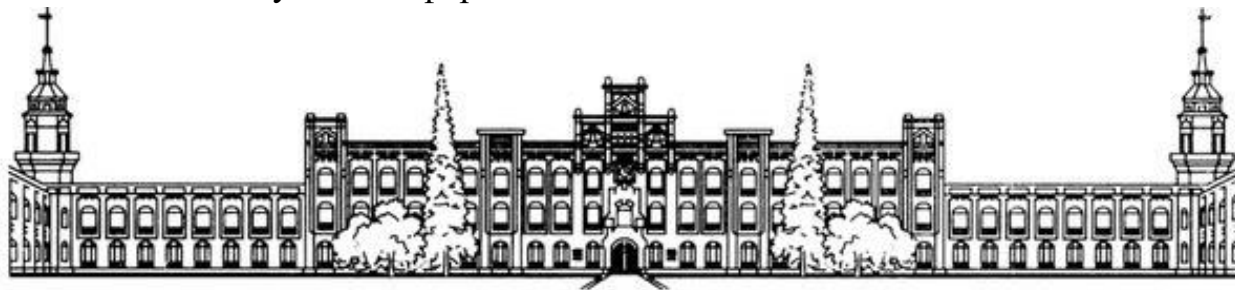


Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет Інформатики та Обчислювальної Техніки



Кафедра інформаційних систем та технологій

Лабораторна робота №4
з дисципліни «Вступ до технології Data Science»

на тему

**«РЕАЛІЗАЦІЯ ПРОЦЕСІВ ІНТЕЛЕКТУАЛЬНОГО
АНАЛІЗУ ДАНИХ: МІНІ ПРОЕКТИ В ГАЛУЗЯХ
OLAP, Data Mining, Text Mining, Voice Recognition»**

Виконала:
студентка групи ІС-12
Павлова Софія

Перевірив:
Баран Д. Р.

1. Постановка задачі

Мета роботи:

Виявити дослідити та узагальнити особливості інтелектуального аналізу даних та технологій OLAP, Data Mining, Text Mining, Voice Recognition.

Завдання II рівня:

	Задача на вільний вибір із власного досвіду професійної діяльності.
--	---

Розробити програмний скрипт, що реалізує обробку природньої мови – створити bot-асистента з вибору аніме для перегляду з урахуванням побажань користувача.

2. Виконання

2.1. Парсинг аніме сайту

Індустрія розваг – динамічна і її тренди міняються щогодини. Тому для того, щоб наш бот-асистент володів актуальною інформацією про тенденції в світі аніме, бот має оперувати даними про аніме з сучасної платформи для перегляду аніме.

Для цього напишемо парсер сайту <https://anitube.in.ua/anime/>, який буде збирати дані про 11 перших аніме-тайтлів з головної сторінки.

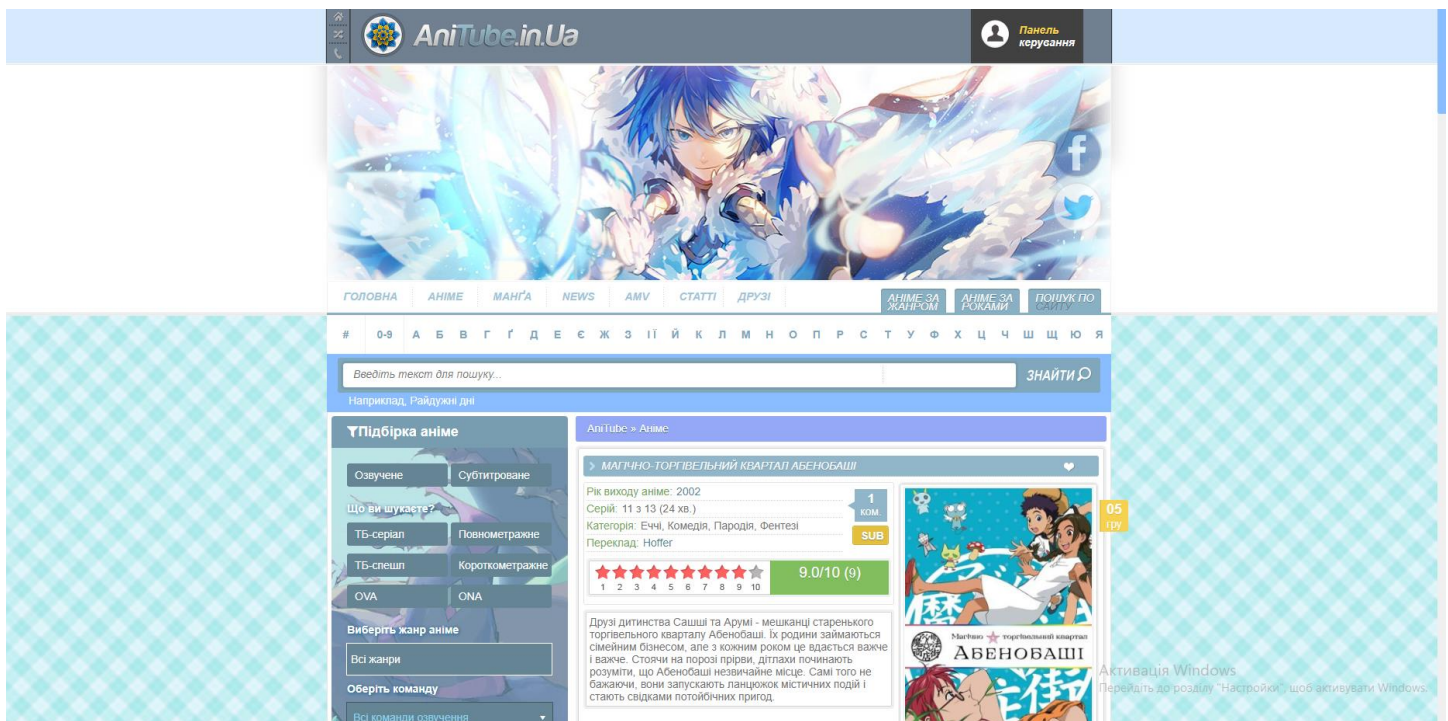


Рисунок 1 – Сайт для парсингу

Дані будуть представлені в такому порядку: **«Назва»**, **«Дубляж/Субтитри»**, **«Рік випуску»**, **«Кількість серій»**, **«Жанр»**, **«Студія»**, **«Рейтинг»**, **«Опис»**.

Так, як ознайомлення з технологією парсингу сайтів не є метою даної лабораторної роботи – залишимо за користувачем право друкувати структуру HTML документу або ні.

Лістинг коду:

```
# Аналіз структури html документу
print('Друкувати структуру HTML документу?')
print('0 - так')
print('1 - ні')
data_mode = int(input('mode:'))
soup = BeautifulSoup(response.text, 'lxml')
if data_mode == 0:
    print(soup)
```

Результат:

Якщо не друкувати структуру HTML документу, вивід програми буде лаконічнішим і легшим для аналізу.

```
Обрано інформаційне джерело: https://anitube.in.ua/anime/
Друкувати структуру HTML документу?
0 - так
1 - ні
mode:1

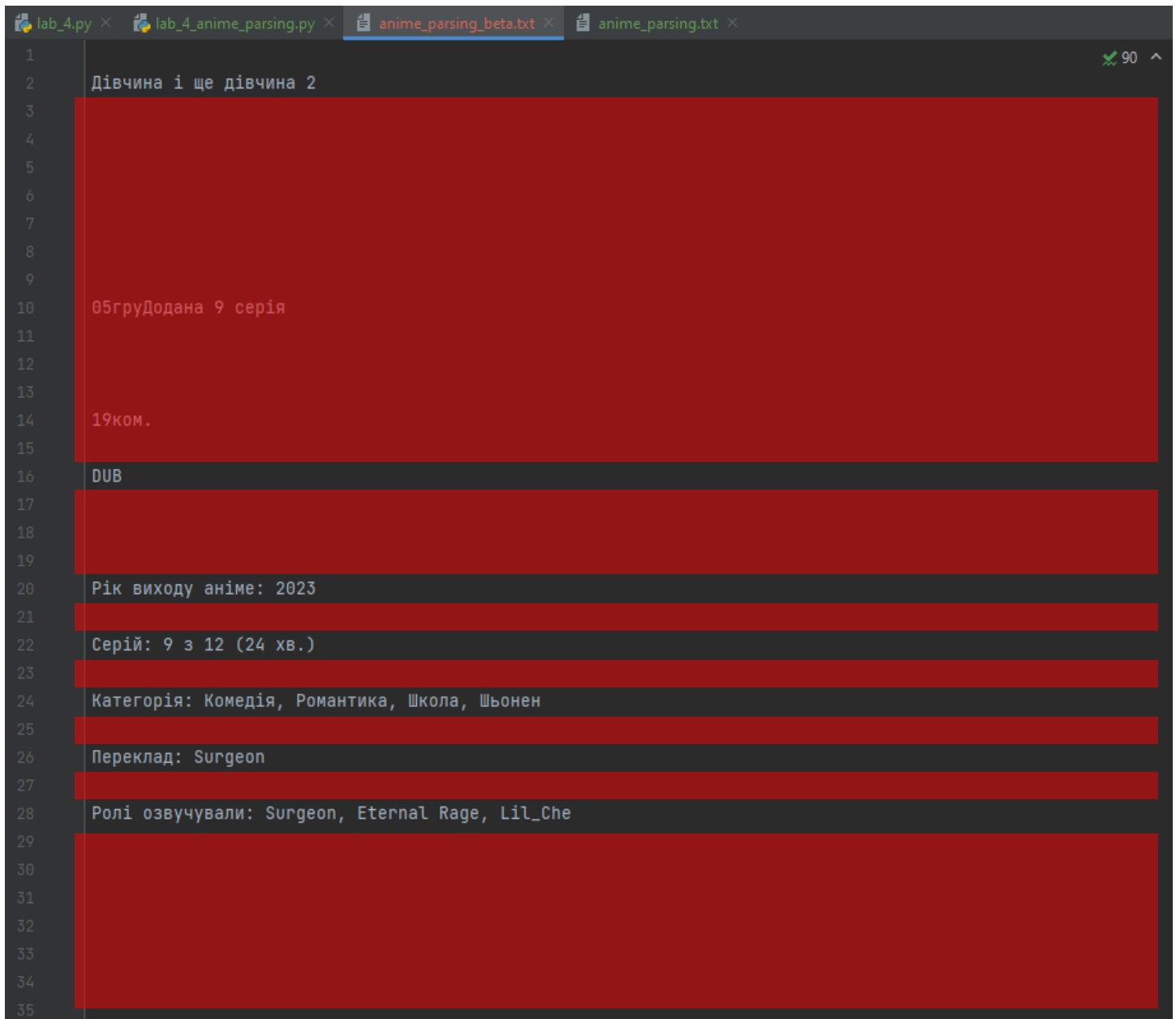
-----
Список аніме:
-----

Дівчина і ще дівчина 2
DUB
Рік виходу аніме: 2023
Серій: 9 з 12 (24 хв.)
Категорія: Комедія, Романтика, Школа, Шьонен
Переклад: Surgeon
Ролі озвучували: Surgeon, Eternal Rage, Lil_Che
8.2/10 (35)
Продовження однойменного тайтлу.

Сходження героя щита (3 сезон)
16+
D+S
Рік виходу аніме: 2023
Серій: 9 з 12 (24 хв.)
Категорія: Бойовик, Драма, Пригоди, Фентезі
Переклад: Surgeon, Joer, Shiman, Florentia Mysteria
Ролі озвучували: Crupt, LunarShadow, Eternal Rage, BaRMaN, Dixy, Trina_D, Idea, Shiman, Venko, Lianeli, Chis
8.5/10 (134)
Третій сезон однойменного тайтлу.
```

Рисунок 2 – Можливість вибору друкувати структуру HTML сторінки чи ні

Сирі результати парсингу мають багато зайвих пустих рядків та інформації, якою можна знехтувати при аналізі.



```
1
2 Дівчина і ще дівчина 2
3
4
5
6
7
8
9
10 05грудодана 9 серія
11
12
13
14 19ком.
15
16 DUB
17
18
19
20 Рік виходу аніме: 2023
21
22 Серій: 9 з 12 (24 хв.)
23
24 Категорія: Комедія, Романтика, Школа, Шьонен
25
26 Переклад: Surgeon
27
28 Полі озвучували: Surgeon, Eternal Rage, Lil_Che
29
30
31
32
33
34
35
```

Рисунок 3 – Незручне представлення сирого результату парсингу

З метою подання результатів парсингу в зручному для аналізу вигляді, напишемо функцію для обробки даних.

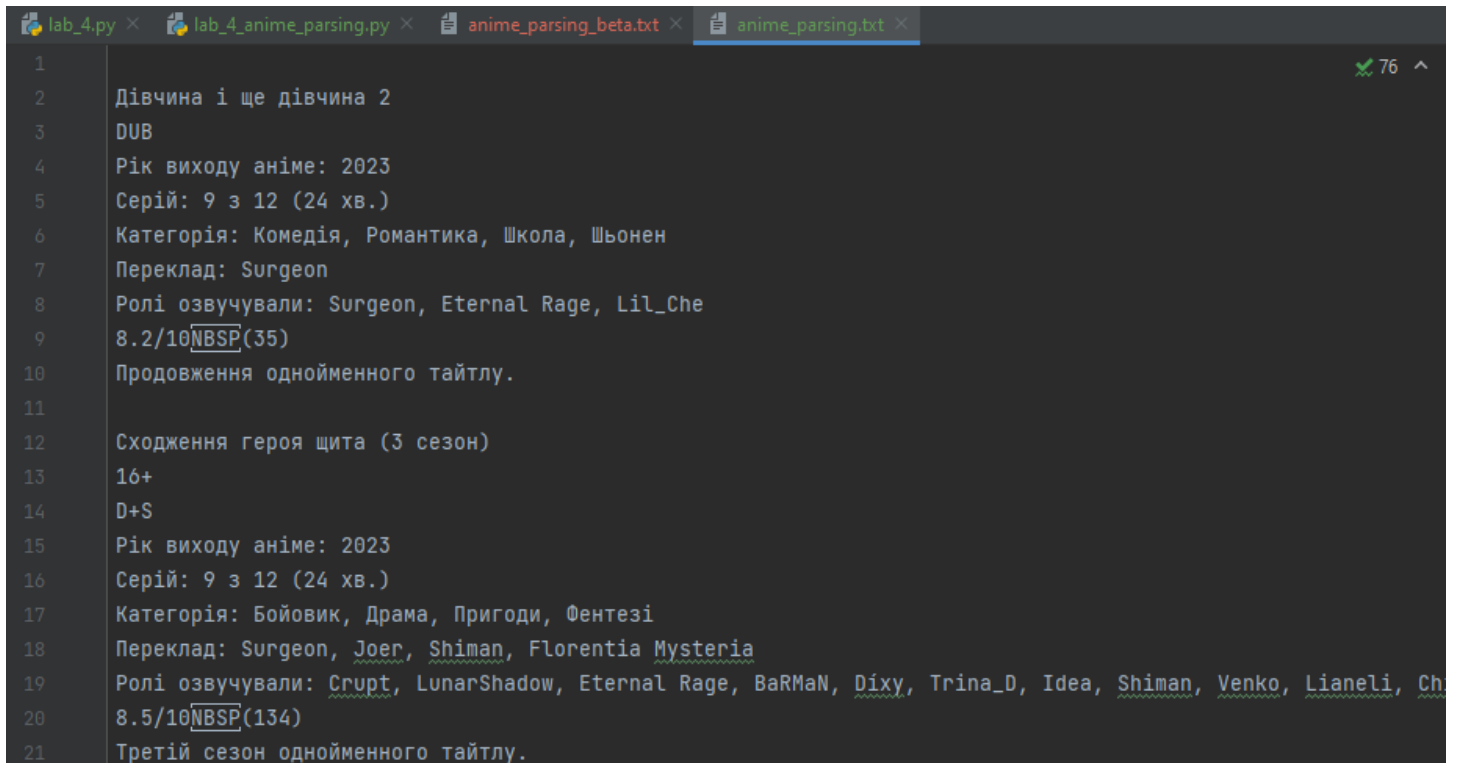
Лістинг коду:

```
def text_mining(filename):
    # Шаблони для вишування
    pattern_comments = r'\n*\d+ком\.'
    pattern_anime_year = r'\n\n\n\n\s*(?=Рік виходу аніме: \d+)'
    pattern_added = r'.*Додана.*\n\n'
    pattern_added_2 = r'.*Додані.*\n\n'
    pattern_added_3 = r'.*Перегляд.*\n\n'
    pattern_rating = r'\n*(.*/10.*)\n\n\n'
    pattern_3_space = r'\n\n\n'
    pattern_5_space = r'\n\n\n\n\n'
    pattern_2_space = r'\n\n'

    # Застосування регулярних виразів для вишування
    with open(filename, encoding="utf-8") as file:
        text = file.read()
    text = re.sub(pattern_comments, '', text)
    text = re.sub(pattern_anime_year, '\n\n', text)
    text = re.sub(pattern_added, '', text)
    text = re.sub(pattern_added_2, '', text)
    text = re.sub(pattern_added_3, '', text)
    text = re.sub(pattern_rating, r'\n\n1\n', text)
    text = re.sub(pattern_3_space, '\n\n', text)
    text = re.sub(pattern_5_space, '\n', text)
    text = re.sub(pattern_2_space, '\n', text)
    return text
```

Результат:

У результаті скорочена кількість пустих рядків і дані легко читаються.



```
1
2 Дівчина і ще дівчина 2
3 DUB
4 Рік виходу аніме: 2023
5 Серій: 9 з 12 (24 хв.)
6 Категорія: Комедія, Романтика, Школа, Шьонен
7 Переклад: Surgeon
8 Полі озвучували: Surgeon, Eternal Rage, Lil_Che
9 8.2/10NBSP(35)
10 Продовження однойменного тайтлу.
11
12 Сходження героя щита (3 сезон)
13 16+
14 D+S
15 Рік виходу аніме: 2023
16 Серій: 9 з 12 (24 хв.)
17 Категорія: Бойовик, Драма, Пригоди, Фентезі
18 Переклад: Surgeon, Joer, Shiman, Florentia Mysteria
19 Полі озвучували: Crupt, LunarShadow, Eternal Rage, BaRMaN, Dixy, Trina_D, Idea, Shiman, Venko, Lianeli, Ch
20 8.5/10NBSP(134)
21 Третій сезон однойменного тайтлу.
```

Рисунок 4 – Результат парсингу, збережений у .txt файл

Наведемо увесь код парсера.

Лістинг коду:

```
import requests
from bs4 import BeautifulSoup
import re
import os

# URL веб-сторінки, яку будемо парсити
url = 'https://anitube.in.ua/anime/'
print('Обрано інформаційне джерело:', url)

def parser_url (url):
    response = requests.get(url)

    # Аналіз структури html документу
    [...]

    # Вилучення із html документу списку аніме
    quotes = soup.find_all('div', class_='story_c')

    # Запис сирого списку аніме у додатковий файл для обробки
    with open('anime_parsing_beta.txt', 'w', encoding='utf-8') as file:
        for quote in quotes:
            file.write(quote.text)

    # Запис обробленого списку аніме в файл "anime_parsing.txt"
    with open('anime_parsing.txt', 'w', encoding='utf-8') as file:
        text = text_mining('anime_parsing_beta.txt')
        file.write(text)
        print('\n-----')
        print('Список аніме:')
        print('-----')
        print(text)

    # Видалення додаткового файлу для обробки
    if os.path.exists('anime_parsing_beta.txt'):
        os.remove('anime_parsing_beta.txt')

    return

def text_mining(filename):
    [...]

parser_url(url)
```

2.2. Bot-асистент з обробкою природної мови

У якості інструмента для розпізнавання природної мови використаємо бібліотеку [SpeechRecognition](#). Для зчитування аудіо з мікрофона нашого пристрою, використаємо бібліотеку [PyAudio](#).

Побудова bot-асистента складатиметься з наступних етапів:

- створення екземпляра-розпізнавача;
- захоплення голосового введення;
- перетворення голосового введення в текст;
- обробка голосових команд;
- основна функції викликів.

Створення екземпляра-розпізнавача

Для початку створимо екземпляр класу Recognizer:

Лістинг коду:

```
import speech_recognition as sr

# Створення екземпляру класу Recognizer
recognizer = sr.Recognizer()
```

Захоплення голосового введення

Створимо функцію для захоплення голосового введення від користувача за допомогою мікрофона.

Лістинг коду:

```
# Запис голосового повідомлення
def capture_voice_input():
    with sr.Microphone() as source:
        print('Говоріть...')
        audio = recognizer.listen(source)
    return audio
```


Перетворення голосового введення в текст

Створимо функцію для перетворення захопленого голосового введення в текст.

Лістинг коду:

```
# Перетворення голосового повідомлення на текст
def convert_voice_to_text(audio):
    try:
        text = recognizer.recognize_google(audio, language='uk-UA')
        print('\n')
        print('-' * 50)
        print('| \tYou: ' + text)
        print('-' * 50)
    except sr.UnknownValueError:
        text = ''
        print('Anime Helper: Вибачте, я Вас не розумію.')
    except sr.RequestError as e:
        text = ''
        print('Anime Helper: Error; {0}'.format(e))
    return text
```

Обробка голосових команд

Передбачимо для нашого бота-асистента наступний функціонал:

- *вітання;*
- вибір аніме зі списку результатів парсингу;
- *прощання.*

Розширимо функціонал пошуку серед результатів парсингу:

- *вибір аніме з українською озвучкою;*
- *вибір аніме з декількома сезонами;*
- *вибір аніме цього року випуску;*
- *вибір аніме за жанром;*

Аби наш бот-асистент був більш універсальним, для кожного вищеописаного випадку функціоналу передбачимо декілька «**слів-тригерів**». Розглянемо кожен з таких випадків функціоналу окремо.

Почнемо з **вітання**.

Лістинг коду:

```
# Заданий перелік слів
hello_words = ['привіт', 'вітаю', 'добрий день', 'доброго дня', 'доброго ранку', 'добрий
ранок', 'добрий вечір', 'доброго вечора', 'підкажи', 'підкажіть']

# Обробка голосових команд
def process_voice_command(text):
    # Якщо 'вітаю'
    if any(word in text.lower() for word in hello_words):
        print('Anime Helper: Привіт! Чим я можу допомогти?')
```

Результат:

У результаті бот просто вітається і чекає на наступні голосові команди.

Говорить...

You: вітаю

Anime Helper: Привіт! Чим я можу допомогти?

Говорить...

Рисунок 5 – Результат обробки запиту вітання

Розглянемо **вибір аніме з українською озвучкою**.

Лістинг коду:

```
# Заданий перелік слів
[...]
dub_words = ['дубляж', 'озвучка', 'дубляжем', 'озвучкою']

# Обробка голосових команд
def process_voice_command(text):
    # Якщо 'вітаю'
    [...]
    # Якщо 'дубляж'
    elif any(word in text.lower() for word in dub_words):
        print('Anime Helper: Якщо Ви шукаєте аніме з дубляжем, ось деякі мої
рекомендації')
        print('')
        print('АНИМЕ З ДУБЛЯЖЕМ:')
        print('')
        with open('anime_parsing.txt', 'r', encoding='utf-8') as file:
            # Розділяємо текст на абзаци за двома новими рядками
            paragraphs = file.read().split('\n\n')
            for paragraph in paragraphs:
                if 'DUB' in paragraph or 'D+' in paragraph:
                    print(paragraph)
                    # Додаємо розділювач для читабельності
                    print('-' * 50)
```

Результат:

У результаті бот виводить лише ті аніме-тайтли з результатів парсингу, які мають позначку «**DUB**» або «**D+S**».

```
Говорить...

-----
|   You: Порадь будь ласка аніме з українською озвучкою
|-----
Anime Helper: Якщо Ви шукаєте аніме з дубляжем, ось деякі мої рекомендації

АНІМЕ З ДУБЛЯЖЕМ:

Дівчина і ще дівчина 2
DUB
Рік виходу аніме: 2023
Серій: 9 з 12 (24 хв.)
Категорія: Комедія, Романтика, Школа, Шьонен
Переклад: Surgeon
Ролі озвучували: Surgeon, Eternal Rage, Lil_Che
8.2/10 (35)
Продовження однойменного тайтлу.
-----
Сходження героя щита (3 сезон)
16+
D+S
Рік виходу аніме: 2023
Серій: 9 з 12 (24 хв.)
Категорія: Бойовик, Драма, Пригоди, Фентезі
Переклад: Surgeon, Joer, Shiman, Florentia Mysteria
Ролі озвучували: Crupt, LunarShadow, Eternal Rage, BaRMaN, Dixy, Trina_D, Idea, Shiman, Venko, Lianeli, Chi
8.5/10 (134)
Третій сезон однойменного тайтлу.
```

Рисунок 6 – Результат обробки запиту на українську озвучку

Розглянемо **вибір багатосезонного аніме**.

Лістинг коду:

```
# Заданий перелік слів
[...]
seasons_words = ['довге', 'сезонів']

# Обробка голосових команд
def process_voice_command(text):
    # Якщо 'вітаю'
    [...]
    # Якщо 'дубляж'
    [...]
    # Якщо 'довге'
    elif any(word in text.lower() for word in seasons_words):
        print('Anime Helper: Якщо Ви хочете подивитись аніме, що має декілька сезонів, ось приклад аніме, що може Вам сподобатись')
        print('')
        print('АНІМЕ НА ДЕКІЛЬКА СЕЗОНІВ:')
        print('')
        with open('anime_parsing.txt', 'r', encoding='utf-8') as file:
            # Розділяємо текст на абзаци за двома новими рядками
            paragraphs = file.read().split('\n\n')
            for paragraph in paragraphs:
                if 'сезон' in paragraph:
                    print(paragraph)
                    # Додаємо розділювач для читабельності
                    print('-' * 50)
```

Результат:

У результаті бот виводить лише ті аніме-тайтли з результатів парсингу, які мають **2 або більше сезони**.

Говоріть...

You: Я хочу подивитись якесь довге аніме

Anime Helper: Якщо Ви хочете подивитись аніме, що має декілька сезонів, ось приклад аніме, що може Вам сподобатись

АНІМЕ НА ДЕКІЛЬКА СЕЗОНІВ:

Сходження героя щита (3 сезон)

16+

D+S

Рік виходу аніме: 2023

Серій: 9 з 12 (24 хв.)

Категорія: Бойовик, Драма, Пригоди, Фентезі
Переклад: Surgeon, Joer, Shiman, Florentia Mysteria
Ролі озвучували: Crupt, LunarShadow, Eternal Rage, BaRMaN, Dixy, Trina_D, Idea, Shiman, Venko, Lianeli, Ch
8.5/10 (134)
Третій сезон однойменного тайтлу.

Магічна Битва (2 сезон)
16+
DUB
Рік виходу аніме: 2023
Серій: 19 з 23 (23 хв.)
Категорія: Бойовик, Фентезі, Школа, Шьонен
Переклад: Серафікус, Spiral Team, SuitOn, TATAKAE, Michae, Mefune, KingGalant, Delta, Momo, Sherond
Ролі озвучували: Bodya500icq, Rainy984, DedrDs, Nutix, Suni, Yunko, Ihor Korzhenko, Scarlet, Mefune, Dumb
9.6/10 (1204)
Другий сезон "Магічної битви" охоплює події арок "Іскра Божа", "Згаслий племін" й "Шібуйський інцидент".

Рисунок 7 – Результат обробки запиту на багатосезонне аніме

Розглянемо **вибір аніме цього року випуску**.

Лістинг коду:

```
# Заданий перелік слів
[...]
new_words = ['нове', 'новинки', 'свіже', 'свіжі', 'цього року']

# Обробка голосових команд
def process_voice_command(text):
    # Якщо 'вітаю'
    [...]
    # Якщо 'дубляж'
    [...]
    # Якщо 'довге'
    [...]
    # Якщо 'новинка'
    elif any(word in text.lower() for word in new_words):
        print('Anime Helper: Якщо Ви хочете подивитись аніме-новинки, ось деякі популярні')
        print('')
        print('АНІМЕ 2023 РОКУ:')
        print('')
        with open('anime_parsing.txt', 'r', encoding='utf-8') as file:
            # Розділяємо текст на абзаци за двома новими рядками
            paragraphs = file.read().split('\n\n')
            for paragraph in paragraphs:
                if '2023' in paragraph:
                    print(paragraph)
                    # Додаємо розділювач для читабельності
                    print('-' * 50)
```

Результат:

У результаті бот виводить лише ті аніме-тайтли з результатів парсингу, які випущені у 2023 році.

```
Говорить...

-----
|   You: що можна подивитись з аніме цього року
|-----
Anime Helper: Якщо Ви хочете подивитись аніме-новинки, ось деякі популярні

АНІМЕ 2023 РОКУ:

Дівчина і ще дівчина 2
DUB
Рік виходу аніме 2023
Серій: 9 з 12 (24 хв.)
Категорія: Комедія, Романтика, Школа, Шьонен
Переклад: Surgeon
Ролі озвучували: Surgeon, Eternal Rage, Lil_Che
8.2/10 (35)
Продовження однойменного тайтлу.
-----

Сходження героя щита (3 сезон)
16+
D+S
Рік виходу аніме 2023
Серій: 9 з 12 (24 хв.)
Категорія: Бойовик, Драма, Пригоди, Фентезі
Переклад: Surgeon, Joer, Shiman, Florentia Mysteria
Ролі озвучували: Crupt, LunarShadow, Eternal Rage, BaRMaN, Dixy, Trina_D, Idea, Shiman, Venko, Lianeli, Chi
8.5/10 (134)
Третій сезон однойменного тайтлу.
```

Рисунок 8 – Результат обробки запиту на аніме-новинки

Розглянемо **вибір аніме за жанром**.

Лістинг коду:

```
# Заданий перелік слів
[...]
genre_words = ['жанру', 'жанрі', 'жанр']

# Обробка голосових команд
def process_voice_command(text):
    # Якщо 'вітаю'
    [...]
    # Якщо 'дубляж'
    [...]
    # Якщо 'довге'
    [...]
    # Якщо 'новинка'
    [...]
    # Якщо 'жанр'
    elif any(word in text.lower() for word in genre_words):
        # Розділяємо живу мову на слова
        words = text.split()
        # Визначаємо жанр
        for i, word in enumerate(words):
            if any(word in word.lower() for word in genre_words):
                if i < len(words) - 1:
                    genre_value = words[i + 1].lower()
                print('Anime Helper: Якщо Ви хочете подивитись аніме в жанрі', genre_value, ',
ось деякі рекомендації')
                print('')
                print('АНІМЕ В ЖАНРІ', genre_value.upper(), ':')
                print('')
                with open('anime_parsing.txt', 'r', encoding='utf-8') as file:
                    # Розділяємо текст на абзаци за двома новими рядками
                    paragraphs = file.read().split('\n\n')
                    for paragraph in paragraphs:
                        if genre_value in paragraph.lower():
                            print(paragraph)
                            # Додаємо розділювач для читабельності
                            print('-' * 50)
```

Результат:

У результаті бот спочатку розділяє голосовий ввід на слова і запам'ятовує наступне слово після слова-тригера «жанр». По якому потім шукає і виводить лише ті аніме-тайтли з результатів парсингу, у яких вказаний відповідний жанр.

Говоріть...

You: порадь мені щось з жанру романтика

Anime Helper: Якщо Ви хочете подивитись аніме в жанрі романтика , ось деякі рекомендації

АНІМЕ В ЖАНРІ РОМАНТИКА :

Дівчина і ще дівчина 2

DUB

Рік виходу аніме: 2023

Серій: 9 з 12 (24 хв.)

Категорія: Комедія, Романтика, Школа, Шьонен

Переклад: Surgeon

Ролі озвучували: Surgeon, Eternal Rage, Lil_Che

8.2/10 (35)

Продовження однойменного тайтлу.

Біла ніч

DUB

Рік виходу аніме: 2018

Серій: 4 з 25 (15 хв.)

Категорія: Романтика, Школа

Переклад: ooddworld

Ролі озвучували: Andrew, SnowArt_dub, Blooming Dog, OLEGizi4, Torichaan, RIHA LEE, Greedy

5.5/10 (2)

Лін Лун навчається в школі, на вигляд вона проста дівчинка, ось тільки постійно потрапляє в неймовірні ситуації

Рисунок 9 – Результат обробки запиту на аніме за жанром

І нарешті розглянемо прощання.

Лістинг коду:

```
# Заданий перелік слів
[...]
bye_words = ['до побачення', 'бувай', 'дякую', 'на все добре']

# Обробка голосових команд
def process_voice_command(text):
    # Якщо 'вітаю'
    [...]
    # Якщо 'дубляж'
    [...]
    # Якщо 'довге'
    [...]
    # Якщо 'новинка'
    [...]
```



```
# Якщо 'жанр'
[...]
# Якщо 'бувай'
elif any(word in text.lower() for word in bye_words):
    print('Anime Helper: До побачення! Гарного дня!')
    return True
```

Результат:

У результаті бот прощається і завершує роботу.

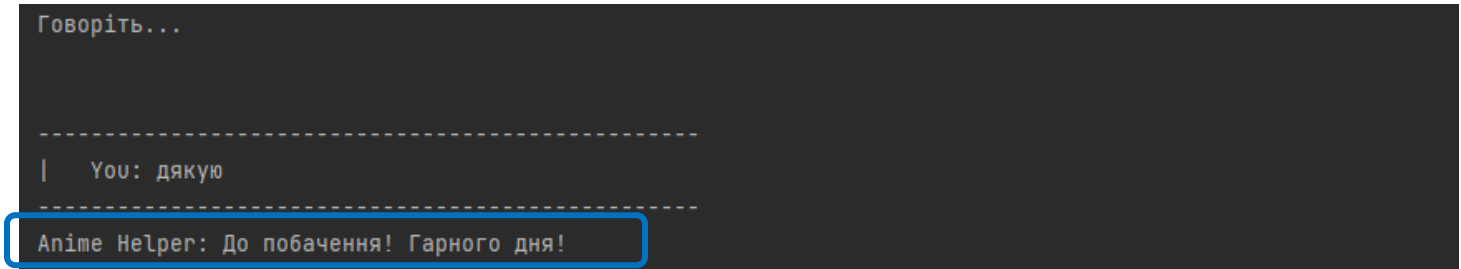


Рисунок 10 – Результат обробки запиту прощання

Також слід передбачити варіант, якщо жодне з слів-тригерів не буде використане і жоден з описаних сценаріїв не трапиться.

Лістинг коду:

```
# Заданий перелік слів
[...]

# Обробка голосових команд
def process_voice_command(text):
    # Якщо 'вітаю'
    [...]
    # Якщо 'дубляж'
    [...]
    # Якщо 'довге'
    [...]
    # Якщо 'новинка'
    [...]
    # Якщо 'жанр'
    [...]
    # Якщо 'бувай'
    [...]
    else:
        print('Anime Helper: Мені потрібно більше подробиць, щоб дати якісну пораду з вибору аніме. Опиши будь ласка детальніше, що ти шукаєш?')
        return False
```

Результат:

У результаті бот просить надати додаткові подробиці.

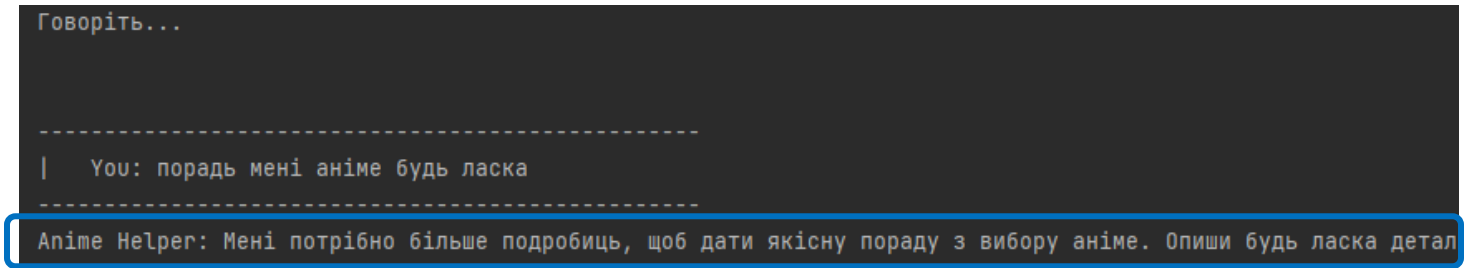


Рисунок 11 – Результат додаткового сценарія

Функція головних викликів

Створимо основну функцію для запуску системи розпізнавання голосу.

Лістинг коду:

```
# Головні виклики
def main():
    end_program = False
    while not end_program:
        audio = capture_voice_input()
        text = convert_voice_to_text(audio)
        end_program = process_voice_command(text)

if __name__ == '__main__':
    main()
```

Тепер при запуску програми, система розпізнавання голосу почне прослуховувати голосові команди і бот-асистент почне давати текстові відповіді на голосові запити.

Висновок:

Під час виконання лабораторної роботи сформульовано задачу з власного досвіду професійної роботи – розробка bot-асистента, що допомагає обрати аніме для перегляду.

Розроблено програмний скрипт парсеру аніме-сайта, який зчитує з головної сторінки 11 перших аніме-тайтлів і зберігає результат парсингу в тимчасовому текстовому файлі. Результат парсингу підлягає обробці та очищенню від зайвих символів та пустих рядків і зберігається в новому текстовому файлі для подальшої роботи з ботом.

Розроблено структуру та функціонал бота-асистента, спілкування з яким відбувається шляхом надавання боту голосових запитів. Написано програмний скрипт для розпізнавання природної мови та належної роботи бота-асистента. Передбачено декілька слів-тригерів, що запускають кожен із сценаріїв роботи програми.

Здійснено тестування кожного сценарія виконання бота-асистента, результати якого наведено в звіті. Розроблений бот-асистент чудово справляється з поставленою задачею.