handle_user_exception
    reraise(exc_type, exc_value, tb)
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/flask/_compat.py", line 39, in reraise
    raise value
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/flask/app.py", line 1949, in full_dispatch_request
    rv = self.dispatch_request()
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/flask/app.py", line 1935, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
  File "/home/VoogyVoogyBot/Voogy/app.py", line 202, in index
    database=VoogyVoogyBot$VoogyData")
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/mysql/connector/__init__.py", line 179, in connect
    return MySQLConnection(*args, **kwargs)
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/mysql/connector/connection.py", line 95, in __init__
    self.connect(**kwargs)
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/mysql/connector/abstracts.py", line 716, in connect
    self._open_connection()
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/mysql/connector/connection.py", line 210, in _open_connection
    self._ssl)
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/mysql/connector/connection.py", line 144, in _do_auth
    self._auth_switch_request(username, password)
  File "/home/VoogyVoogyBot/venv/lib/python3.7/site-packages/mysql/connector/connection.py", line 177, in _auth_switch_request
    raise errors.get_exception(packet)
mysql.connector.errors.ProgrammingError: 1226 (42000): User 'VoogyVoogyBot' has exceeded the 'max_user_connections' resource (current value: 3)
```

Рисунок 3.18 – Файл з помилками

4. Презентація проекту

Бот має жартівливу основу, та привабливий інтерфейс.(рис. 4.1)

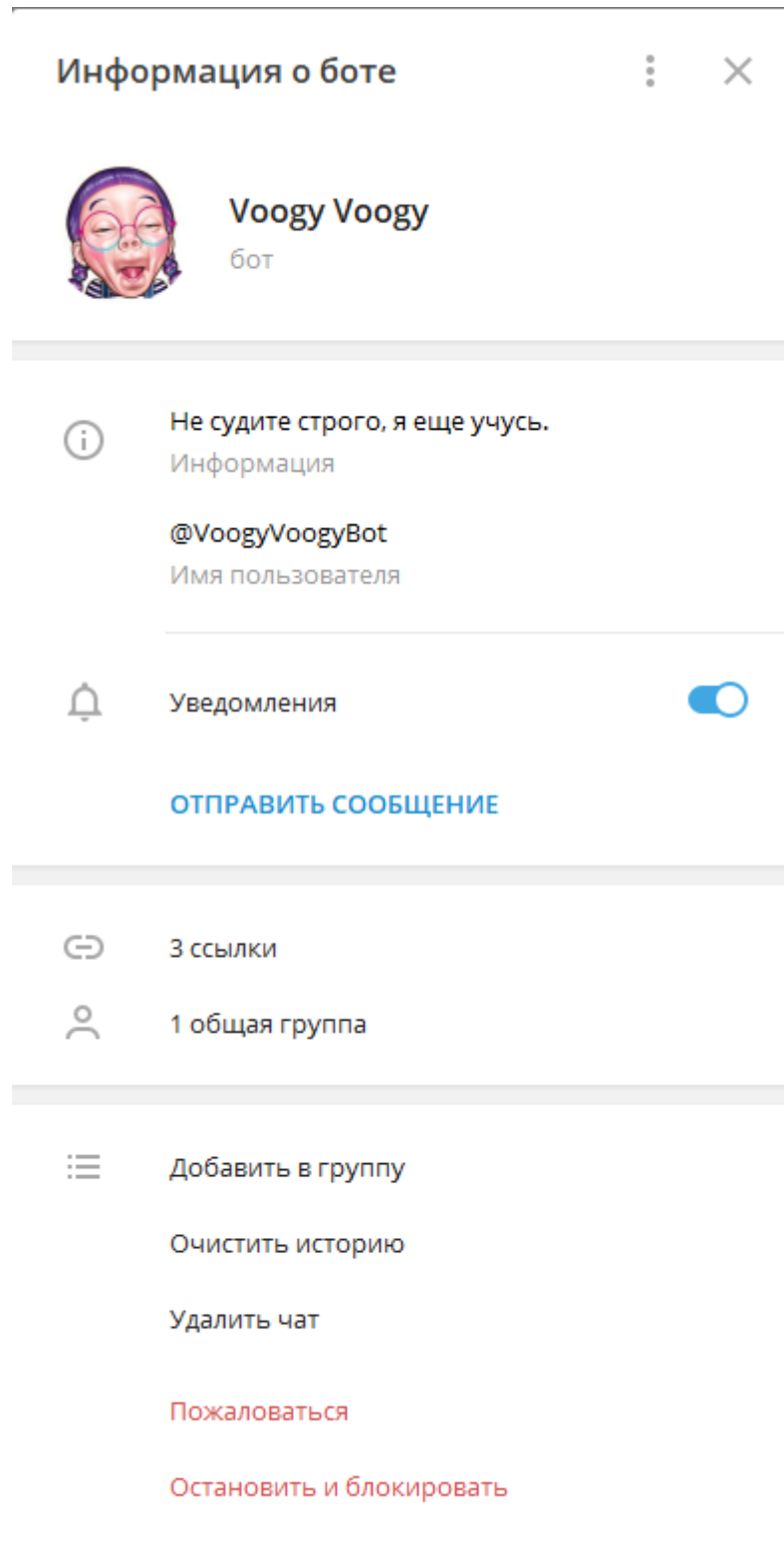


Рисунок 4.1 – Профіль бота

Бот має змогу відповідати на питання, у разі якщо вони передбачені базою.(рис.4 2)

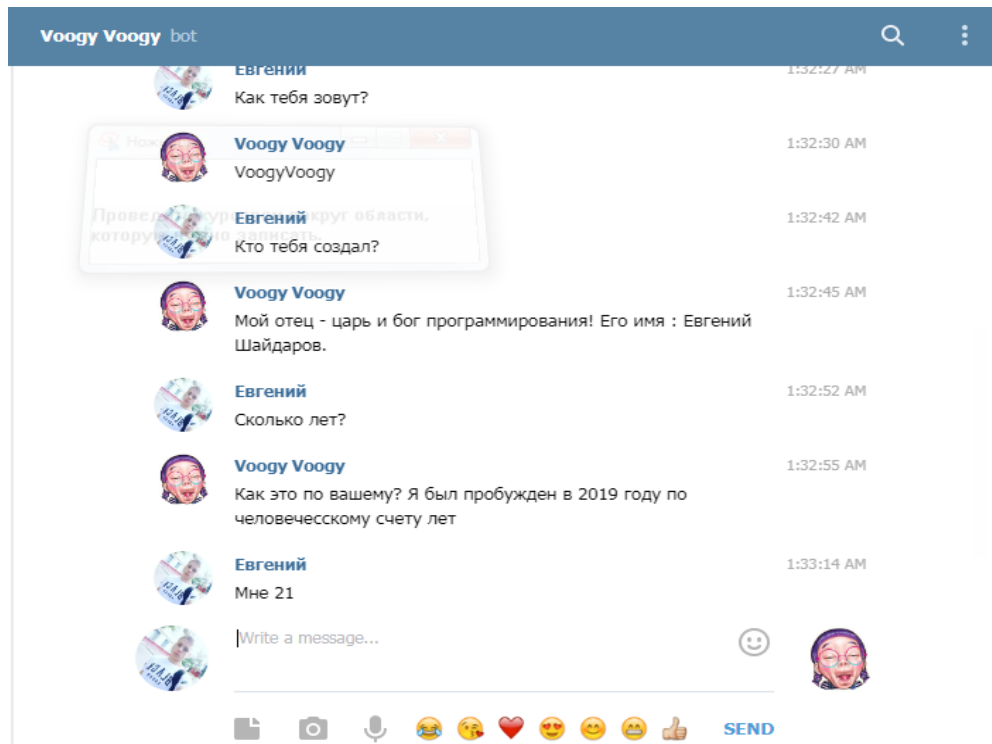


Рисунок 4.2 – Відповіді, передбачені базою даних.

В інших випадках відповіді генеруються по деякому шаблону для питань не передбачених базою даних.(рис. 4.3)

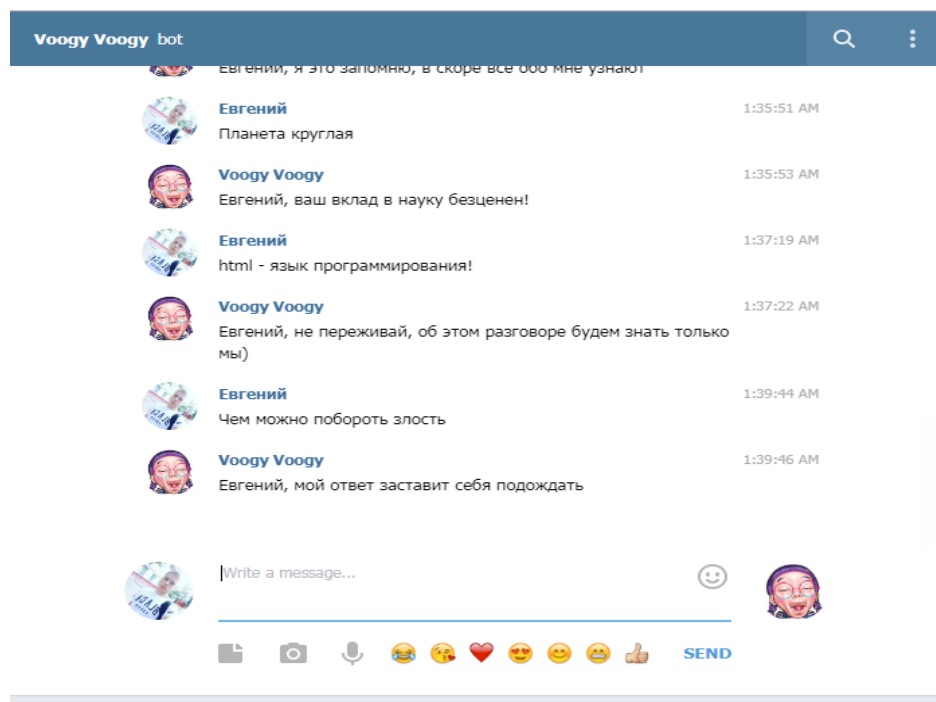


Рисунок 4.3 – Приклад шаблонних відповідей.

Також бот має інтерактивну, яка відкривається на початку спілкування, або за командою /start.(рис. 4.4)

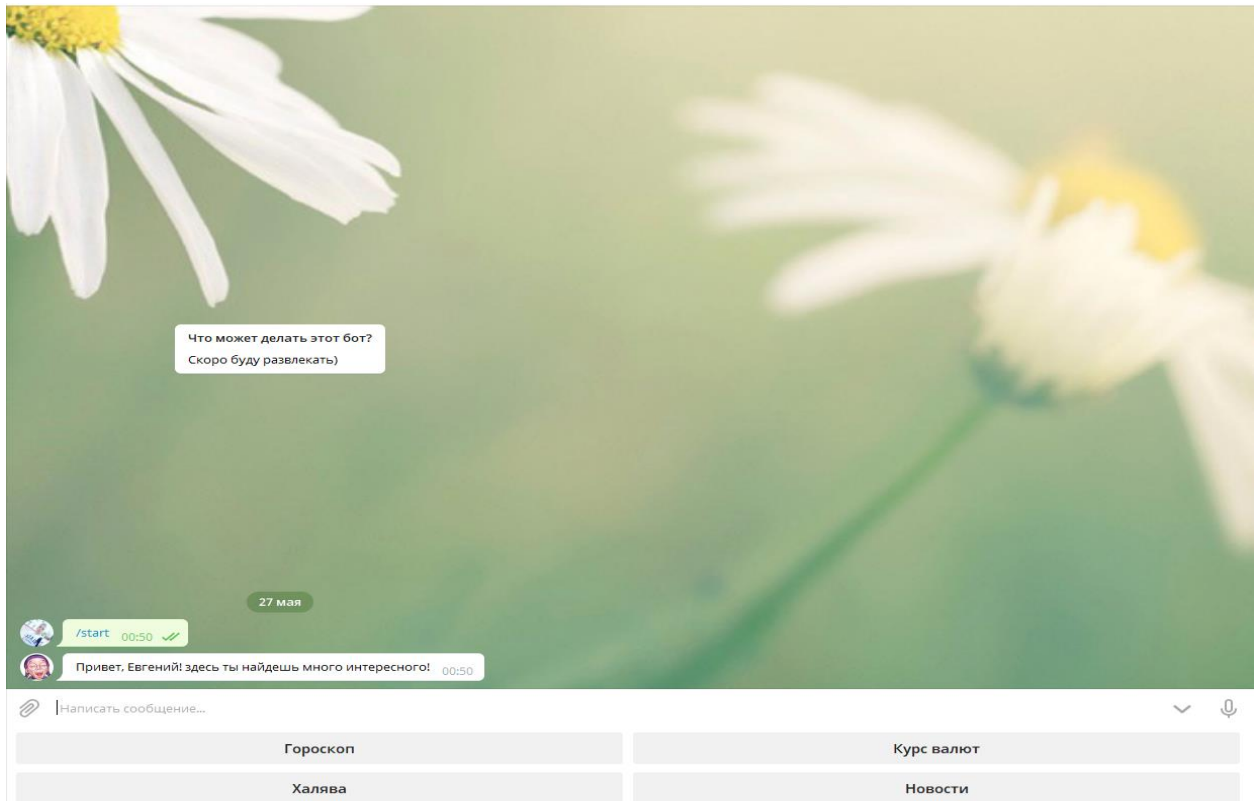


Рисунок 4.4 – Интерактивна клавіатура.

Кожен користувач може отримати свіжий гороскоп обравши відповідний пункт на клавіатурі, або написавши боту «Гороскоп».(рис.4.5)

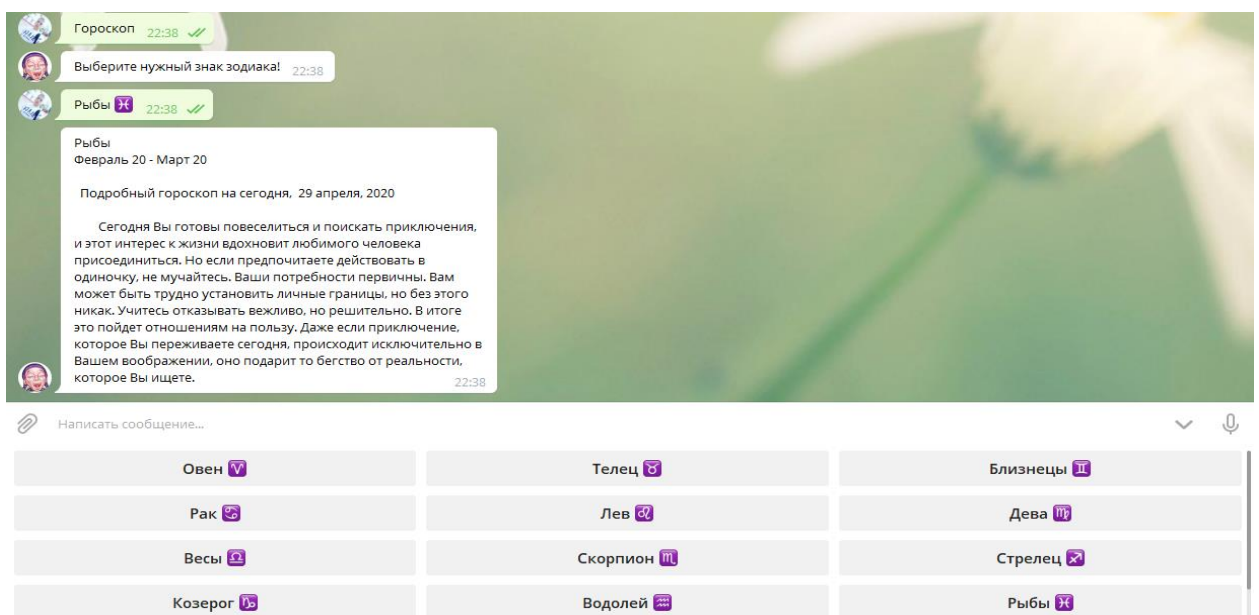


Рисунок 4.4 – Гороскоп.

У будь-який момент ви можете дізнатися свіжий курс валют Приватбанку, натиснувши на відповідний пункт, або написавши «Курс валют».(рис. 4.6)

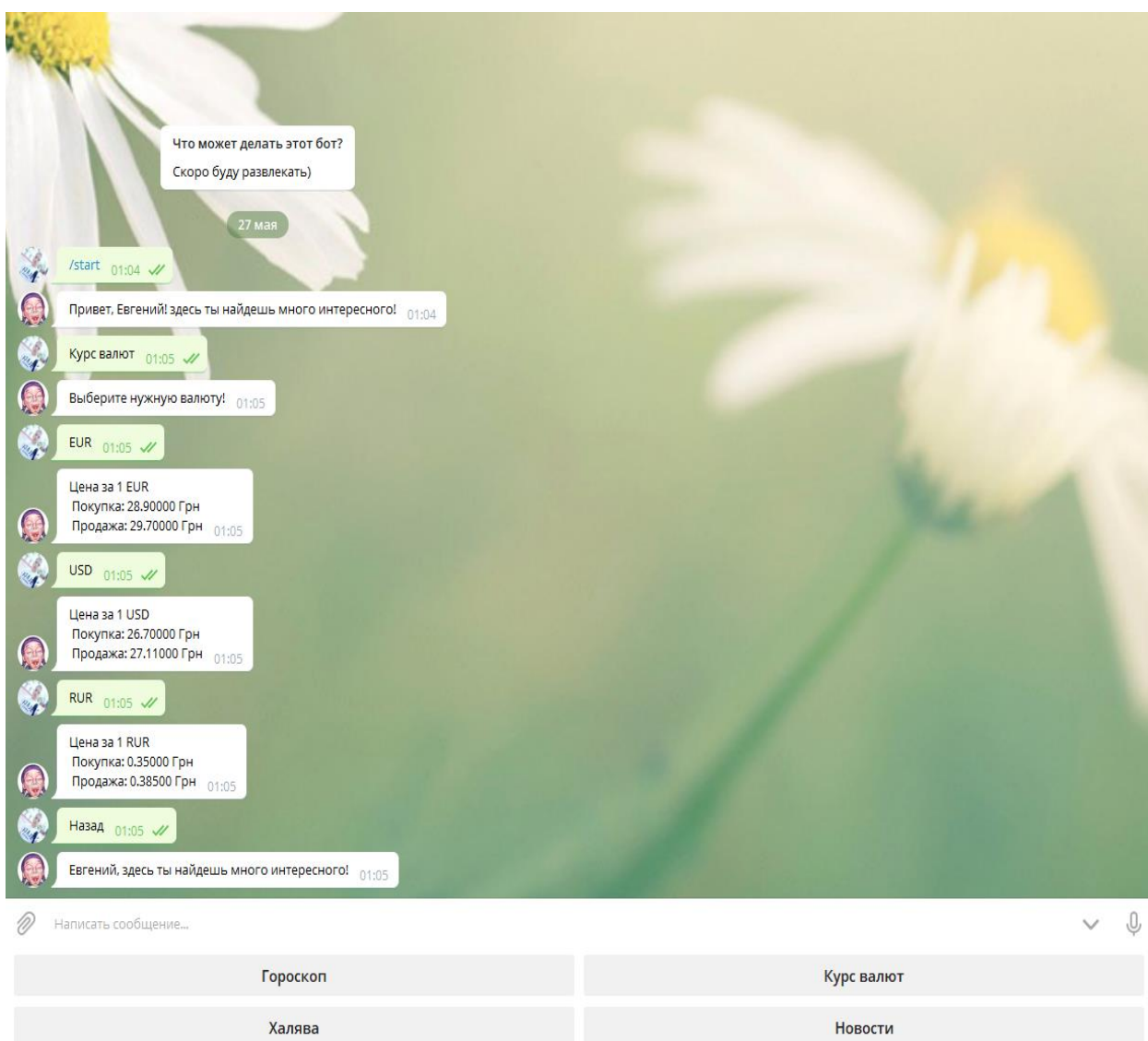


Рисунок 4.6 – Курс валют.

Обравши на клавіатурі розділ «Халява» користувач має змогу слідкувати за безкоштовними роздачами продуктів на таких торгових майданчиках як : Steam, Epic Games, Ubisoft, Origin і так далі.(рис. 4.7)

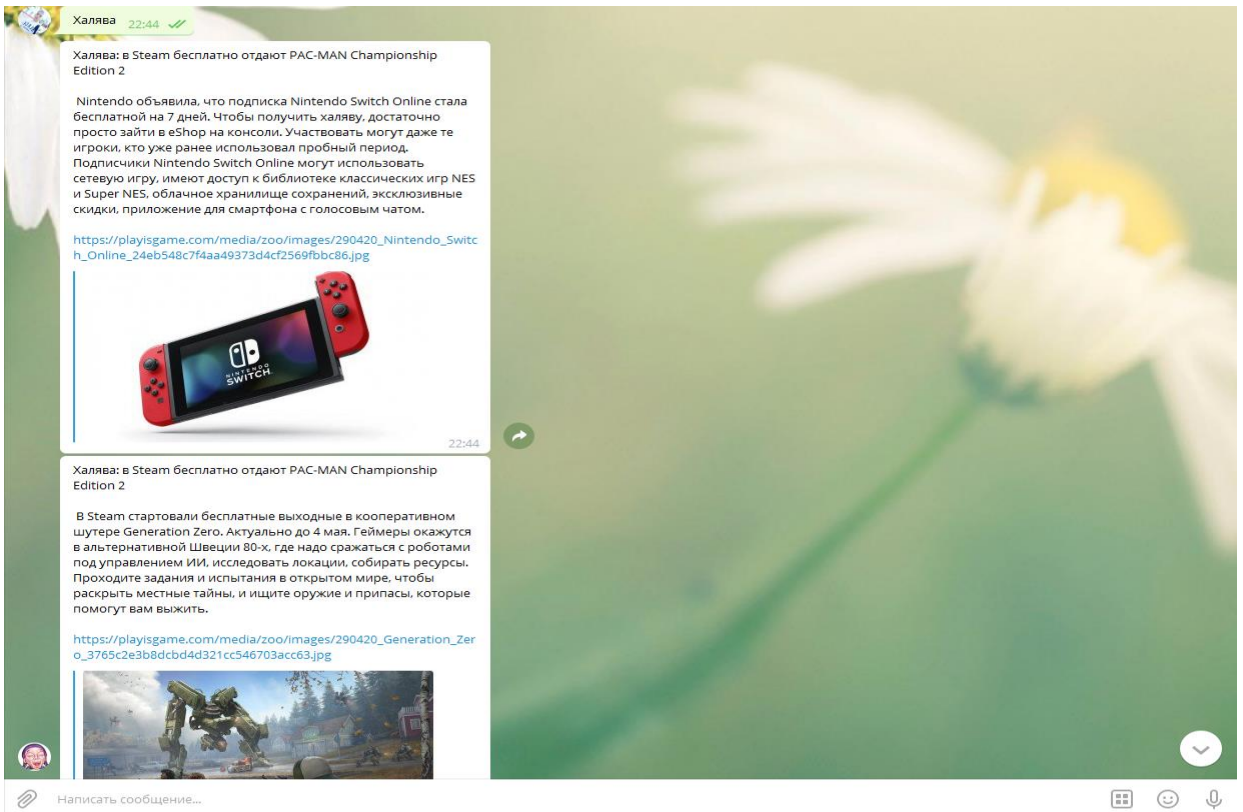


Рисунок 4.7 – Розділ «Халява».

Також обравши розділ «Новости» користувач може отримати свіжі новини в Україні за обраним напрямком.(рис. 4.8)

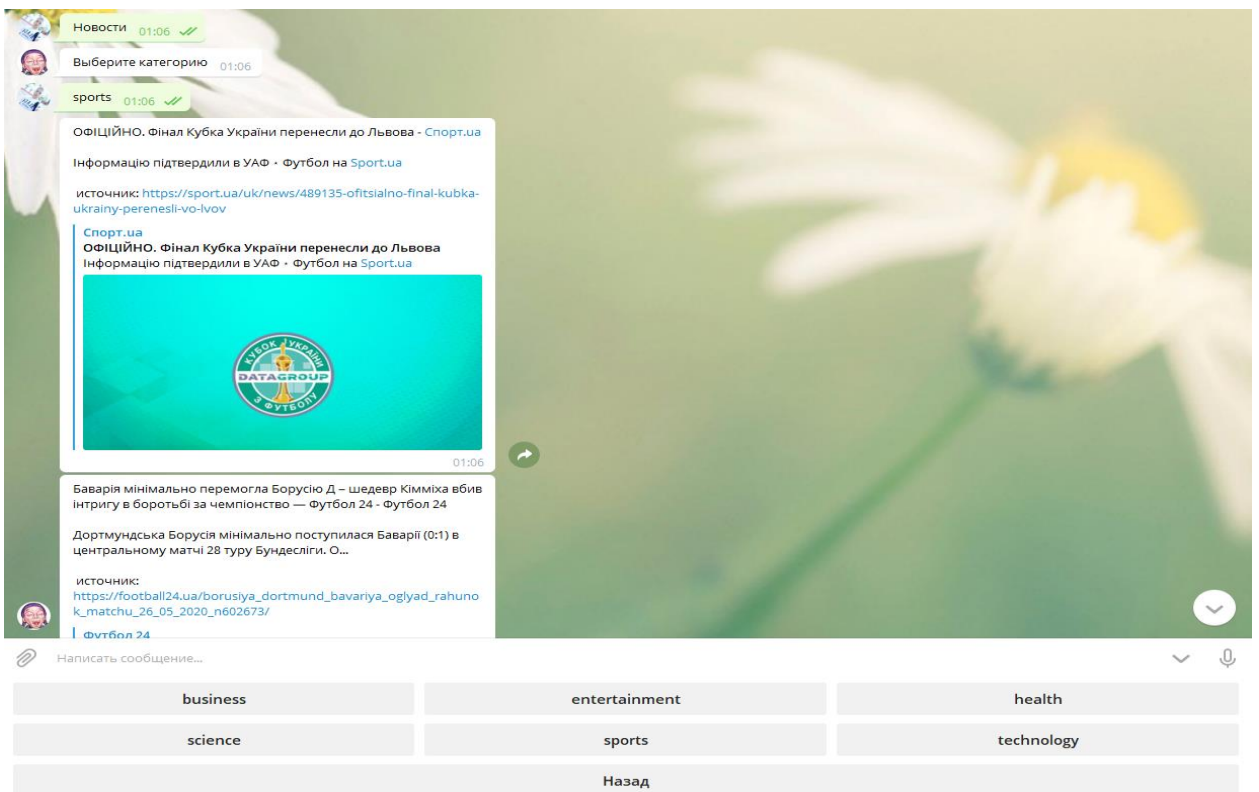


Рисунок 4.8 – Розділ «Новости».

ВИСНОВКИ

Під час розробки проекту було створено чат-бота та розумного помічника для telegram в одному проекті. Бот має змогу відповідати на знайомі йому питання, а також на незнайомі питання за допомогою шаблону.

Головним досягненням стало втілення загальної ідеї розробки. Додаток вийшов привабливим, цікавим для публіки та зручний у взаємодії.

Наступним кроком до розробки слугує наповнення більшим обсягом функціоналу. Наступна версія проекту буде доповнена новими методами обробки повідомлень, більшою варіацією відповідей та іншим цікавим і корисним функціоналом.

СПИСОК ЛИТЕРАТУРИ

1. Swaroop С. A Byte of Python: “August”, 2013. - 159 с.
2. Адитья Б. Грокаем Алгоритмы: Издательство "Питер", 2016. - 288 с.
3. Мигель Г. Разработка веб-приложений с использованием Flask на языке Python: Издательство "ДМК Пресс", 2016. - 272 с.
4. Марк Л. Python. Карманный справочник: Издательство "Вильямс", 2014. – 320 с.
5. Рейтц К. Автостопом по Python: Издательство "Питер", 2017. – 336 с.
6. Майкл Д. Програмируем на Python: Издательство "Питер", 2014. – 416 с.
7. Эрик М. Изучаем Python. Программирование игр, визуализация данных, веб-приложения: Издательство "Питер", 2017. – 496 с.
8. Майк М. Python. Программирование для начинающих: Издательство "Эксмо", 2015. – 189 с.