

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система аналізу контенту соціальних мереж на основі телеграм-боту»

на здобуття освітнього ступеня магістра
зі спеціальності 122 Комп'ютерні науки
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні науки
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

(підпис)

Вадим Шпінь
(Ім'я, ПРІЗВИЩЕ здобувача)

Виконав:
здобувач вищої освіти
група КНДМ-62

Вадим Шпінь

Керівник:
науковий ступінь,
вчене звання

Сергій Прокопов
к.т.н., доцент

Рецензент:
науковий ступінь,
вчене звання

(Ім'я, ПРІЗВИЩЕ)

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерних наук

Ступінь вищої освіти Магістр

Спеціальність Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри Комп'ютерних наук

_____ Віктор ВИШНІВСЬКИЙ
« _____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Шпінь Вадим Анатолійович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Система аналізу контенту соціальних мереж на основі телеграм-боту»

керівник кваліфікаційної роботи Сергій Прокопов к.т.н доцент,

(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2024р. №320

2. Строк подання кваліфікаційної роботи «15» грудня 2024р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, методи аналізу даних з соціальних мереж, статистичні методи та інструменти для візуалізації даних, а також підходи до програмної інженерії.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1. Дослідження методів аналізу контенту соціальних мереж та їх реалізація за допомогою програмного забезпечення.

4.2. Проектування архітектури телеграм-бота та підбір технологій для його реалізації.

4.3. Розробка прототипу телеграм-бота для збору та аналізу даних із соціальних мереж.

5. Перелік графічного матеріалу: *презентація*

5.1 Актуальність та мета.

5.2 Методологія.

5.3 Архітектура Telegram-бота.

5.4 Приклад інтерфейсу бота.

5.5 Порівняння з іншими інструментами.

5.6 Висновки.

6. Дата видачі завдання «16» вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	16.09-27.09.24	
2	Вибір найбільш підходящих методів та інструментів для аналізу і візуалізації даних.	30.09-11.10.24	
3	Вибір технологій та інструментів для реалізації телеграм-бота і системи аналізу.	14.10-25.10.24	
4	Розробка API для збору та обробки даних із соціальних мереж.	28.10-08.11.24	
5	Розробка функціональності телеграм-бота для взаємодії з користувачами.	11.11-22.12.24	
6	Тестування та налагодження всіх компонентів системи для забезпечення стабільної роботи.	25.11-29.11.24	
7	Оформлення роботи: вступ, висновки, реферат	02.12-06.12.24	
8	Розробка демонстраційних матеріалів	09.12-13.12.24	

Здобувач вищої освіти

_____ (підпис)

Вадим Шпінь

(Ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи

_____ (підпис)

Сергій Прокопов

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 76 стор., 2 табл., 34 рис., 23 джерел.

Наукове завдання – розробити та реалізувати систему аналізу контенту соціальних мереж, яка базується на телеграм-боті, здатному автоматизовано збирати, обробляти та представляти інформацію з різних соціальних платформ.

Мета роботи – створити телеграм-бот, який автоматично аналізує контент із соціальних мереж і надає користувачам інформацію про тренди, популярні публікації та статистику.

Об'єкт дослідження – контент соціальних мереж, який включає публікації, тренди, статистику популярності та інші дані, що доступні для аналізу через API платформ.

Предмет дослідження – технології проведення аналізу та візуалізації даних.

Методи дослідження – аналіз API соціальних мереж, системний аналіз для проєктування бота, обробка даних, програмування на Python.

Короткий зміст роботи: розглянуто способи збору даних через API платформ, таких як YouTube та Twitter, а також методи їх обробки та відображення у зручному форматі. Описано архітектуру системи, використані програмні засоби та результати тестування бота на реальних даних.

Ключові слова: телеграм-бот, аналіз контенту, соціальні мережі, API, тренди, обробка даних, автоматизація, Python.

ABSTRACT

Text part of the master's qualification work: 76 pages, 2 tables, 34 figures, 23 sources.

Scientific task is to develop and implement a system for analyzing the content of social networks, which is based on a Telegram bot capable of automatically collecting, processing and presenting information from various social platforms.

Goal of the work is to create a Telegram bot that automatically analyzes content from social networks and provides users with information about trends, popular publications and statistics.

Object of research – the content of social networks, which includes publications, trends, popularity statistics and other data available for analysis through the platform's API.

Subject of research – the technology of data analysis and visualization.

Research methods – API analysis of social networks, system analysis for bot design, data processing, Python programming.

Summary of the work: ways of collecting data through the API of platforms such as YouTube and Twitter, as well as methods of their processing and display in a convenient format are considered. The architecture of the system, the software tools used, and the results of testing the bot on real data are described.

Keywords: Telegram bot, content analysis, social networks, API, trends, data processing, automation, Python.

ЗМІСТ

ВСТУП.....	9
1 ТЕОРЕТИЧНІ ОСНОВИ ТА ДОСЛІДЖЕННЯ КОНТЕНТУ СОЦІАЛЬНИХ МЕРЕЖ.....	12
1.1 Теоретичні основи.....	12
1.2 Дослідження контенту	20
1.3 Тенденції та майбутнє дослідження контенту соціальних мереж	27
2 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРОБКИ.....	33
2.1 Мова програмування Python	33
2.2 PyCharm	36
2.3 Бібліотеки Python	40
3 РОЗРОБКА ТЕЛЕГРАМ-БОТУ	50
3.1 Проектування архітектури телеграм-боту	50
3.2 Реалізація функціоналу телеграм-боту для збору та обробки даних.....	71
3.3. Аналіз ефективності телеграм-боту порівняно з іншими інструментами ...	78
ВИСНОВКИ	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	85

ВСТУП

У сучасному світі соціальні мережі стали невід'ємною частиною повсякденного життя мільйонів людей. Вони надають можливість спілкування, обміну інформацією, а також є потужним інструментом для дослідження суспільних настроїв та аналізу контенту. Соціальні мережі, такі як Facebook, Twitter, Instagram та YouTube, мають мільярди активних користувачів по всьому світу і відіграють важливу роль у формуванні громадської думки та поширенні інформації.

Одним із найпопулярніших інструментів для автоматизації та взаємодії з користувачами є телеграм боти. Вони дозволяють автоматично збирати, обробляти та аналізувати великий обсяг даних з різних соціальних платформ. Телеграм боти можуть виконувати різноманітні функції: від простого інформування користувачів до складних аналітичних задач, що включають обробку великих масивів даних та видобування корисної інформації.

Актуальність теми. Зростання обсягів інформації в соціальних мережах ставить перед дослідниками виклики щодо швидкості обробки, якості аналізу та забезпечення точності отриманих даних. Telegram як платформа має унікальні можливості для створення ботів, що дозволяють автоматизувати процес збору та аналізу даних. Використання телеграм-ботів для дослідження контенту може значно знизити затрати часу та ресурсів, одночасно забезпечуючи інформативність результатів. Ця тема є особливо актуальною для аналізу інформаційних кампаній, виявлення трендів та моніторингу громадської думки.

Мета дослідження. Розробити та проаналізувати телеграм-бот як інструмент для дослідження контенту соціальних мереж, оцінити його ефективність та практичну значущість. В рамках дослідження буде розглянуто інтеграцію з популярними соціальними мережами, такими як Twitter та YouTube, а також методи збору і обробки даних. Особлива увага буде приділена аналізу трендів, виявленню популярних тем та настроїв у суспільстві.

Об'єкт дослідження. Процеси аналізу контенту соціальних мереж.

Предмет дослідження. Телеграм-бот як інструмент автоматизації дослідження контенту соціальних мереж.

Завдання дослідження:

1. Провести огляд існуючих методів аналізу контенту соціальних мереж та роль ботів у цьому процесі.
2. Розробити телеграм-бота для збору даних з соціальних мереж.
3. Інтегрувати телеграм-бота з API популярних соціальних мереж, таких як Twitter та YouTube.
4. Проаналізувати ефективність телеграм-боту порівняно з іншими інструментами.

Методи дослідження

- Метод моделювання – для розробки алгоритмів телеграм-боту
- Методи збору даних – використання API соціальних мереж для автоматичного збору даних про публікації, коментарі, лайки та інші взаємодії.
- Методи обробки даних: Використання бібліотек Python, таких як pandas та numpy, для очищення, нормалізації та аналізу зібраних даних.
- Порівняльний аналіз – оцінка ефективності інструменту щодо альтернатив

Джерела дослідження. Для проведення дослідження використовувалися різні джерела інформації. Аналізувалися матеріали з газет, таких як "Дзеркало тижня", а також журнали, наприклад, "Forbes Україна". Вивчалися телевізійні та радіопрограми на каналах "1+1", "ICTV", "СТБ. Значну увагу приділяли інтернет-ресурсам, включаючи офіційний сайт Telegram, публічні групи та канали в Telegram, Також досліджувалися друковані праці авторів, що стосуються використання соціальних мереж, аналізу контенту та кібербезпеки. Для реалізації проекту були вивчені приклади телеграм-ботів, таких як боти для новинних розсилок, моніторингу соціальних мереж та збору даних. Ці джерела надали необхідну інформацію для проведення всебічного дослідження контенту соціальних мереж за допомогою телеграм-боту.

Наукова новизна одержаних результатів. Полягає у впровадженні інноваційного підходу до аналізу та моніторингу контенту в соціальних мережах за допомогою телеграм-боту. У рамках дослідження було розроблено та реалізовано унікальний телеграм-бот, який автоматизує процес збору, аналізу та візуалізації даних з різних соціальних мереж. Це дозволяє значно скоротити час та ресурси, необхідні для проведення досліджень у цій сфері. Таким чином, результати дослідження роблять значний внесок у розвиток методології аналізу контенту соціальних мереж та демонструють перспективи використання телеграм-ботів у цій галузі.

Практичне значення одержаних результатів. Полягає у створенні ефективного інструменту для автоматизованого збору та аналізу даних з соціальних мереж. Розроблений телеграм-бот дозволяє оперативно отримувати та обробляти великі обсяги інформації, що є важливим для маркетингових досліджень, моніторингу громадської думки, аналізу поведінки користувачів та вивчення трендів. Завдяки цьому інструменту можна швидко реагувати на зміни в інформаційному просторі, що є ключовим для успішної адаптації стратегій комунікації та маркетингу.

1 ТЕОРЕТИЧНІ ОСНОВИ ТА ДОСЛІДЖЕННЯ КОНТЕНТУ СОЦІАЛЬНИХ МЕРЕЖ

Цей розділ слугує основою для розуміння сучасних тенденцій та процесів, що формуються в цифровому світі. Він сприяє глибшому усвідомленню того, як у соціальних мережах відбувається створення, поширення та споживання контенту.

1.1 Теоретичні основи

Соціальні мережі – це віртуальні простори, де люди можуть створювати свої цифрові образи, спілкуватися, обмінюватися різноманітними матеріалами та взаємодіяти один з одним. Ці матеріали, які ми називаємо контентом, можуть бути будь-якими: від простих текстових повідомлень до складних мультимедійних проектів [1, с. 48].

Процес створення нового контенту користувачами і називається генерацією контенту. Він є рушійною силою соціальних мереж, оскільки постійно поповнює їх новими ідеями, думками та інформацією. Після створення, контент починає свій шлях до аудиторії – це процес, який називається поширенням контенту. Він може відбуватися через різноманітні канали і досягати великої кількості людей.

Останнім етапом цього циклу є споживання контенту. Саме в цей момент користувачі сприймають, аналізують і інтерпретують інформацію, яку вони отримали. Таким чином, соціальні мережі є динамічними системами, де постійно відбувається взаємодія між творцями контенту, його поширювачами та споживачами.

Соціальні мережі стали важливим елементом сучасного життя, надаючи людям можливість спілкування, обміну інформацією та участі в суспільному житті. Вони дозволяють користувачам створювати профілі, публікувати контент,

взаємодіяти з іншими користувачами через лайки, коментарі та репости. Популярні соціальні мережі, такі як Facebook, Twitter, Instagram, YouTube, мають мільярди активних користувачів по всьому світу.

Ідея створення віртуальних спільнот, де люди можуть спілкуватися та ділитися інформацією, виникла ще задовго до появи інтернету в сучасному розумінні. Проте саме з розвитком мережі Інтернет наприкінці XX століття з'явилися перші спроби реалізувати цю ідею на практиці.

Одним із перших експериментів у цій галузі стала соціальна мережа Six Degrees (Рисунок 1.1), запущена в 1997 році. Ця платформа дозволяла користувачам створювати профілі, додавати друзів та переглядати їхні профілі. Хоча Six Degrees не досягла масової популярності, вона заклала фундамент для подальшого розвитку соціальних мереж, продемонструвавши потенціал такого типу платформ.

Наступним важливим етапом у розвитку соціальних мереж став запуск MySpace у 2003 році. MySpace відрізнявся від Six Degrees більш широкими можливостями для персоналізації профілів, що дозволяло користувачам створювати унікальні сторінки з музикою, відео та іншим контентом. MySpace швидко набрав популярності, особливо серед молоді.

Паралельно з MySpace з'явилася ще одна знакова платформа – LinkedIn. На відміну від MySpace та Six Degrees, LinkedIn була орієнтована на професійну комунікацію. Ця мережа дозволяла користувачам створювати детальні резюме, шукати роботу та підтримувати зв'язки з колегами.

Однак справжній прорив у розвитку соціальних мереж стався з появою Facebook у 2004 році. Спочатку доступна лише для студентів американських університетів, Facebook швидко поширилася на весь світ. Головною відмінністю Facebook від попередніх платформ була її відкритість та простота використання. Користувачі могли легко створювати профілі, додавати друзів, ділитися фотографіями, відео та іншим контентом. Крім того, Facebook запровадив нові

функції, такі як групи та події, що сприяло створенню різноманітних спільнот за інтересами.

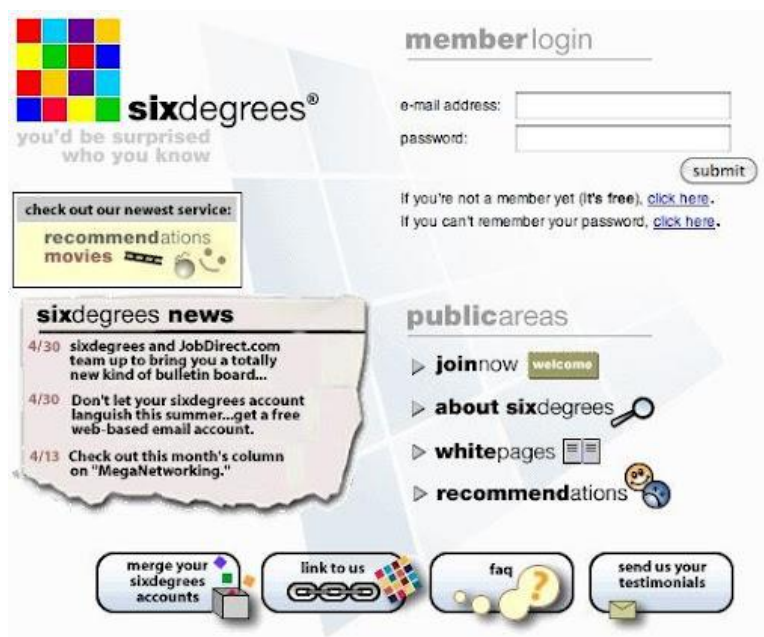


Рисунок 1.1 – Інтерфейс соціальної мережі «Six Degrees»

Секрет успіху Facebook криється в його здатності задовольнити базові людські потреби в спілкуванні та зв'язку [2, с. 206]. Інтуїтивно зрозумілий інтерфейс зробив цю соціальну мережу доступною навіть для тих, хто не мав досвіду користування інтернетом. Кожен міг легко створити свій профіль, знайти друзів та родичів, а також ділитися новинами та фотографіями зі свого життя.

Важливу роль у популяризації Facebook зіграла його відкритість. Користувачі могли легко знаходити один одного завдяки системі пошуку за іменем, навчальним закладом чи роботою. Це сприяло тому, що люди об'єднувалися в групи за інтересами, відновлювали старі знайомства та заводили нові.

Ще одним фактором успіху Facebook став широкий спектр функцій, які він пропонував. Крім можливості спілкуватися в чаті та обмінюватися повідомленнями, користувачі могли створювати події, групи, грати в ігри та багато іншого. Така різноманітність задовольняла потреби різних аудиторій і робила Facebook універсальною платформою для спілкування.

І, звичайно ж, не можна не згадати про вірусне поширення Facebook. Завдяки рекомендаціям друзів, соціальна мережа швидко зростала в популярності. Кожен

новий користувач приводив з собою ще кількох, і таким чином Facebook охопив мільйони людей по всьому світу.

Поєднання простоти використання, відкритості, різноманітності функцій та вірусного поширення зробило Facebook феноменом сучасного інтернету і на тривалий час визначило напрямок розвитку соціальних мереж.

Поява соціальних мереж, а особливо Facebook, кардинально змінила спосіб спілкування людей. Соціальні мережі стали невід'ємною частиною повсякденного життя мільярдів людей у всьому світі. Вони вплинули на такі сфери, як політика, бізнес, культура та навіть міжособистісні стосунки.

Теоретичні засади дослідження соціальних мереж базуються на комплексній сукупності принципів та моделей, які прагнуть пояснити механізми створення, поширення та сприйняття контенту в цифровому середовищі [3]. Ці засади охоплюють широкий спектр дисциплін, зокрема соціологію, комунікативістику та психологію. Зокрема, соціальні теорії дозволяють зрозуміти, як суспільні структури, норми та взаємовідносини впливають на контент, що створюється та поширюється в мережі. Теорії комунікації, в свою чергу, аналізують процеси передачі інформації, допомагаючи розкрити механізми взаємодії між авторами контенту та аудиторією. А теорії впливу та переконання фокусуються на тому, як контент може формувати думки, поведінку та прийняття рішень користувачів.

Соціальні мережі впливають на різні аспекти життя, включаючи політику, економіку, культуру та особисте життя людей:

Політика: Соціальні мережі стали важливим інструментом для політичних кампаній, громадянської активності та мобілізації населення. Вони дозволяють політикам і активістам безпосередньо звертатися до громадськості, поширювати свої «месиджі» та організовувати масові заходи [4, с. 12]. Наприклад, під час протестних рухів у різних країнах соціальні мережі використовувалися для координації дій і поширення інформації.

Економіка: Соціальні мережі відіграють важливу роль у бізнесі та маркетингу. Вони дозволяють компаніям взаємодіяти зі своїми клієнтами, просувати товари та послуги, а також аналізувати ринок і конкурентів [5, с. 5].

Соціальні мережі також стали платформою для електронної комерції, де користувачі можуть купувати товари безпосередньо через платформи, такі як Instagram та Facebook.

Культура: Соціальні мережі сприяють поширенню культурних цінностей, ідей та трендів [6, с. 225]. Вони дозволяють людям ділитися своїми творчими роботами, обговорювати культурні події та знаходити однодумців. Наприклад, YouTube став платформою для музикантів, художників і відеоблогерів, де вони можуть публікувати свої роботи і взаємодіяти зі своєю аудиторією.

Соціальні мережі надають користувачам різноманітні функції, які сприяють їх активності та взаємодії:

1. Створення профілів: Користувачі можуть створювати особисті профілі, де вони вказують свою інформацію, такі як ім'я, місце проживання, інтереси та фотографії. Профілі допомагають користувачам представляти себе іншим і знаходити людей з подібними інтересами.

2. Публікація контенту: Соціальні мережі дозволяють користувачам публікувати різноманітний контент, включаючи текстові пости, фотографії, відео та посилання. Цей контент може бути видимим для друзів, підписників або ширшої аудиторії залежно від налаштувань приватності [7, с. 4].

3. Взаємодія з контентом: Користувачі можуть взаємодіяти з контентом інших через лайки, коментарі, репости та інші форми взаємодії. Це сприяє активній участі користувачів і створенню спільнот навколо певних тем або інтересів.

4. Месенджери та чати: Багато соціальних мереж включають функції обміну повідомленнями, що дозволяє користувачам спілкуватися в режимі реального часу. Це можуть бути приватні повідомлення, групові чати або обговорення в коментарях до постів.

5. Групи та спільноти: Соціальні мережі дозволяють створювати групи та спільноти за інтересами, де користувачі можуть обговорювати певні теми, ділитися інформацією та організовувати заходи. Наприклад, групи у Facebook або спільноти на Reddit.

Соціальні мережі мають як позитивний, так і негативний вплив на суспільство:

Позитивний вплив:

Соціальні мережі стали невід'ємною частиною нашого життя, впливаючи на різні аспекти нашого буття. Одним з найважливіших позитивних аспектів соціальних мереж є їхня здатність підтримувати соціальні зв'язки. Незалежно від географічної відстані, люди можуть легко залишатися на зв'язку зі своїми друзями та родиною, ділитися новинами, фотографіями та іншими подіями зі свого життя. Це особливо важливо для людей, які живуть далеко від своїх близьких або мають розгалужену соціальну мережу.

Крім того, соціальні мережі є потужним інструментом для отримання інформації [8, с. 47]. Завдяки їм, користувачі мають доступ до величезної кількості новин та інформації з різних джерел. Це дозволяє бути в курсі останніх подій у світі, формувати свою думку та брати участь у обговореннях актуальних питань. Така інформаційна доступність сприяє підвищенню обізнаності та розширенню світогляду.

Ще один позитивний аспект соціальних мереж – це можливість для самовираження. Кожен користувач може створити свій унікальний образ в Інтернеті, ділитися своїми думками, ідеями та творчими досягненнями [9, с.121]. Це сприяє розвитку індивідуальності, творчості та дозволяє знайти однодумців. Соціальні мережі стали платформою для самореалізації та розвитку талантів.

Таким чином, соціальні мережі відіграють важливу роль у сучасному суспільстві. Вони допомагають людям підтримувати зв'язки, отримувати інформацію та самовиражатися. Однак, як і будь-який інструмент, соціальні мережі мають свої недоліки і їх використання потребує відповідального підходу.

Негативний вплив:

Соціальні мережі, незважаючи на свої очевидні переваги, також несуть у собі ряд потенційних загроз. Однією з найбільших проблем є питання приватності. Користувачі, реєструючись на платформах, добровільно надають величезний обсяг особистої інформації. Ці дані можуть бути використані для цілеспрямованої

реклами, а також потрапити в руки зловмисників. Така ситуація викликає серйозні побоювання щодо захисту приватності та безпеки користувачів.

Ще однією серйозною проблемою, пов'язаною з використанням соціальних мереж, є кібербулінг. Анонімність та доступність цих платформ створюють сприятливе середовище для проявів агресії та образ. Люди можуть стати жертвами цькування, принижень та інших форм онлайн-насилюства. Це може мати серйозні наслідки для психічного здоров'я постраждалих, включаючи депресію, тривогу та навіть суїцидальні думки.

Також надмірне використання соціальних мереж може призвести до розвитку залежності. Постійний доступ до величезного обсягу інформації та спілкування з друзями онлайн може відволікати від реального життя, знижувати продуктивність та негативно впливати на психічне здоров'я. Люди можуть відчувати постійну потребу перевіряти свої акаунти, що призводить до розладів сну, зниження концентрації уваги та інших проблем [10, с. 154].

Тобто, соціальні мережі, незважаючи на свої переваги, можуть мати й негативний вплив на життя людини. Для того, щоб уникнути цих проблем, важливо усвідомлювати ризики, пов'язані з їх використанням, та вживати заходів для захисту своєї приватності та психічного здоров'я.

Для аналізу соціальних мереж використовуються різні інструменти та технології. Одним із найпотужніших інструментів є API (Application Programming Interface), що надають доступ до великих обсягів даних, таких як публікації, коментарі, лайки та інші метрики. API дозволяє автоматизувати процес збору даних та надає можливість створювати додатки для аналізу та візуалізації цих даних.

Інструменти для аналізу соціальних мереж включають:

1. API соціальних мереж: Наприклад, Twitter API, Facebook Graph API, YouTube Data API. Вони дозволяють отримувати дані про активність користувачів, популярність публікацій, взаємодії тощо [11, с. 58].

2. Аналітичні платформи: Такі як Google Analytics, які дозволяють відстежувати взаємодію користувачів з контентом.

3. Інструменти для обробки даних: Pandas, Numpy, які використовуються для очищення та аналізу даних [12].

4. Інструменти для візуалізації даних: Matplotlib, Seaborn, Plotly для створення графіків, діаграм та інших візуальних представлень даних.

Телеграм боти є потужним інструментом для автоматизації різних задач, включаючи аналіз соціальних мереж. Вони дозволяють створювати інтерактивні додатки, що можуть виконувати різні функції, такі як збір даних з соціальних мереж, аналіз контенту, надання користувачам інформації в реальному часі. Використання телеграм ботів спрощує процес взаємодії з користувачами та дозволяє ефективно обробляти великі обсяги даних. Телеграм-боти значно спрощують наше життя та роботу завдяки своїм численним перевагам. Перш за все, вони дозволяють автоматизувати рутинні завдання. Уявіть собі, що замість того, щоб вручну збирати та обробляти великі обсяги даних, ви можете доручити це боту. Це значно економить час та звільняє вас для виконання більш творчих завдань.

Швидкий зворотний зв'язок – ще одна важлива перевага ботів. Отримати потрібну інформацію можна миттєво, не чекаючи на відповідь людини. Це особливо актуально в умовах, коли час – гроші. Боти можуть надавати інформацію про погоду, курси валют, розклади транспорту та багато іншого в режимі реального часу.

Масштабованість – це ще одна сильна сторона ботів. Вони здатні обробляти великі обсяги даних та одночасно взаємодіяти з великою кількістю користувачів. Це робить їх ідеальним інструментом для бізнесу, який хоче масштабувати свої операції.

Боти легко інтегруються з різними API та додатками. Це означає, що їх функціональність можна розширювати практично безмежно. Наприклад, бот може бути інтегрований з системою обліку, CRM-системою або будь-яким іншим програмним забезпеченням, яке використовується у вашому бізнесі.

Телеграм-боти є універсальним інструментом, який може бути використаний в різних сферах діяльності. Вони дозволяють автоматизувати процеси, підвищити

ефективність роботи, розширити можливості бізнесу та покращити взаємодію з клієнтами.

Розуміння теоретичних основ і дослідження контенту соціальних мереж є критично важливим для ефективного використання цих платформ у різних сферах життя. Вивчення процесів створення, поширення та споживання контенту дозволяє розробляти ефективні стратегії для досягнення цілей, будь то особисті, бізнесові чи суспільні.

1.2 Дослідження контенту

Дослідження контенту соціальних мереж є ключовим аспектом для розуміння, як інформація створюється, поширюється і сприймається в цифровому середовищі. Цей підрозділ розглядає різні методи і підходи до аналізу контенту, а також інструменти, які використовуються для цього.

Існує кілька основних методів дослідження контенту в соціальних мережах:

Контент-аналіз є одним з найпоширеніших методів дослідження контенту [13, с. 32]. Він може бути як кількісним, так і якісним. Кількісний контент-аналіз включає підрахунок частоти появи певних слів, фраз або тем у текстах. Якісний контент-аналіз фокусується на інтерпретації змісту та значень.

- **Кількісний контент-аналіз:** Використовується для визначення частотності певних ключових слів або тем у великій кількості текстів. Наприклад, можна дослідити, як часто згадується певна тема в твітах за останній місяць.

- **Якісний контент-аналіз:** Передбачає глибоке вивчення текстів для виявлення прихованих значень, мотивів та контексту. Наприклад, можна аналізувати коментарі під публікаціями для розуміння настроїв і емоцій аудиторії.

Семантичний аналіз включає вивчення значень слів і фраз у контексті. Він дозволяє визначити тональність (позитивну, негативну або нейтральну) та емоційне забарвлення тексту.

- Аналіз тональності: Визначення емоційного тону тексту (позитивний, негативний, нейтральний). Наприклад, аналіз відгуків на продукт для виявлення загальної задоволеності клієнтів.
- Аналіз ключових слів: Виявлення основних тем і понять, які часто згадуються разом з певним ключовим словом. Це допомагає зрозуміти асоціації та контекст використання.

Мережевий аналіз включає вивчення взаємодій між користувачами в соціальних мережах. Це дозволяє визначити впливових осіб (інфлюенсерів), структуру мережі та шляхи поширення інформації.

- Аналіз соціальних графів: Дослідження зв'язків між користувачами, виявлення центральних вузлів та ключових інфлюенсерів.
- Аналіз поширення інформації: Вивчення, як інформація поширюється в мережі, які користувачі виступають в ролі "містків" між різними групами.

Аналіз залученості включає вимірювання взаємодії користувачів з контентом, таких як лайки, коментарі, поширення і перегляди. Це допомагає оцінити ефективність контенту і зрозуміти, що викликає найбільший інтерес у аудиторії.

- Метрики залученості: Лайки, коментарі, поширення, перегляди, кліки, час перегляду.
- Аналіз поведінки користувачів: Вивчення дій користувачів, таких як час взаємодії, частота відвідувань, типи контенту, що викликають найбільший інтерес.

Одним із ключових етапів дослідження контенту соціальних мереж є збір даних. Для цього використовуються різні методи, включаючи API запити до соціальних мереж, веб-скрапінг, а також інші технології для автоматизації збору інформації. У цьому підрозділі детально розглянемо кожен з цих методів.

API (Application Programming Interface) - це набір інструментів і протоколів, який дозволяє одному програмному додатку взаємодіяти з іншим. У контексті

соціальних мереж, API дозволяє розробникам отримувати доступ до даних користувачів, публікацій, коментарів, лайків та інших взаємодій. Використання API запитів є одним із найефективніших методів збору даних, оскільки вони надають структуровану інформацію безпосередньо з серверів соціальних мереж [14].

Приклади API соціальних мереж:

Twitter API: Дозволяє отримувати інформацію про твіти, хештеги, згадки, профілі користувачів, а також взаємодії, такі як ретвіти, лайки та відповіді. Twitter API надає декілька кінцевих точок (endpoints), кожна з яких призначена для певного типу даних. Для доступу до API необхідно створити обліковий запис розробника у Twitter та отримати API ключі.

Facebook Graph API: Забезпечує доступ до даних користувачів, сторінок, груп, подій, постів, коментарів та реакцій. Graph API використовує графову модель для представлення даних, де вузли є об'єктами (користувачі, сторінки, пости), а ребра - взаємозв'язками між ними. Для роботи з API необхідно отримати токен доступу, який видається після авторизації.

YouTube Data API: Дозволяє отримувати інформацію про відео, канали, плейлисти, коментарі та інші метрики. API надає доступ до різних кінцевих точок, які дозволяють виконувати пошук, отримувати метадані про відео, а також аналізувати взаємодії користувачів з контентом. Для використання API необхідно створити проект у Google Cloud Platform та отримати API ключ.

Переваги використання API запитів:

- Структуровані дані: API надає дані у форматі JSON або XML, що спрощує їх обробку та аналіз.
- Актуальність: Дані отримуються безпосередньо з серверів соціальних мереж, що гарантує їх актуальність та точність.
- Швидкість і ефективність: API запити дозволяють швидко отримувати великі обсяги даних без необхідності ручного збору.

Недоліки використання API запитів:

- Обмеження за кількістю запитів: Багато соціальних мереж встановлюють ліміти на кількість запитів, які можна виконати за певний період часу.

- **Необхідність автентифікації:** Для доступу до даних через API зазвичай потрібно пройти процес автентифікації та отримати токени доступу.
- **Зміни в API:** Соціальні мережі можуть змінювати свої API, що може вимагати оновлення коду для коректної роботи.

Веб-скрапінг (Web Scraping) - це технологія для автоматичного збору інформації з веб-сторінок; це перетворення інформації з веб-сторінок, призначених для перегляду в браузері. Зазвичай виконується комп'ютерними програмами, які імітують поведінку користувача в Інтернеті, або зв'язавшись із сервером безпосередньо через протокол HTTP, або керуючи повністю захищеним Web-браузером [15]. Але може і бути скрапінг за допомогою копіювання даних людиною. Цей метод використовується, коли API не надає необхідних даних або коли потрібна додаткова інформація, яка не доступна через офіційні API.

Основні інструменти для веб-скрапінгу:

1. **Beautiful Soup:** Бібліотека Python для парсингу HTML та XML документів. Вона дозволяє легко витягувати дані з веб-сторінок, використовуючи синтаксичний аналіз дерева документа.
2. **Scrapy:** Потужний фреймворк для веб-скрапінгу, який дозволяє створювати ефективні веб-краулери та парсери. Scrapy підтримує асинхронне виконання запитів, що дозволяє збирати дані з великої кількості веб-сторінок швидко та ефективно [16].
3. **Selenium:** Інструмент для автоматизації браузера, який часто використовується для веб-скрапінгу. Selenium дозволяє взаємодіяти з веб-сторінками так, як це робить користувач, що дозволяє збирати дані навіть з динамічних сторінок, що використовують JavaScript.

Переваги веб-скрапінгу:

Веб-скрапінг надає безпрецедентний доступ до інформації, яка розміщена на веб-сторінках. Незалежно від того, чи надає веб-сайт відкритий API для доступу до даних, за допомогою скрапінгу можна витягти будь-яку необхідну інформацію. Це особливо корисно для аналізу конкурентів, дослідження ринку та збору даних для наукових досліджень.

Парсери, які використовуються для веб-скрапінгу, можна налаштувати для витягування будь-якої конкретної інформації з веб-сторінок. Це дозволяє адаптувати процес збору даних до будь-яких, навіть найскладніших, вимог. Наприклад, можна збирати текстовий контент, зображення, посилання, ціни, характеристики товарів тощо.

Багато веб-сайтів надають API для доступу до своїх даних, але часто встановлюють обмеження на кількість запитів або тип даних, які можна отримати. Веб-скрапінг дозволяє обійти ці обмеження і отримати доступ до більшої кількості даних.

Недоліки веб-скрапінгу:

Використання веб-скрапінгу може порушувати умови використання веб-сайтів, що передбачають заборону на автоматизований збір даних. Це може призвести до юридичних наслідків, таких як блокування IP-адреси, подання позову або навіть кримінальне переслідування. Тому перед початком скрапінгу необхідно ретельно ознайомитись з умовами використання веб-сайтів та законодавством.

Структура веб-сторінок часто змінюється. Це означає, що парсери, які були розроблені для роботи з однією версією веб-сайту, можуть перестати працювати після оновлення. Це вимагає постійного моніторингу та оновлення парсерів.

Розробка ефективних парсерів – це складний процес, який вимагає знань програмування та розуміння структури веб-сторінок. Крім того, збір великих обсягів даних може бути ресурсоємним і вимагати потужного обладнання або хмарних сервісів.

Незважаючи на свої недоліки, веб-скрапінг залишається потужним інструментом для збору даних з Інтернету. Для успішного використання веб-скрапінгу необхідно ретельно зважити всі ризики та переваги, а також дотримуватися етичних норм і законодавства.

Інтеграція з іншими сервісами дозволяє збирати дані з різних джерел, таких як RSS-канали, новинні сайти, блоги та інші ресурси. Це може бути корисно для отримання додаткової інформації та забезпечення більш повного аналізу контенту соціальних мереж.

Приклади інтеграції:

RSS-канали: Використання RSS-каналів для автоматичного отримання оновлень з новинних сайтів, блогів та інших ресурсів. RSS-канали надають структуровану інформацію у форматі XML, що дозволяє легко інтегрувати їх у систему збору даних.

Новинні агрегатори: Використання сервісів, таких як Google News, для отримання новин з різних джерел. Агрегатори можуть надавати API для доступу до новин, що дозволяє автоматизувати процес збору інформації.

Соціальні плагіни: Використання плагінів та віджетів, які надають доступ до даних з соціальних мереж. Наприклад, плагіни для WordPress можуть надавати інформацію про взаємодії користувачів з контентом на сайті.

Інтеграція з різноманітними сервісами під час дослідження контенту соціальних мереж значно розширює можливості аналізу. Вона дозволяє отримувати більш повну картину взаємодії користувачів, виявляти нові тренди та глибше розуміти аудиторію.

Переваги:

По-перше, інтеграція забезпечує доступ до величезного обсягу даних з різних джерел. Це дозволяє не обмежуватися лише текстом публікацій, а аналізувати додаткову інформацію про користувачів, їхню поведінку, геолокацію та ін. Такий комплексний підхід дає змогу отримати більш глибоке розуміння контексту, в якому генерується контент.

По-друге, автоматизація процесів є ще однією важливою перевагою інтеграції. Використання API та інших інструментів дозволяє автоматизувати збір, обробку та аналіз даних. Це значно скорочує час, необхідний для проведення дослідження, і звільняє дослідників від рутинної роботи.

Крім того, інтеграція надає гнучкість у налаштуванні процесу збору даних. Завдяки цьому можна адаптувати систему під конкретні потреби дослідження, змінювати критерії відбору даних та масштабувати процес відповідно до обсягу даних.

Недоліки:

Однак, інтеграція з іншими сервісами має і свої недоліки. По-перше, налаштування такої системи може бути досить складним і вимагати значних технічних навичок. Необхідно розуміти принципи роботи API різних сервісів, вміти писати скрипти для збору та обробки даних, а також вирішувати технічні проблеми, які можуть виникнути під час інтеграції.

По-друге, система, яка інтегрується з різними зовнішніми сервісами, стає залежною від їхньої стабільності та функціональності. Зміни в політиці або функціональності цих сервісів можуть призвести до переривання роботи системи або зміни формату даних, що потребуватиме додаткових налаштувань.

Крім того, багато сервісів встановлюють ліміти на кількість запитів, які можна здійснювати за певний період часу. Це може обмежити обсяг даних, які можна зібрати, особливо при проведенні масштабних досліджень.

Інтеграція з іншими сервісами є потужним інструментом для дослідників контенту соціальних мереж, але вимагає зваженого підходу. З одного боку, вона дозволяє отримувати більш повну і детальну інформацію, автоматизувати процеси та підвищити ефективність досліджень. З іншого боку, вона має свої технічні складнощі, залежить від зовнішніх сервісів і може мати обмеження за кількістю даних. Тому, приймаючи рішення про інтеграцію, необхідно зважити всі за і проти та обрати оптимальне рішення для конкретного дослідження.

Після збору даних необхідно провести їх обробку та аналіз. Це включає очищення даних, видалення дублікатів, нормалізацію, а також аналіз текстів для визначення основних тем, настроїв та інших характеристик. Для цього використовуються різні інструменти та бібліотеки, такі як Pandas, Numpy, NLTK, що дозволяють ефективно обробляти та аналізувати великі обсяги даних.

Процес обробки та аналізу даних включає:

- Очищення даних: Видалення некоректних або дублікатних записів, заповнення пропусків.
- Нормалізація даних: Приведення даних до єдиного формату для спрощення подальшого аналізу.

- Аналіз тексту: Використання методів обробки природної мови (NLP) для визначення основних тем, настроїв та інших характеристик тексту. Наприклад, можна використовувати бібліотеки NLTK або spaCy для аналізу текстів.
- Аналіз взаємодій: Визначення основних патернів взаємодії користувачів з контентом, аналіз популярності публікацій тощо.

Останнім етапом є візуалізація результатів аналізу. Це дозволяє наочно представити основні висновки та тенденції, виявлені під час дослідження. Використовуються різні інструменти для візуалізації, такі як Matplotlib, Seaborn, Plotly, що дозволяють створювати графіки, діаграми та інші візуальні представлення даних.

Візуалізація результатів включає: створення графіків та діаграм (використання різних типів графіків для представлення даних, таких як лінійні графіки, гістограми, кругові діаграми тощо); інтерактивна візуалізація: (використання інструментів для створення інтерактивних графіків, що дозволяє користувачам взаємодіяти з даними); візуалізація мереж взаємодій (створення графів взаємодій користувачів, що дозволяє виявити основні патерни та зв'язки між ними).

Загалом, процес аналізу даних з соціальних мереж є складним і багатогранним завданням. Він вимагає застосування як статистичних методів, так і методів обробки природної мови. Однак, результати такого аналізу можуть бути дуже цінними для дослідників, маркетологів, соціологів та інших фахівців, які цікавляться сучасними цифровими комунікаціями.

1.3 Тенденції та майбутнє дослідження контенту соціальних мереж

Тенденції та майбутнє дослідження контенту соціальних мереж є надзвичайно динамічними та багатогранними. Важливі зміни, що відбуваються в

цій галузі, значною мірою пов'язані з розвитком технологій, зокрема штучного інтелекту, інтеграцією з іншими науковими дисциплінами та новими викликами, що виникають у зв'язку з еволюцією соціальних платформ.

Розвиток штучного інтелекту (ШІ) в аналізі соціальних даних є однією з найбільш інноваційних і динамічних сфер сучасних технологій. Штучний інтелект значно змінив підхід до дослідження та аналізу даних, зокрема в контексті соціальних мереж. Завдяки можливостям обробки великих обсягів інформації, машинного навчання та аналізу природної мови, ШІ дозволяє отримувати глибокі інсайти та робити точні прогнози на основі даних з різних соціальних платформ.

На початкових етапах розвитку ШІ застосовувалися прості алгоритми для автоматизації базових завдань, таких як класифікація текстів або виявлення спаму. Однак з часом алгоритми стали більш складними, включаючи моделі машинного навчання, які можуть навчатися на великих наборах даних і вдосконалювати свої прогнози з часом. Це дозволило значно підвищити точність і ефективність аналізу даних.

Однією з ключових областей, де ШІ продемонстрував свою ефективність, є обробка природної мови (NLP). NLP дозволяє аналізувати текстові дані з соціальних мереж, виявляти тональність повідомлень, емоційне забарвлення, ключові слова та теми. Наприклад, алгоритми аналізу тональності можуть автоматично визначати, чи є коментарі позитивними, негативними або нейтральними. Це особливо корисно для брендів і компаній, які хочуть швидко оцінювати реакцію споживачів на свою продукцію або маркетингові кампанії.

ШІ також використовується для виявлення трендів і патернів у поведінці користувачів. За допомогою алгоритмів машинного навчання можна аналізувати великі обсяги даних для виявлення закономірностей, які можуть бути неочевидними для людини. Наприклад, можна виявити, які теми стають популярними в певний період часу або які типи контенту викликають найбільшу залученість аудиторії. Це дозволяє компаніям адаптувати свою контент-стратегію відповідно до актуальних інтересів і потреб користувачів.

Однією з найперспективніших галузей є використання глибинного навчання для аналізу відеоконтенту. Відео стає все більш популярним форматом контенту в соціальних мережах, і аналіз відеоматеріалів дозволяє отримати додаткові інсайти. Алгоритми глибинного навчання можуть аналізувати зображення та відео для виявлення об'єктів, сцен, осіб та навіть емоцій, що виражаються на обличчях. Це відкриває нові можливості для маркетингових досліджень, виявлення трендів та оцінки впливу відеоконтенту на аудиторію.

Крім того, розвиток ШІ сприяє автоматизації багатьох процесів, пов'язаних з аналізом соціальних даних. Наприклад, чат-боти та віртуальні асистенти, що використовують ШІ, можуть автоматично відповідати на запити користувачів, збирати зворотний зв'язок і навіть проводити опитування. Це дозволяє компаніям знижувати витрати на обслуговування клієнтів і підвищувати якість взаємодії з аудиторією.

Загалом, розвиток штучного інтелекту в аналізі соціальних даних відкриває безліч можливостей для дослідження та використання інформації. Завдяки ШІ можна отримувати глибокі інсайти, автоматизувати рутинні завдання, аналізувати великі обсяги даних з високою точністю та швидкістю. Це дозволяє компаніям, дослідникам та аналітикам приймати обґрунтовані рішення, що базуються на реальних даних, та адаптувати свої стратегії відповідно до змін у поведінці та інтересах користувачів соціальних мереж.

Інтеграція досліджень контенту соціальних мереж з іншими науковими дисциплінами, такими як психологія, соціологія та лінгвістика, відкриває нові горизонти для комплексного розуміння соціальних явищ у цифровому середовищі. Кожна з цих дисциплін вносить свій унікальний вклад у дослідження контенту, дозволяючи отримати більш глибокі інсайти та розробляти більш ефективні стратегії для взаємодії з аудиторією.

Психологія, зокрема, допомагає зрозуміти, як емоції, мотивації та когнітивні процеси впливають на створення і споживання контенту в соціальних мережах.

Психологічні дослідження можуть виявити, які типи контенту найбільше резонують з аудиторією, які емоції викликають найсильнішу реакцію та як сприйняття інформації може змінюватися залежно від контексту. Наприклад, дослідження можуть показати, що позитивний контент має більшу схильність до поширення, ніж негативний, або що певні візуальні елементи можуть підсилювати емоційний відгук.

Соціологія, у свою чергу, надає інструменти для аналізу соціальних структур, відносин та динаміки групової поведінки в контексті соціальних мереж. Соціологічні теорії та методи дозволяють досліджувати, як соціальні мережі впливають на формування громадської думки, як відбувається поширення інформації між різними соціальними групами та як соціальні норми і цінності формуються та транслиуються через цифрові медіа. Соціологи можуть, наприклад, досліджувати вплив інфлюенсерів на поведінку їхніх підписників або вивчати, як соціальні рухи використовують соціальні мережі для мобілізації та координації дій.

Лінгвістика додає до цього аналізу глибоке розуміння мовних аспектів комунікації в соціальних мережах. Лінгвістичні дослідження фокусуються на аналізі текстових даних, виявленні мовних патернів, семантичному аналізі, а також на вивченні діалектів і жаргонів, що використовуються в різних онлайн-спільнотах. Лінгвісти можуть визначати, як користувачі використовують мову для вираження своєї ідентичності, як змінюється мова під впливом нових технологій та які мовні стратегії є найбільш ефективними для залучення аудиторії. Наприклад, аналіз хештегів може виявити основні теми та тренди, що цікавлять користувачів, або дослідження коментарів можуть показати, як мова використовується для побудови соціальної підтримки або конфлікту.

Інтеграція досліджень контенту соціальних мереж з цими науковими дисциплінами дозволяє створити більш цілісну картину взаємодії у цифровому середовищі. Вона сприяє розвитку нових методологій та інструментів, що поєднують кількісний і якісний аналіз, забезпечуючи більш точні та комплексні інсайти. Крім того, така міждисциплінарна співпраця стимулює інновації та

дозволяє розробляти більш адаптивні і ефективні стратегії для досліджень та практичної діяльності в області соціальних медіа.

Розвиток соціальних мереж в останні роки призвів до появи нових викликів та можливостей, що впливають на спосіб взаємодії користувачів, створення і споживання контенту, а також на загальне функціонування цифрових платформ. Одним з головних викликів є зростання популярності відеоконтенту. Відео стало провідним форматом для багатьох соціальних мереж, таких як YouTube, TikTok, Instagram Reels та Facebook Watch. Це вимагає від користувачів і брендів адаптації до нових форматів, що включають короткі відео, прямі трансляції і влогінг. Відеоконтент має здатність привертати більше уваги і залучення, але також вимагає більше ресурсів на створення і редагування.

Поява нових платформ також створює нові можливості та виклики. Нові соціальні мережі, такі як Clubhouse, що спеціалізується на аудіоконтенті, або Threads від Instagram, що пропонує нові способи обміну короткими повідомленнями, змінюють ландшафт цифрових комунікацій. Вони пропонують унікальні форми взаємодії, які вимагають від користувачів адаптації та освоєння нових навичок. Це також створює виклики для маркетологів, які повинні розробляти нові стратегії для залучення аудиторії на цих платформах.

Конфіденційність та безпека даних стають ще одним важливим викликом у розвитку соціальних мереж. Зростаюча кількість користувачів та обсяг даних, що генеруються, підвищує ризики витоків інформації та кібератак. Це вимагає від платформ посилення заходів безпеки та розробки нових політик конфіденційності. Також користувачі стають більш обізнаними та вимогливими щодо захисту своїх персональних даних, що змушує компанії бути більш прозорими та відповідальними.

Автоматизація та використання штучного інтелекту (ШІ) відкривають нові можливості для аналізу та управління соціальними мережами. ШІ може допомогти виявляти тренди, аналізувати поведінку користувачів, автоматизувати взаємодію та

покращувати таргетинг реклами. Однак це також викликає питання щодо етики використання ШІ, включаючи проблеми з упередженістю алгоритмів та їх вплив на приватність користувачів.

Зміни в регулюванні та законодавстві також впливають на розвиток соціальних мереж. Уряди різних країн вводять нові закони та правила, що регулюють діяльність цифрових платформ, зокрема щодо контенту, реклами та захисту даних. Це створює додаткові виклики для компаній, які повинні адаптуватися до різних юридичних вимог у різних юрисдикціях.

Загалом, розвиток соціальних мереж продовжує створювати як нові можливості, так і виклики. Зростання популярності відеоконтенту, поява нових платформ, питання конфіденційності та безпеки, використання ШІ та зміни в законодавстві вимагають постійного моніторингу та адаптації. Водночас ці зміни відкривають нові горизонти для взаємодії, маркетингу та комунікації, що робить соціальні мережі важливим інструментом у сучасному цифровому світі.

2 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРОБКИ

У сучасному світі інформаційних технологій розробка програмного забезпечення є невід'ємною частиною багатьох галузей і сфер діяльності. Від створення простих додатків до розробки складних систем і платформ, процес розробки програмного забезпечення вимагає використання широкого спектру інструментів та технологій. Розділ присвячений вивченню цих інструментів, їх функціональних можливостей, а також їхньому впливу на ефективність і якість кінцевого продукту.

2.1 Мова програмування Python

На сьогоднішній день сфера інформаційних технологій (ІТ) стрімко розвивається. І хто володіє мовами програмування, той перед собою відчиняє двері у світле майбутнє ІТ-індустрії.

Існує багато мов програмування, які по-своєму особливі. Розглянемо мову програмування Python, як одну з найбільш поширених та простих мов програмування. Перш за все, це пов'язано з її зрозумілістю та легкістю освоєння. Багато, хто розпочинає вивчати програмування саме з цієї мови. З її допомогою можна створювати зображення, обробляти різні дані, розробляти вебсайти і додатки з графічним інтерфейсом, працювати з базами даних. Ще однією перевагою Python є кросплатформеність мови, що дозволяє створювати програми, які будуть працювати на всіх операційних системах.

У рейтингу популярності всіх мов програмування компанії ТЮВЕ станом на березень 2021 року Python займає третє місце. Така популярність спричинена різноманітністю навчальних ресурсів: посібників з Python, вебресурсів, курсів та

книг. Найкращий спосіб здобути базові знання з Python – вибрати навчальний курс або урок та детально вивчити його опрацювати.

Python був створений Гвідо ван Россумом, нідерландським програмістом, у кінці 1980-х років і вперше випущений у 1991 році. Історія створення Python починається з його роботи в дослідницькому інституті CWI (Centrum Wiskunde & Informatica) в Нідерландах, де він працював над проектом мови програмування ABC.

ABC була мовою, розробленою для навчання програмуванню, однак вона мала деякі обмеження, які Гвідо ван Россум хотів усунути. Він вирішив створити нову мову, яка поєднувала б найкращі риси ABC з новими ідеями. Гвідо прагнув створити мову, яка була б простою у використанні, читабельною та універсальною.

На початку 1990-х років Гвідо ван Россум розпочав роботу над Python під час різдвяних канікул. Він обрав назву "Python" на честь британського комедійного шоу "Летючий цирк Монті Пайтона", яке він дуже любив. Гвідо прагнув створити мову, яка була б легкою для читання та писання, і яка б мала чіткий, виразний синтаксис.

Python 1.0 був офіційно випущений у лютому 1991 року. У перших версіях Python були реалізовані базові концепції, такі як об'єктно-орієнтоване програмування, обробка виключень, модулі та інтерактивний інтерпретатор. Згодом мова Python почала активно розвиватися, і в 2000 році була випущена версія Python 2.0, яка включала багато нових можливостей, таких як повна підтримка Unicode, покращена система збору сміття та нові вбудовані структури даних.

У 2008 році була випущена версія Python 3.0, яка не була зворотно сумісною з попередніми версіями, але внесла багато покращень та нових можливостей, таких як покращена підтримка міжнародних мов, новий синтаксис для обробки тексту та багато іншого. Python 3.x продовжує активно розроблятися і використовується у багатьох галузях, включаючи веб-розробку, науку про дані, машинне навчання та автоматизацію.

Python став однією з найпопулярніших мов програмування у світі завдяки своїй простоті, гнучкості та великій спільноті розробників, яка постійно розширює можливості мови та створює нові бібліотеки та інструменти.

Python є однією з найпопулярніших мов програмування завдяки своїй простоті, читабельності та потужним можливостям. Вона забезпечує широкий спектр бібліотек і фреймворків для розробки різноманітних додатків, включаючи телеграм ботів, роботу з API, обробку даних та візуалізацію результатів.

Python вирізняється зрозумілим і простим синтаксисом, що робить його чудовим інструментом для швидкої розробки та створення прототипів. Навіть найскладніші алгоритми можна реалізувати з мінімальними витратами часу та зусиль. Саме через цю особливість Python часто обирають як першу мову програмування для початківців, оскільки він дає можливість зосередитися на логіці створення програми, а не на подоланні труднощів зі складною структурою коду.

Python пропонує багатий арсенал вбудованих і сторонніх бібліотек, які значно спрощують розробку навіть найскладніших додатків. Наприклад, бібліотеки такі як NumPy та SciPy широко використовуються для наукових обчислень, а бібліотека Django - для веб-розробки.

Завдяки активній спільноті розробників, Python має чудову підтримку та великий вибір ресурсів для навчання і вирішення проблем. Це включає чисельні форуми, блоги, книги, відео-уроки та інші навчальні матеріали. Велика спільнота також сприяє швидкому розвитку та вдосконаленню мови, а також створенню нових бібліотек та інструментів.

Python підтримує модулі та пакети модулів, що дозволяє розробникам організовувати код у логічні частини. Це сприяє модульності та повторному використанню коду, оскільки кожен модуль може містити функції, класи та змінні, які можна використовувати в інших частинах програми або навіть в інших проектах. Це дозволяє значно зменшити кількість дубльованого коду та полегшує його підтримку.

Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах, таких як Windows, macOS та Linux. Це означає, що додатки, написані на Python, можуть легко виконуватися на будь-якій з цих платформ без необхідності значних змін у коді. Крім того, стандартні бібліотеки Python містять велику кількість готових до використання

модулів та функцій, що дозволяють виконувати широкий спектр завдань, від роботи з файлами до мережових операцій та взаємодії з базами даних.

Програмна мова Python підтримує кілька парадигм програмування, включаючи об'єктно-орієнтовану, процедурну, аспектно-орієнтовану та функціональну.

Об'єктно-орієнтоване програмування дозволяє створювати програми, засновані на об'єктах, які є інстанціями класів. Це сприяє більш природному моделюванню реальних об'єктів та їх взаємодій.

Процедурне програмування дозволяє писати програми як послідовність інструкцій або процедур, що робить код зрозумілим та легким для читання.

Аспектно-орієнтоване програмування допомагає розділити програми на окремі аспекти, зменшуючи зв'язність коду та підвищуючи його модульність.

Функціональне програмування дозволяє створювати програми, засновані на функціях, що приймають вхідні дані та повертають результати, не змінюючи стану програми. Це сприяє створенню чистого, передбачуваного коду, який легше тестувати та відлагоджувати.

Python - розповсюджена високорівнева мова загального призначення з акцентом на продуктивність розробника [17].

Python є надійним і універсальним інструментом для аналізу даних. Його простота, гнучкість і велика спільнота роблять його ідеальним вибором як для новачків, так і для досвідчених фахівців. Завдяки різноманітним бібліотекам і фреймворкам, Python дозволяє вирішувати широкий спектр завдань, від простої візуалізації даних до створення складних моделей машинного навчання.

2.2 PyCharm

У світі програмування існує безліч інструментів, які допомагають розробникам впоратися зі складними завданнями та підвищити продуктивність.

Однак є один особливий «інструмент», який вирізняється з-поміж інших. PyCharm - це кросплатформенне, розумне інтегроване середовище розробки. Воно надає все необхідне для комфортної роботи на Python, від автозавершення до потужних відладчиків та інструментів рефакторингу.

PyCharm був створений компанією JetBrains – відомим міжнародним розробником інструментів для програмування, що підтримують різні мови, такі як Java, Kotlin, C#, F#, C++, Ruby, Python, PHP, JavaScript та багато інших. Раніше відома як IntelliJ, компанія JetBrains сьогодні пропонує широкий спектр засобів для командної роботи та підтримки розробки провідних мов програмування.

PyCharm надає користувачу інтегроване середовище розробки (ICP; *англ. IDE — Integrated Development Environment*) із такими корисними функціями, як підсвічування синтаксису та помилок, навігація серед проєктів і в коді програми, відображення файлової структури проєкту, швидкий перехід між файлами, класами, методами тощо [18, с. 24].

Програму PyCharm можна завантажити з офіційного сайту. [19]

PyCharm надає розробникам безліч ключових функцій, що значно спрощують процес програмування та підвищують ефективність роботи. Однією з таких функцій є потужний налагоджувач, який дозволяє знаходити та виправляти помилки в коді. Використовуючи точки зупинки та аналізуючи значення змінних, розробники можуть швидко і точно визначати проблеми та усувати їх.

Інструменти рефакторингу, доступні в PyCharm, допомагають змінювати структуру коду без втрати його функціональності. Це включає автоматичне перейменування змінних, функцій і класів, а також реорганізацію коду для покращення його читабельності та підтримуваності.

PyCharm також інтегрується з популярними системами контролю версій, такими як Git та Mercurial. Це дозволяє розробникам зручно працювати з історією змін, відстежувати зміни в коді та співпрацювати з іншими членами команди. Інтеграція з системами контролю версій забезпечує ефективне управління версіями проєкту та спрощує процес злиття змін.

Інтелектуальне автодоповнення коду, яке пропонує PyCharm, значно прискорює процес написання коду і знижує ймовірність помилок. Ця функція надає розробникам пропозиції щодо завершення коду на основі контексту, що дозволяє швидко вводити складні конструкції та зменшує кількість синтаксичних помилок.

Статичний аналіз коду в PyCharm дозволяє виявляти потенційні помилки та підтримувати високу якість коду. Інспектування коду автоматично перевіряє його на наявність помилок, попереджає про можливі проблеми та пропонує рекомендації щодо їх усунення. Це допомагає розробникам підтримувати кодову базу в належному стані та запобігає появі критичних помилок.

Крім того, PyCharm надає зручний інтерфейс для роботи з віртуальними оточеннями Python. Ця функція дозволяє розробникам створювати та керувати різними оточеннями, що ізолює залежності та бібліотеки для кожного проекту. Такий підхід полегшує управління пакетами та забезпечує, що кожен проект використовує тільки ті залежності, які необхідні для його роботи, що зменшує ризик конфліктів та забезпечує стабільність розробки.

PyCharm надає розширене автозаповнення коду, яке забезпечує інтелектуальні підказки та автозавершення. Це допомагає розробникам швидше писати код і зменшувати кількість помилок. Автозаповнення особливо корисне для новачків, які тільки починають працювати з Python, оскільки воно полегшує процес знаходження необхідних функцій, методів та класів. Крім того, ця функція значно прискорює процес розробки, дозволяючи швидко знаходити та використовувати відповідні елементи коду.

PyCharm має потужний інструментарій для відлагодження, який дозволяє ефективно знаходити та виправляти помилки в коді. Відлагоджувач підтримує встановлення брейкпоінтів, перегляд значень змінних у реальному часі, виконання коду крок за кроком та інші корисні функції. Це дозволяє розробникам відстежувати потік виконання програми та виявляти причини помилок, що суттєво підвищує ефективність процесу відлагодження.

PyCharm також підтримує інтеграцію з системами контролю версій, такими як Git та SVN, що значно полегшує роботу в команді. Це дозволяє ефективно

керувати версіями коду, робити коміти, зливати зміни та вирішувати конфлікти. Інтеграція з Git також забезпечує можливість автоматичного відстеження змін у коді та проведення ревізій, що сприяє кращій організації та контролю за змінами в проекті.

PyCharm включає вбудовану підтримку тестування, що дозволяє легко писати та запускати тести за допомогою різних фреймворків, таких як `pytest` та `unittest`. Це важливо для забезпечення якості коду та запобігання помилкам у майбутньому. PyCharm також надає інструменти для візуалізації результатів тестування, що дозволяє швидко виявляти та виправляти проблеми. Ця функціональність сприяє більш ефективному тестуванню та підвищенню якості програмного забезпечення.

PyCharm доступний у двох версіях: Community Edition, яка є безкоштовною та відкритою, і Professional Edition, яка пропонує розширені можливості для комерційного використання. Це робить PyCharm доступним для широкого кола розробників, від новачків до професіоналів. Розглянуто різницю між двома версіями PyCharm, порівняльна характеристика Community Edition і Professional Edition наведена у Таблиці 2.1, що дозволяє детально оцінити функціональні можливості та відмінності між цими двома редакціями, включаючи підтримку різних фреймворків, інтеграцію з системами контролю версій та інші аспекти, важливі для розробників.

Таблиця 2.1. Порівняльна характеристика Community Edition і Professional Edition

Характеристика	PyCharm Community Edition	PyCharm Professional Edition
Вартість	Безкоштовно	Платна ліцензія з певною ціною
Для кого	Програмісти-початківці, учні, хобі-проекти	Професійні розробники, комерційні проекти
Підтримка Python	Повна підтримка Python	Повна підтримка Python
Засоби налагодження	Обмежений функціонал, включно з базовим налагоджувачем	Повноцінний відладчик із просунутими можливостями
Підтримка фреймворків	Обмежена підтримка Django, Flask та інших фреймворків	Повна підтримка Django, Flask, Pyramid та інших фреймворків

Продовження Таблиці 2.1

Web Development	Обмежена підтримка HTML, CSS, JavaScript	Повноцінна підтримка HTML, CSS, JavaScript, і TypeScript
Візуальний редактор форм	Не доступний	Доступний
Інтеграція з базами даних	Обмежена підтримка SQLite	Повна підтримка SQLite, PostgreSQL, MySQL та інших баз даних
Комерційні інструменти	Не доступні	Інструменти для розробки веб-додатків, SQL, та інші функції
Плагіни та розширення	Обмежена підтримка плагінів	Повна підтримка плагінів і розширень
Оновлення та підтримка	Обмежена підтримка та безкоштовні оновлення	Повна підтримка і доступ до оновлень протягом року

PyCharm Community Edition - це безкоштовна версія, яка ідеально підходить для початківців та студентів. Вона забезпечує всі необхідні інструменти для ефективної роботи з Python, включаючи інтелектуальне автодоповнення коду, налагодження, рефакторинг та інтеграцію з системами контролю версій. Community Edition також підтримує широкий спектр фреймворків, таких як Django і Flask. Однак, вона має обмежений набір функцій порівняно з Professional Edition.

PyCharm Professional Edition - це платна версія, яка розроблена для професійних розробників, які працюють над великими проектами. Крім всіх функцій, доступних у Community Edition, Professional Edition пропонує додаткові можливості.

Обидві версії PyCharm є потужними інструментами для розробки на Python. Вибір між ними залежить від індивідуальних потреб та цілей.

2.2 Бібліотеки Python

Бібліотеки Python є невід'ємною частиною екосистеми цієї мови програмування, забезпечуючи розробникам широкий спектр інструментів та

функцій для вирішення різноманітних завдань. Ці бібліотеки значно розширюють можливості стандартної бібліотеки Python, дозволяючи легко виконувати складні операції, які б інакше вимагали значних зусиль та часу. Вони охоплюють практично всі аспекти програмування, від обробки даних і наукових обчислень до веб-розробки та машинного навчання.

Однією з ключових переваг використання бібліотек Python є їхня простота та зручність у використанні. Більшість бібліотек мають добре задокументовані API, що дозволяє швидко освоїти їх навіть новачкам. Крім того, велика спільнота користувачів Python активно розробляє та підтримує ці бібліотеки, що забезпечує наявність безлічі прикладів, навчальних матеріалів та форумів для обговорень і вирішення проблем.

Бібліотеки Python — це колекції попередньо написаного коду, який допомагає виконувати певні завдання без необхідності створювати їх з нуля. Кожна бібліотека розроблена для конкретних завдань і надає зручний інтерфейс для доступу до її функцій [20].

Одні бібліотеки зосереджені на математичних операціях і обчисленнях, забезпечуючи функції для роботи з лінійною алгеброю, статистикою та обробкою великих масивів чисел. Вони дозволяють виконувати складні математичні обчислення з високою ефективністю. Такі бібліотеки є основою для роботи з великими обсягами даних і обчисленням моделей машинного навчання. Нижче наведені основні з них:

NumPy. Ця бібліотека є фундаментом для наукових обчислень у Python. Вона пропонує функціонал для роботи з багатовимірними масивами (numpy arrays) та дозволяє виконувати векторизовані операції, що значно підвищує швидкість обчислень. NumPy також підтримує функції для роботи з лінійною алгеброю, статистикою та генерацією випадкових чисел.

SciPy. Бібліотека розширює можливості NumPy, додаючи функції для інтеграції, оптимізації, статистичних аналізів та роботи з лінійною алгеброю. SciPy використовується там, де потрібні більш складні обчислення.

SymPy. Це бібліотека для символьної математики. Вона дозволяє виконувати алгебраїчні операції, вирішувати рівняння, розраховувати похідні, інтеграли та інші математичні вирази в символьній формі.

Інші бібліотеки спеціалізуються на обробці тексту та природної мови. Вони дозволяють аналізувати текстові дані, проводити токенізацію, синтаксичний розбір та інші завдання, важливі для розуміння людської мови. Це дуже корисно для створення чат-ботів, пошукових систем або автоматичного перекладу. Python пропонує широкий спектр бібліотек, які використовуються в різних застосунках, таких як створення чат-ботів, пошукових систем, автоматичний переклад і текстова аналітика. Ось ключові бібліотеки:

NLTK (Natural Language Toolkit) — одна з найстаріших і найпотужніших бібліотек для обробки природної мови. Вона пропонує інструменти для:

- Токенізації (розбиття тексту на слова чи речення).
- Синтаксичного та морфологічного аналізу.
- Виділення частин мови (POS-tagging).
- Створення частотних словників і роботи з n-грамами.
- Аналізу настроїв, класифікації текстів і роботи з корпусами (зокрема, готові корпуси для навчання).

SpaCy – це сучасна і швидка бібліотека, яка використовується для обробки великих обсягів тексту. Основні функції:

- Розбір речень і токенізація.
- Визначення частин мови (POS-tagging) і залежностей.
- Витяг іменованих сутностей (NER, наприклад, імена, дати, локації).
- Інтеграція з моделями глибинного навчання.
- Підтримка багатьох мов і адаптація моделей.

Gensim – це бібліотека для роботи з тематичним моделюванням та векторизацією текстів. Основні можливості:

- Обробка текстових даних великих обсягів.
- Побудова тематичних моделей (LDA, LSI).
- Генерація векторних представлень слів (Word2Vec).

Деякі бібліотеки орієнтовані на розробку веб-додатків, надаючи зручні засоби для створення серверної частини веб-сайтів, маршрутизації, обробки запитів і взаємодії з базами даних. Вони дозволяють будувати масштабовані системи з мінімальними зусиллями. Ось основні з них:

Django — це потужний і повнофункціональний фреймворк для веб-розробки, що слідує принципу "батареї включені" [21, с. 147]. Він включає всі необхідні інструменти для створення складних веб-додатків:

- Інтеграція з базами даних через ORM (Object-Relational Mapping).
- Зручна система маршрутизації.
- Адміністративна панель для управління даними.
- Захист від найпоширеніших веб-атак (наприклад, CSRF, SQL-ін'єкції).
- Підходить для великих проєктів, таких як платформи електронної комерції чи соціальні мережі.

Flask — мінімалістичний і легкий фреймворк, який надає розробникам більше свободи. Особливості:

- Підходить для створення невеликих або середніх веб-додатків.
- Висока гнучкість у налаштуванні.
- Підтримка розширень для інтеграції з базами даних, автентифікації, роботи з API тощо.
- Простий і зручний у використанні для стартапів та невеликих команд.

FastAPI — сучасний фреймворк для створення веб-додатків і API. Особливості:

- Використання асинхронного програмування для досягнення високої продуктивності.
- Автоматична генерація документації API (Swagger і ReDoc).
- Інтеграція з Pydantic для валідації даних.
- Ідеально підходить для створення RESTful API та мікросервісів.

Також існують бібліотеки для візуалізації даних, які допомагають створювати графіки, діаграми та інші форми представлення інформації. Вони

використовуються для аналізу даних, презентацій або створення інтерактивних інформаційних панелей.

Інша важлива категорія бібліотек дозволяє працювати з файлами різних форматів, включаючи зображення, відео, аудіо та документи. Вони потрібні для автоматизації завдань, таких як обробка медіа-файлів або генерація звітів.

Всі ці бібліотеки активно підтримуються спільнотою та оновлюються для забезпечення сумісності з новими версіями Python. Python надає інтегрований менеджер пакетів `pip`, який спрощує встановлення та оновлення бібліотек.

Бібліотека **`python-telegram-bot`** є одним з найпопулярніших інструментів для розробки ботів у месенджері Telegram за допомогою мови програмування Python. Вона забезпечує зручний та потужний інтерфейс для взаємодії з Telegram Bot API, дозволяючи створювати ботів з різними функціональними можливостями, від простих чат-ботів до складних автоматизованих систем.

Однією з основних переваг бібліотеки `python-telegram-bot` є її простота у використанні та добре структурована документація, яка допомагає розробникам швидко освоїтися та почати розробку. Бібліотека підтримує всі основні функції Telegram Bot API, включаючи відправку та отримання повідомлень, обробку медіафайлів, створення клавіатур, проведення опитувань та багато іншого.

Простий інтерфейс легко налаштовується та використовує методи для роботи з повідомленнями, командами та іншими функціями Telegram. Це дозволяє створювати боти з різноманітними функціями, такими як відповіді на команди користувачів, обробка повідомлень, відправка мультимедійних файлів тощо. Бібліотека забезпечує зручний спосіб роботи з основними компонентами Telegram API, такими як чат-боти, канали та групи.

Бібліотека підтримує асинхронні виклики, що дозволяє ефективно обробляти велику кількість запитів. Це особливо корисно для ботів, які обслуговують багато користувачів одночасно. Використання асинхронного програмування допомагає знизити затримки та підвищити продуктивність бота, забезпечуючи швидке реагування на запити користувачів.

Можливість створення складних логік обробки повідомлень та інтеграції з іншими сервісами. Наприклад, можна налаштувати бот для взаємодії з іншими API, базами даних, аналітичними інструментами тощо. Це дозволяє створювати багатофункціональні боти, які можуть виконувати різноманітні завдання, від простих відповідей на запити до складних інтеграцій з зовнішніми системами.

Основна структура бібліотеки базується на концепції диспетчера оновлень (Updater) та обробників (Handlers). Диспетчер оновлень відповідає за отримання нових повідомлень від Telegram та передачу їх відповідним обробникам. Обробники, у свою чергу, визначають, як саме слід обробляти певні типи повідомлень. Наприклад, обробник повідомлень (MessageHandler) може бути налаштований для реагування на текстові повідомлення, тоді як обробник команд (CommandHandler) обробляє команди, що починаються зі слешу (/).

Бібліотека також підтримує роботу з асинхронними функціями, що дозволяє створювати більш ефективні та масштабовані боти. Використання асинхронного програмування дозволяє ботам обробляти велику кількість запитів одночасно, що є критично важливим для ботів з високим навантаженням.

Окрім цього, python-telegram-bot забезпечує зручні інструменти для роботи з вебхуками. Використання вебхуків дозволяє Telegram надсилати оновлення безпосередньо на сервер бота, що забезпечує швидшу обробку повідомлень у порівнянні з періодичним опитуванням сервера (polling).

Завдяки своїй гнучкості та широким можливостям, бібліотека python-telegram-bot є ідеальним вибором для розробників, які бажають створювати надійні та функціональні Telegram-боти. Вона підходить як для новачків, так і для досвідчених розробників, завдяки своїй простоті у використанні та потужному функціоналу.

Бібліотека **requests** є популярним інструментом для роботи з HTTP-запитами в Python. Вона забезпечує простий і інтуїтивно зрозумілий інтерфейс для взаємодії з API різних сервісів, включаючи соціальні мережі.

Основні можливості requests: легкий синтаксис для відправлення HTTP-запитів (GET, POST, PUT, DELETE) та обробки відповідей. Це дозволяє швидко

інтегрувати різні API та отримувати необхідні дані без складних налаштувань. Використання бібліотеки `requests` значно спрощує процес роботи з веб-сервісами та дозволяє зосередитися на логіці програми.

Вбудовані механізми для обробки помилок і перевірки статусу відповідей. Це допомагає розробникам легко обробляти ситуації, коли сервер повертає помилки або несподівані відповіді. Використання бібліотеки `requests` дозволяє автоматично обробляти різні типи HTTP-помилки та забезпечує надійність роботи програми.

Легке налаштування автентифікації для роботи з захищеними API. Бібліотека підтримує різні методи автентифікації, такі як Basic Auth, OAuth, що дозволяє безпечно взаємодіяти з сервісами. Використання бібліотеки `requests` забезпечує захищений доступ до конфіденційної інформації та дозволяє створювати безпечні додатки.

Бібліотека **pandas** є однією з найпопулярніших та найпотужніших бібліотек для обробки та аналізу даних у Python. Вона забезпечує високоефективні, гнучкі та виразні структури даних, що дозволяють легко маніпулювати табличними даними.

Основною структурою даних у `pandas` є `DataFrame`, яка являє собою двовимірну таблицю з даними, подібну до таблиці в базі даних або електронній таблиці. Кожен стовпець у `DataFrame` може мати різний тип даних (числа, рядки, дати тощо), що робить її надзвичайно універсальною.

Однією з ключових переваг `pandas` є її здатність легко імпортувати дані з різних джерел, включаючи файли CSV, Excel, бази даних SQL та інші формати. Це дозволяє швидко завантажувати дані для подальшого аналізу.

Бібліотека `pandas` також забезпечує великий набір інструментів для маніпуляції з даними. Наприклад, ви можете легко фільтрувати, сортувати, групувати та агрегувати дані.

Крім цього, `pandas` надає інструменти для роботи з відсутніми значеннями, що є важливим аспектом обробки даних. Ви можете легко виявити відсутні значення та замінити їх на інші значення або видалити відповідні рядки.

Ще одна важлива функція pandas — це можливість об'єднувати декілька DataFrame. Це дозволяє легко з'єднувати дані з різних джерел, використовуючи операції злиття (merge) та об'єднання (concatenate).

Бібліотека pandas також підтримує роботу з часовими рядами та датами, що є важливою функцією для фінансового аналізу та інших областей, де час є важливим фактором. Ви можете легко конвертувати рядки в дати, виконувати операції з датами та виконувати аналіз часових рядів.

Легке поєднання з іншими бібліотеками для візуалізації та машинного навчання, такими як Matplotlib, Seaborn, Scikit-learn. Це дозволяє створювати складні аналітичні додатки, які включають обробку даних, візуалізацію результатів та побудову моделей машинного навчання. Використання бібліотеки pandas у поєднанні з іншими інструментами забезпечує комплексний підхід до аналізу даних.

Таким чином, pandas є незамінним інструментом для будь-якого розробника, який працює з даними. Вона забезпечує потужні можливості для імпорту, обробки, маніпуляції та аналізу даних, що дозволяє швидко отримувати цінну інформацію з великих обсягів даних. Завдяки своїй гнучкості та широкому спектру функцій, pandas стала вибором номер один для аналітиків даних, науковців та інженерів з усього світу.

Matplotlib — це гнучка бібліотека для візуалізації даних у Python, яка надає розробникам можливість створювати широкий спектр графіків та діаграм [22, с. 116]. Вона є однією з найпопулярніших бібліотек для побудови візуалізацій завдяки своїй простоті використання та широкому функціоналу. Matplotlib дозволяє створювати статичні, анімовані та інтерактивні графіки, що робить її універсальним інструментом для роботи з даними.

Основною складовою Matplotlib є модуль pyplot, який забезпечує високорівневий інтерфейс для створення графіків [23]. Pyplot надає функції для побудови різних типів графіків, включаючи лінійні графіки, гістограми, точкові діаграми, стовпчикові діаграми, кругові діаграми та багато інших. Ці функції

дозволяють легко створювати графіки з мінімальними зусиллями, використовуючи прості команди.

Однією з ключових особливостей Matplotlib є її здатність налаштовувати графіки до найдрібніших деталей. Розробники можуть змінювати кольори, шрифти, розміри та розташування елементів графіку, додавати підписи до осей, заголовки, легенди та анотації. Це дозволяє створювати професійні та інформативні візуалізації, які легко сприймаються та аналізуються.

Matplotlib також підтримує роботу з декількома підграфіками в одному вікні, що дозволяє порівнювати різні набори даних на одному графіку. Це досягається за допомогою функції `subplot`, яка розділяє вікно на кілька частин, кожна з яких може містити власний графік. Такий підхід є особливо корисним для аналізу даних, коли потрібно представити декілька аспектів одночасно.

Крім того, Matplotlib інтегрується з іншими популярними бібліотеками для обробки даних, такими як NumPy та pandas. Це дозволяє легко завантажувати, обробляти та візуалізувати дані без необхідності перетворення між різними форматами. Наприклад, дані з DataFrame у pandas можуть бути безпосередньо передані у функції Matplotlib для створення графіків.

Ще однією важливою особливістю Matplotlib є підтримка інтерактивних графіків, які дозволяють користувачам взаємодіяти з візуалізаціями в реальному часі. Це включає можливість збільшення та зменшення масштабу, панорамування, а також вибір та виділення окремих елементів графіку. Інтерактивні можливості Matplotlib роблять її корисною для презентацій та аналізу даних, коли потрібно детально розглянути окремі частини графіку.

Matplotlib також підтримує збереження графіків у різних форматах, включаючи PNG, PDF, SVG та інші. Це дозволяє зберігати візуалізації для подальшого використання у звітах, презентаціях або публікаціях. Бібліотека забезпечує високу якість зображень, що робить її придатною для створення графіків для наукових статей та інших офіційних документів.

Наявна підтримка інтерактивних графіків для детального аналізу даних. Користувачі можуть взаємодіяти з графіками, змінювати масштаб, переглядати

додаткову інформацію, що робить аналіз даних більш зручним та ефективним. Використання бібліотеки Matplotlib дозволяє створювати інтерактивні візуалізації, які можуть бути використані у веб-додатках та інших інтерактивних середовищах.

Завдяки своїй широкій функціональності та гнучкості, Matplotlib є незамінним інструментом для науковців, інженерів, аналітиків даних та інших фахівців, які працюють з візуалізацією даних. Вона дозволяє легко створювати інформативні та естетично привабливі графіки, що допомагають краще розуміти та аналізувати дані.

На завершення, бібліотеки Python є ключовим елементом, що робить цю мову програмування надзвичайно потужною та універсальною. Вони надають інструменти для виконання широкого спектру завдань, від простих операцій з даними до складних наукових обчислень та машинного навчання. Завдяки великій кількості доступних бібліотек, розробники можуть швидко знаходити та використовувати готові рішення для своїх проектів, що значно підвищує продуктивність та ефективність розробки.

Активна спільнота Python постійно підтримує та розширює ці бібліотеки, забезпечуючи їх актуальність та високу якість. Це створює сприятливе середовище для навчання та розвитку нових технологій, роблячи Python одним з найпопулярніших виборів серед програмістів у всьому світі.

Використання бібліотек Python дозволяє розробникам зосередитися на вирішенні конкретних завдань, не витрачаючи час на реалізацію базових функцій з нуля. Це робить процес розробки більш ефективним та дозволяє створювати більш складні та інноваційні рішення. Бібліотеки Python продовжують залишатися важливим інструментом у скарбниці кожного розробника, сприяючи розвитку сучасних технологій та інновацій.

3 РОЗРОБКА ТЕЛЕГРАМ-БОТУ

У цьому розділі висвітлено процес створення телеграм-боту, який призначений для автоматизованого збору, обробки та аналізу контенту соціальних мереж. Розробка інструменту здійснюється з використанням сучасних методів програмування, доступних API Telegram, а також бібліотек і фреймворків для роботи з великими даними. Окрему увагу приділено архітектурі бота, його функціональним можливостям та процесу тестування.

3.1 Проектування архітектури телеграм-боту

Після вибору інструментів та технологій для реалізації проєкту можна розпочати його виконання. Перший крок — докладніше розібратися у процесі реєстрації бота. Для цього варто знайти @botfather у пошуку додатку і відкрити з ним діалог (рис. 3.1).

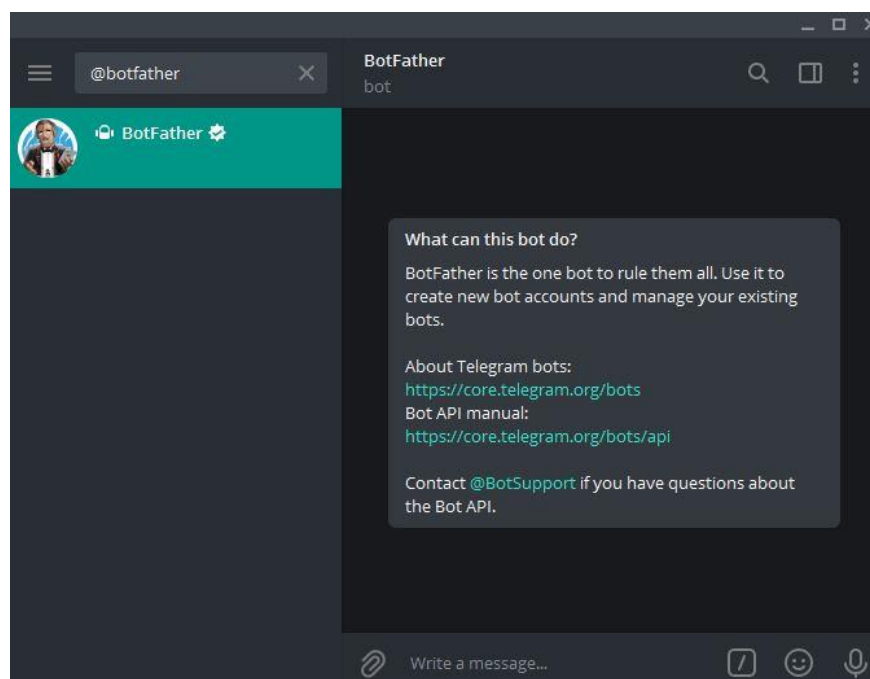


Рисунок 3.1 – Початок реєстрації бота у додатку Telegram

Скориставшись командою «/start», потрібно надіслати запит, щоб активувати штучний інтелект і отримати відповідь. Після обробки запиту система надсилає меню з доступними функціями (рис. 3.2).

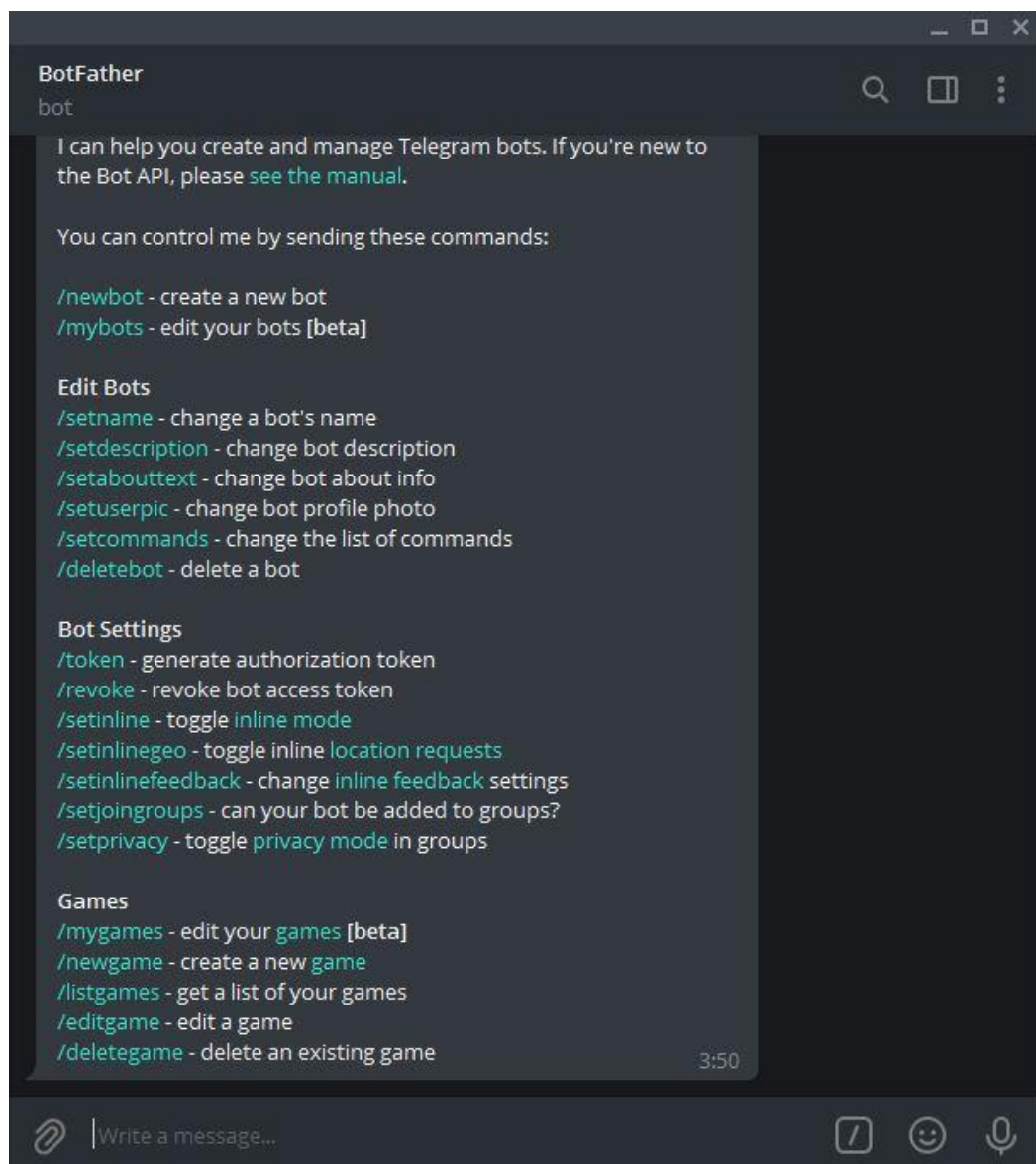


Рисунок 3.2 – Функції, що надає BotFather

Щоб розгорнути бота в додатку, потрібно виконати команду «/newbot».

Далі в чаті слід вказати назву бота. Хоча ця назва не впливає на роботу бота, вона повинна давати користувачам розуміння його призначення. Після цього Bot Father запропонує вибрати унікальне ім'я користувача для бота. Ім'я має складатися з латинських літер, цифр і нижнього підкреслення, а також обов'язково закінчуватися на "bot". Після завершення попередніх процесів в чат приходить

повідомлення про завершення реєстрації бота, в якому надається токен для доступу до HTTP API, а також посилання на бота в додатку телеграм (рис. 3.3).

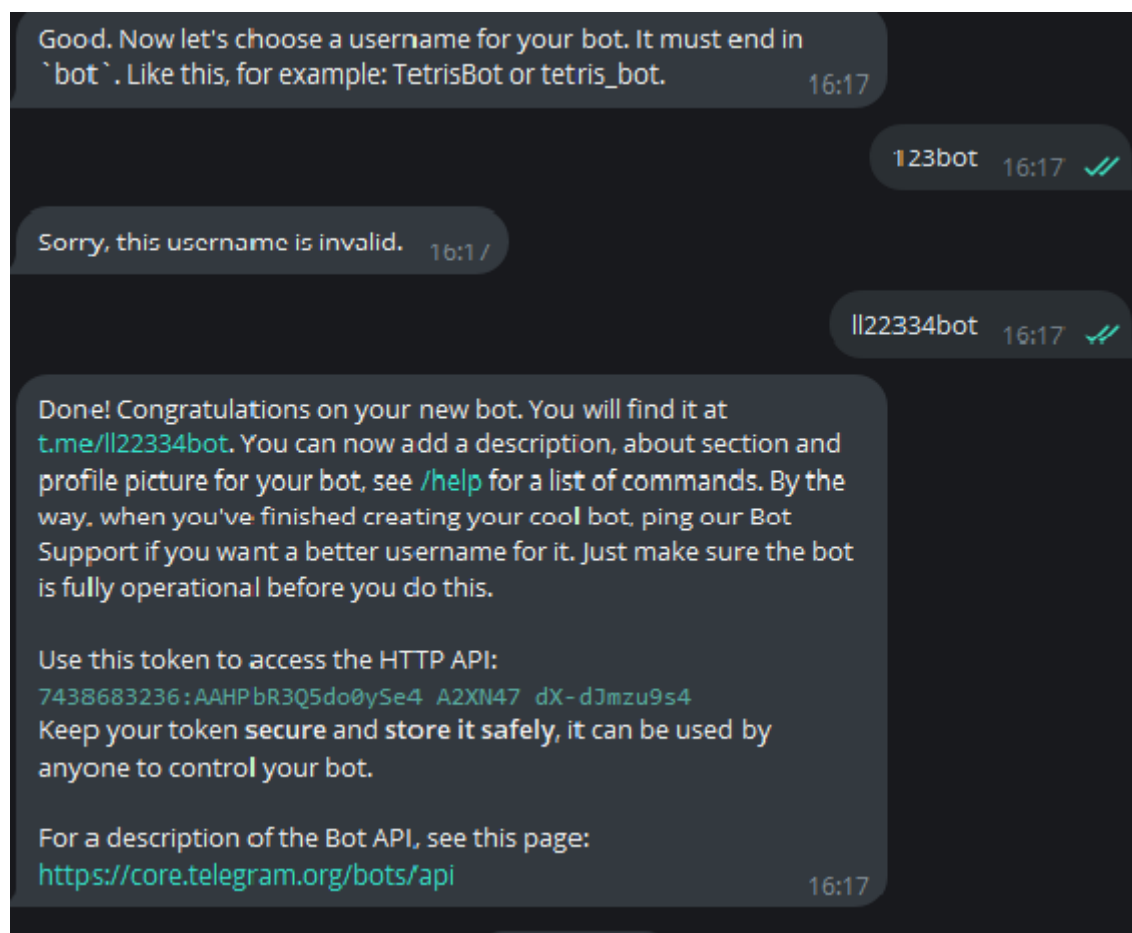


Рисунок 3.3 – Повідомлення з ключем доступу та посиланням на бота

Після цього можна перейти за посиланням і ознайомитися з ботом, який надалі буде доповнюватися командами та формувати свій функціонал. На цьому етапі він виглядає як порожній шаблон, відкриваючи простір для подальших експериментів (рис. 3.4). Натиснувши на аватар бота, можна переглянути інформацію про його профіль, аналогічно до профілю звичайного користувача. (рис. 3.5).

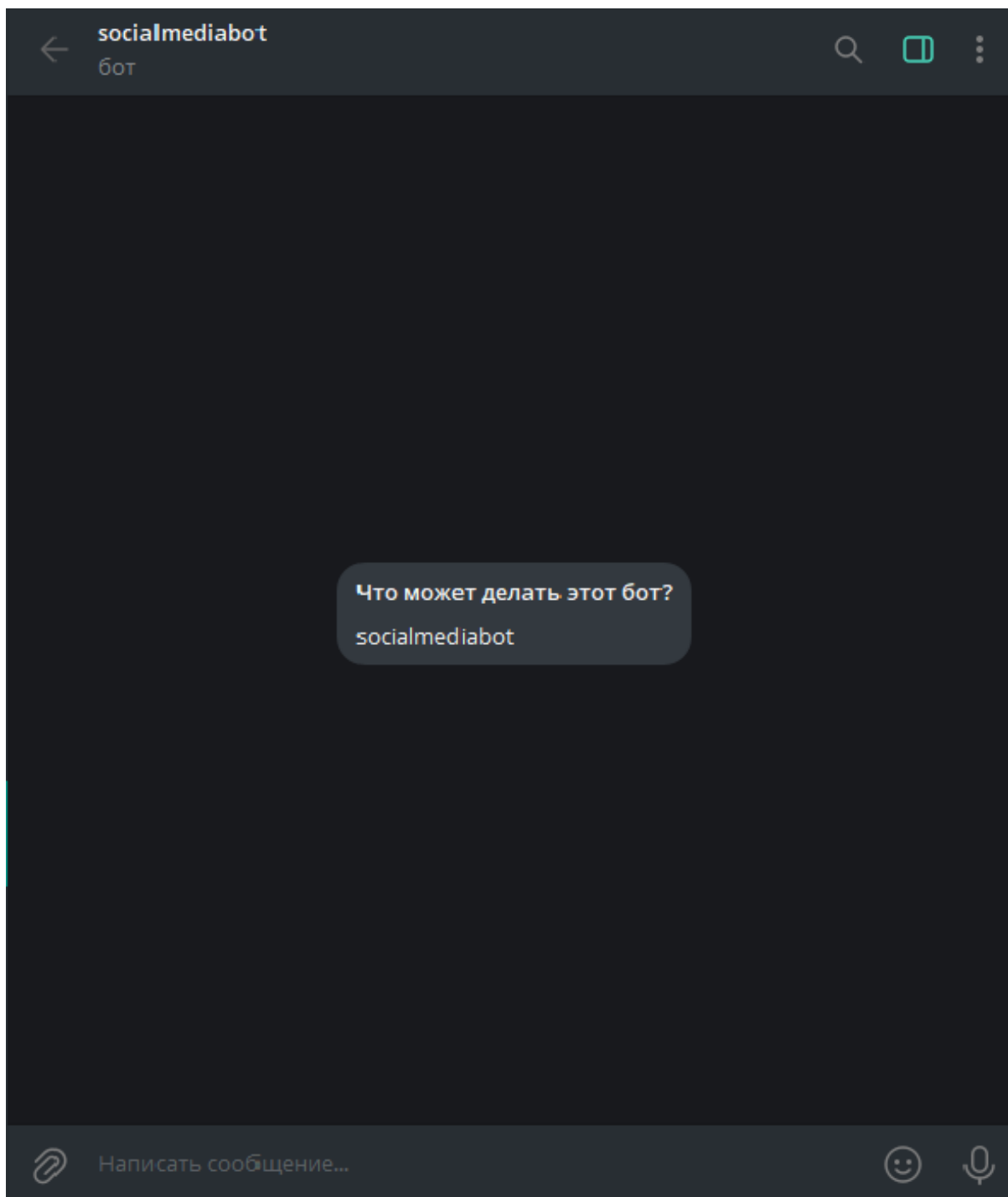


Рисунок 3.4 – Вікно чату з зареєстрованим ботом

Щоб почати роботу з ботом, потрібно виконати команду «/start». Однак наразі бот не реагуватиме на цей запит, оскільки ще не налаштований для обробки HTTP-запитів. У майбутньому ця функціональність буде додана через написання відповідного коду для управління ботом.

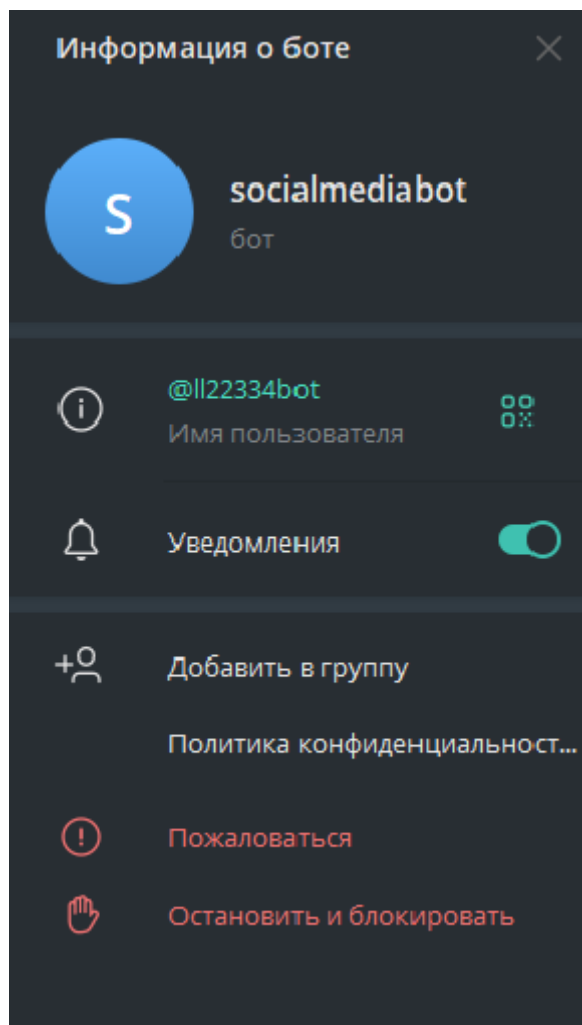


Рисунок 3.5 – Вікно додаткової інформації користувача

Якщо подивитися на цей інформаційний блок, можна помітити, що він зовні не відрізняється від звичайного чату з іншим користувачем (живою людиною). Це вікно чату виступає своєрідним чистим полотном для роботи. Увесь код, написаний у середовищі розробки, буде оброблятися та відображатися в чаті бота у вигляді текстових повідомлень, зображень, кнопок чи інших елементів інтерфейсу. Це фактично платформа для взаємодії з програмою.

При цьому шифрування даних і з'єднань не стосується стандартного протоколу Telegram. Відповіді на запити користувача передаються через протокол HTTPS. Формат таких запитів виглядає так:
https://api.telegram.org/bot'token'/назва_метода

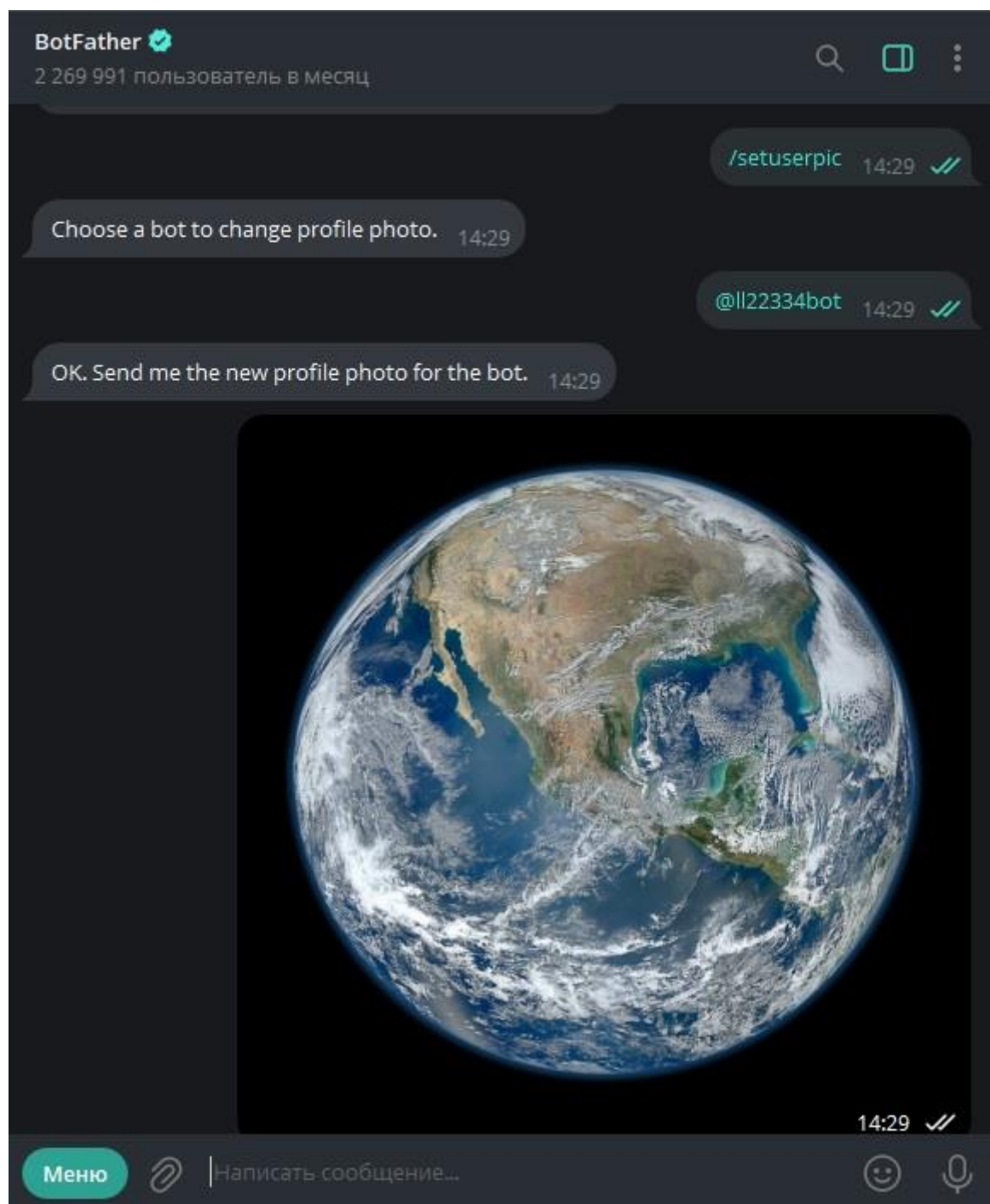


Рисунок 3.6 – Оновлення зображення для бота

Щоб оновити зображення профілю бота, потрібно зайти в Bot Father, виконати команду «/setuserpic» і надіслати в чат потрібне зображення (рис. 3.6). Після цього, відкривши чат із ботом, можна переконатися, що нове зображення профілю успішно встановлено (рис. 3.7). Візуальне оформлення бота та його грамотний дизайн можуть відіграти важливу роль у приверненні уваги користувачів, стимулюючи їх зацікавитися ботом і почати з ним спілкування.

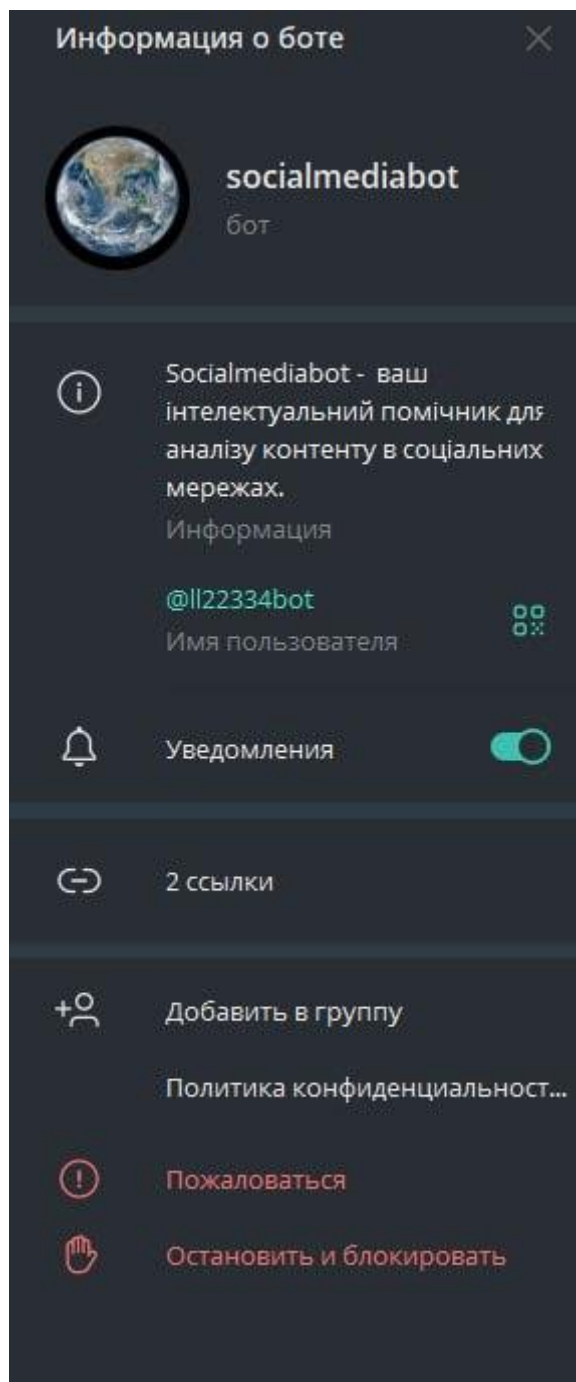


Рисунок 3.7 – Нове зображення та опис профіля бота

Наступним етапом є налаштування середовища розробки та підключення всіх необхідних ресурсів для зручної роботи. Спочатку потрібно встановити останню версію Python на комп'ютер. Завантажуючи інсталятор з офіційного сайту, слід звернути увагу на його сумісність із вашою операційною системою, а також на стабільність обраної версії. Найновішою та найбільш стабільною на сьогодні є версія 3.13.

Обравши стабільний реліз, встановлюємо оптимальну версію. Запустивши інсталятор, натискаємо кнопку *Upgrade Now* і чекаємо на завершення процесу встановлення (рис. 3.8).



Рисунок 3.8 – Встановлення оптимальної версії Python

Перш ніж розпочати програмування, необхідно налаштувати середовище розробки PyCharm. Для створення нового проекту слід виконати такі дії: перейти у меню "Файл" (File), вибрати пункт "Новий проект" (New Project) та вказати шлях до папки, де буде збережено всі файли проекту. Цей процес зображено на рисунку 3.9., обрано середовище серед запропонованих:

- virtualenv;
- pipenv;
- conda.

Середовище розробки для Python підтримує три типи віртуальних середовищ. Pipenv, як більш сучасна альтернатива virtualenv, вирішує проблему ручного керування залежностями проектів. Він автоматично генерує і оновлює файл requirements.txt, що дозволяє відтворювати середовище проекту в будь-який момент. Це спрощує співпрацю в команді та розгортання проектів.

“Conda” є чітко орієнтованим на використання Num Py та Sci Py пакетів, які призначені для роботи з масивами, функціями вищої математики та наукових, чи інженерних математичних розрахунків.

Pipenv - це найпростіший і найінтуїтивніший спосіб створювати та керувати віртуальними середовищами в Python.

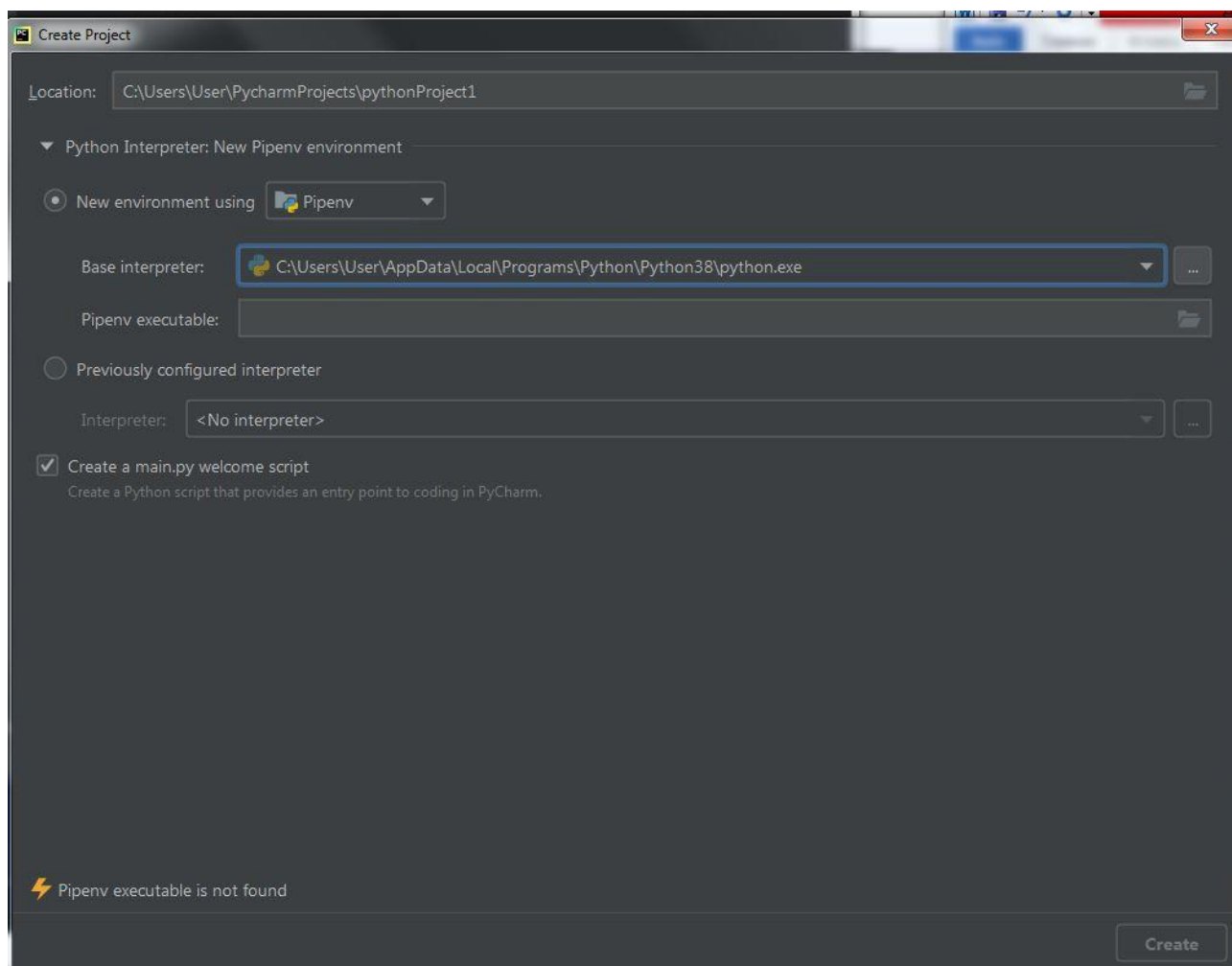


Рисунок 3.9 – Налаштування проекту в середовищі розробки

При ініціалізації нового проекту середовище розробки за замовчуванням асоціює його з останньою версією інтерпретатора Python, виявленою в системі. Це обумовлено попередньою інсталяцією Python.

Інтерпретатор Python є програмним забезпеченням, призначеним для виконання програм, написаних на мові Python. Він виконує лексичний аналіз, синтаксичний аналіз та семантичний аналіз коду, перетворюючи його в еквівалентну послідовність машинних інструкцій. Таким чином, інтерпретатор забезпечує виконання логіки програми.

Більшість сучасних програм написані на мовах високого рівня, які потребують перетворення в машинний код для виконання. Цей процес може здійснюватися за допомогою компілятора або інтерпретатора.

Компілятор перетворює весь вихідний код програми в машинний код одноразово перед виконанням. Це може бути неефективним для великих проектів, оскільки кожна зміна в коді вимагає повторної компіляції.

Інтерпретатор, в свою чергу, не потребує бачити всю програму, йому потрібен лише той код, що виконується зараз, що займає в цілому менше часу на виконання програми (рис. 3.10), принцип роботи було спроектовано в ресурсі для створення діаграм.



Рисунок 3.10 – Принцип роботи інтерпретатора

Pip - це система управління пакетами для Python, що забезпечує зручний механізм для встановлення та керування сторонніми модулями. Починаючи з версії Python 3.4, pip інтегрований в дистрибутив Python, що економить час на додаткову конфігурацію. Оскільки ви використовуєте Python 3.13, pip доступний для вас без додаткової установки.

Основні команди pip:

- `pip help` – допомога згідно доступних команд;
- `pip list` – список встановлених додатків;
- `pip search` – пошук пакетів за назвою;
- `pip install package_name` – встановлення бібліотек та пакетів;
- `pip uninstall package_name` – видалення бібліотек та пакетів;
- `pip show package_name` – надає інформацію про встановлену бібліотеку;
- `pip install -U` – оновлення обраних бібліотек;

На початку роботи було відкрито термінал середовища розробки.

Термінал, або консоль, є основним інструментом для взаємодії з операційною системою на низькому рівні. В середовищі розробки термінал забезпечує зручний інтерфейс для виконання команд і скриптів.

Для управління пакетами Python ми використовуємо пакетний менеджер `pip`. Команда `pip install "назва_пакета"` дозволяє встановлювати пакети з репозиторію PyPI. Варто зазначити, що середовище розробки, як правило, пропонує додаткові можливості для керування пакетами, такі як віртуальні середовища, які дозволяють ізолювати залежності різних проектів.

В середовищі розробки ми маємо зручний доступ до командного рядка, який називається термінал. Саме тут ми виконуємо різноманітні команди для управління нашим проектом. Однією з найпоширеніших команд є `pip install <назва_пакета>`, яка дозволяє встановлювати нові бібліотеки. Наприклад, для роботи з Telegram-ботами ми можемо встановити пакет `TelegramBotApi` за допомогою цієї команди. Детальніше цей процес проілюстровано на рисунку 3.11.

```

Terminal: Local X +
(pythonProject4) C:\Users\User\PycharmProjects\pythonProject4>pip install TelegramBotApi
Collecting TelegramBotApi
  Using cached TelegramBotAPI-0.3.2.tar.gz (7.1 kB)
Collecting pyOpenSSL
  Using cached pyOpenSSL-20.0.1-py2.py3-none-any.whl (54 kB)
Collecting service_identity
  Using cached service_identity-21.1.0-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: requests in e:\pythonproject4\lib\site-packages (from TelegramBotApi) (2.25.1)
Requirement already satisfied: six>=1.5.2 in e:\pythonproject4\lib\site-packages (from pyOpenSSL->TelegramBotApi) (1.16.0)
Collecting cryptography>=3.2
  Using cached cryptography-3.4.7-cp36-abi3-win_amd64.whl (1.6 MB)
Collecting cffi>=1.12
  Using cached cffi-1.14.5-cp38-cp38-win_amd64.whl (179 kB)

```

Рисунок 3.11 – Завантаження бібліотеки до проекту

`TelegramBotAPI` - це інструмент для розробки Telegram-ботів, який приховує складність взаємодії з API Telegram. Використовуючи цю бібліотеку, ви можете створювати ботів, які можуть відповідати на повідомлення, виконувати команди, працювати з базами даних і багато іншого. Замість того, щоб вручну формувати HTTP-запити, ви можете використовувати зручні функції, надані бібліотекою.

Існують два основні механізми для отримання оновлень від Telegram-бота: метод довгого опитування (`getUpdates`) та метод веб-хуків. При використанні методу `getUpdates` бот періодично надсилає запит на сервер Telegram, щоб перевірити наявність нових повідомлень. Метод веб-хуків передбачає налаштування веб-сервера, на який Telegram буде надсилати оновлення. Обидва методи мають свої переваги та недоліки, вибір яких залежить від конкретної задачі. Важливо зазначити, що Telegram зберігає оновлення обмежений час, тому їх необхідно обробляти своєчасно..

Об'єкт `Update` служить для структурування даних про взаємодію користувача з ботом. Він містить різні поля, які описують тип оновлення, його зміст (наприклад, текст повідомлення) та інші деталі. Кожне оновлення відповідає одній конкретній дії користувача, і лише одне поле в об'єкті `Update` буде заповнено відповідними даними.

Метод `getUpdates` використовує технологію довгого опитування для отримання оновлень від Telegram-сервера. Бот періодично надсилає запит на сервер, і якщо з'явилися нові повідомлення або інші події, сервер повертає масив об'єктів `Update`, кожен з яких описує одне оновлення.

Дані, що обробляються модулем, мають структуру JSON. JSON-об'єкти, що представляють різні елементи взаємодії з ботом, візуалізовані на рисунках 3.12-3.14.

User

Цей об'єкт зображає бота чи користувача Telegram.

Поле	Тип	Опис
<code>id</code>	<code>Integer</code>	Ідентифікатор користувача чи бота
<code>first_name</code>	<code>String</code>	Ім'я бота
<code>username</code>	<code>String</code>	Username користувача чи бота

Рисунок 3.12 – Об'єкт `User` та його команди

Об'єкт надає доступ до метаданих користувача або бота, включаючи його унікальний ідентифікатор та ім'я в контексті поточного чату. Всі текстові поля в цьому об'єкті повинні бути закодовані в UTF-8 для забезпечення підтримки різних мовних символів.

Chat

Об'єкт чат – інтерфейс зв'язку між ботом та користувачем.

Поле	Тип	Опис
id	Integer	Ідентифікатор чату. Абсолютне значення не перевищує 1e13
type	Enum	Тип чату: "private", "group", "supergroup" чи "channel"
title	String	Назва для каналу чи групи
first_name	String	Ім'я користувача в чаті
last_name	String	Прізвище користувача в чаті
all_members_are_administrators	Boolean	.True, якщо усі користувачі являються адміністраторами

Рисунок 3.13 – Об'єкт Chat та його команди

Об'єкт експонує набір методів для управління атрибутами чату, включаючи тип чату, опис, ідентифікатори учасників та інші параметри, що визначають інтерфейс взаємодії користувача з ботом.

Message

Об'єкт повідомлень у чаті.

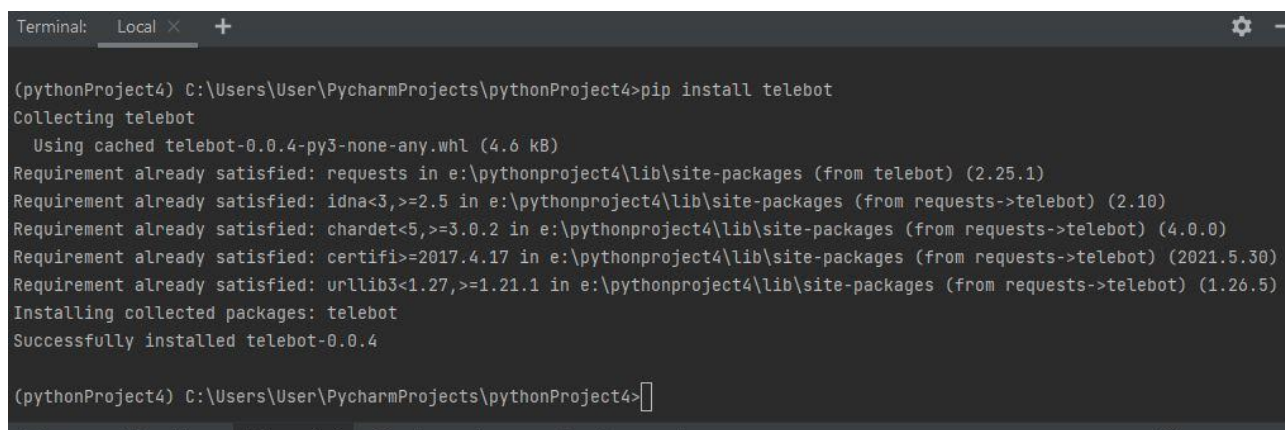
Поле	Тип	Опис
message_id	Integer	Ідентифікатор повідомлення
from	User	Відправник повідомлення
date	Integer	Дата відправки повідомлення
chat	Chat	Діалог в якому було відправлено повідомлення
forward_from	User	Хто є відправником оригінального повідомлення, у разі пересланих повідомлень
forward_date	Integer	Дата відправки оригінального повідомлення, у разі пересланих повідомлень
text	String	Для текстових повідомлень: текст повідомлення, 0-4096 символів
contact	Contact	Інформація про відправлений контакт
location	Location	Інформація про місцезнаходження
group_chat_created	True	Повідомлення про створення групового чату
pinned_message	Message	Вказане повідомлення було прикріплено до шапки чату

Рисунок 3.14 – Об'єкт Message та його команди

Список доступних бібліотек:

- requests;
- pandas;
- matplotlib;
- telebot;
- python-telegram-bot.

Серед усіх доступних бібліотек для розробки Telegram-ботів, telebot виділяється своєю простотою використання. Вона ідеально підходить для невеликих проектів, що не вимагають складних інтеграцій. Хоча всі бібліотеки для Telegram-ботів базуються на одному API, telebot пропонує найбільш зручний для розробника інтерфейс. Процес інсталяції telebot зображено на рисунку 3.15.



```

Terminal: Local x +
(pythonProject4) C:\Users\User\PycharmProjects\pythonProject4>pip install telebot
Collecting telebot
  Using cached telebot-0.0.4-py3-none-any.whl (4.6 kB)
Requirement already satisfied: requests in e:\pythonproject4\lib\site-packages (from telebot) (2.25.1)
Requirement already satisfied: idna<3,>=2.5 in e:\pythonproject4\lib\site-packages (from requests->telebot) (2.10)
Requirement already satisfied: chardet<5,>=3.0.2 in e:\pythonproject4\lib\site-packages (from requests->telebot) (4.0.0)
Requirement already satisfied: certifi>=2017.4.17 in e:\pythonproject4\lib\site-packages (from requests->telebot) (2021.5.30)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in e:\pythonproject4\lib\site-packages (from requests->telebot) (1.26.5)
Installing collected packages: telebot
Successfully installed telebot-0.0.4

(pythonProject4) C:\Users\User\PycharmProjects\pythonProject4>
  
```

Рисунок 3.15 – Встановлення бібліотеки у папку проекту

Бібліотека requests – це потужний інструмент для роботи з веб-сервісами. Вона дозволяє легко виконувати HTTP-запити різних типів (GET, POST, PUT тощо).

Метод GET є найбільш поширеним і використовується для отримання даних з веб-ресурсів. Відповідь сервера, що містить затребувані дані, називається payload. Для отримання відповіді на запит використовується об'єкт response. Об'єкт Response являє собою інструмент для аналізування результату запитів. Він виконується для вивчення даних, отриманих у результаті запиту.

HTTP-код стану 200 вказує на успішне виконання запиту. Для отримання коду стану можна скористатися командою, представленою на рисунку 3.16. Ця інформація є важливою для діагностики та відлагодження програм.

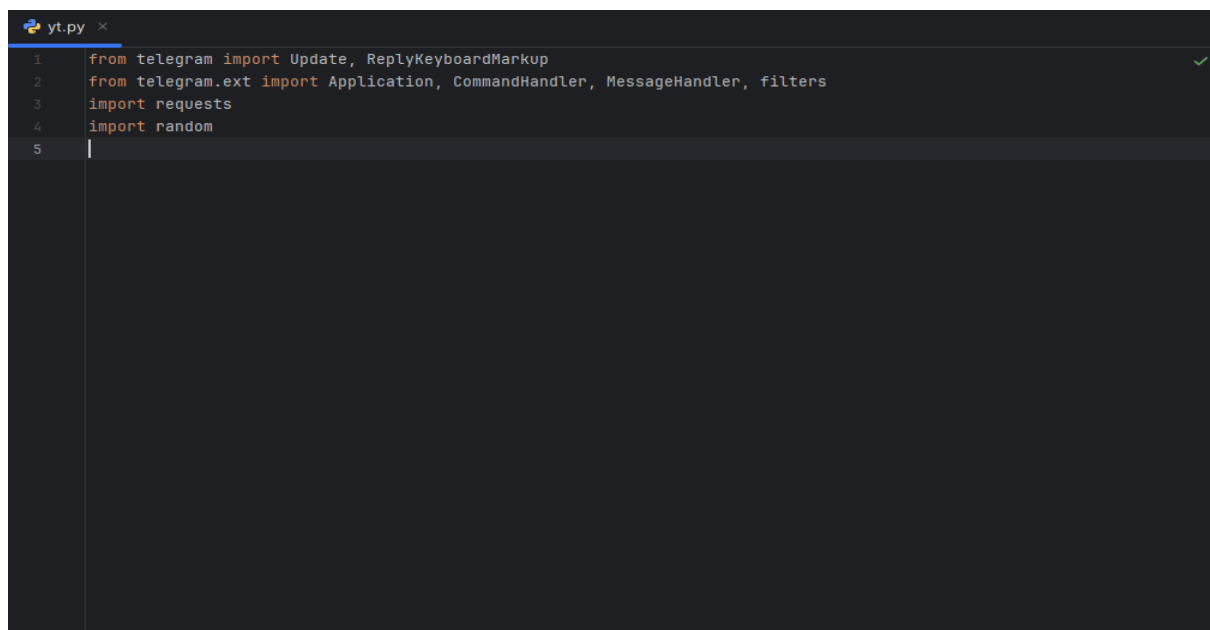


```
1 >>> response.status_code
2 200
```

Рисунок 3.16 – Команда коду станів та відповідь на неї

Для отримання байтів відповіді використовується атрибут `.content`. Це дозволяє отримати доступ до всього вмісту відповіді, включаючи заголовки та тіло. Однак, для подальшої обробки даних зазвичай необхідно декодувати їх з формату, в якому вони передані, наприклад, JSON. Розуміння принципів кодування та формату JSON є важливим для ефективної роботи з отриманими даними.

Після налаштування середовища розробки ми створюємо Python-файл, в якому починаємо писати наш код. Першим кроком є імпорт необхідних модулів. Імпорт дозволяє використовувати функції та класи з інших файлів або бібліотек. Таким чином, ми розширюємо можливості нашої програми. Процес імпорту модулів продемонстровано на рисунку 3.17.



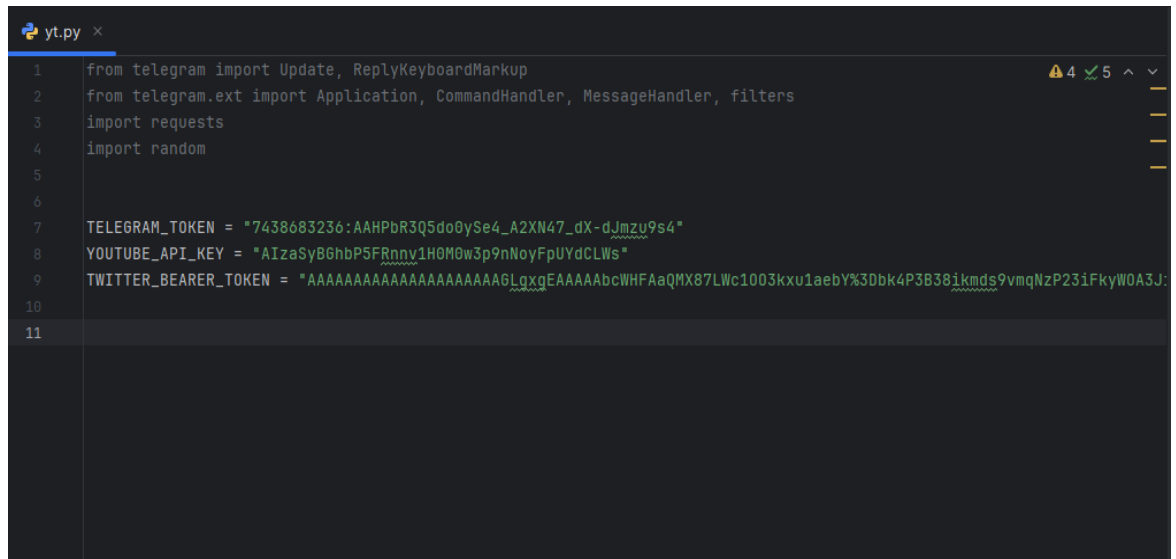
```
yt.py x
1 from telegram import Update, ReplyKeyboardMarkup
2 from telegram.ext import Application, CommandHandler, MessageHandler, filters
3 import requests
4 import random
5
```

Рисунок 3.17 – Імпортування бібліотек у код проекту

API-ключі відіграють ключову роль у взаємодії з сервісами YouTube та Twitter. Вони слугують для ідентифікації нашого додатку та надають йому певні дозволи. Завдяки ключам ми можемо виконувати різноманітні дії, такі як завантаження відео на YouTube або публікацію твітів. API-ключі є механізмом аутентифікації та авторизації, який забезпечує безпечну взаємодію додатків з API.

Вони дозволяють ідентифікувати додаток, контролювати його доступ до ресурсів та запобігати зловживанням. Крім того, ключі можуть надавати додаткові можливості, які недоступні для неавторизованих користувачів.

Команда `"/start"` є точкою входу для взаємодії з ботом. Після введення цієї команди користувач отримає доступ до основних функцій бота, представлених у вигляді меню.

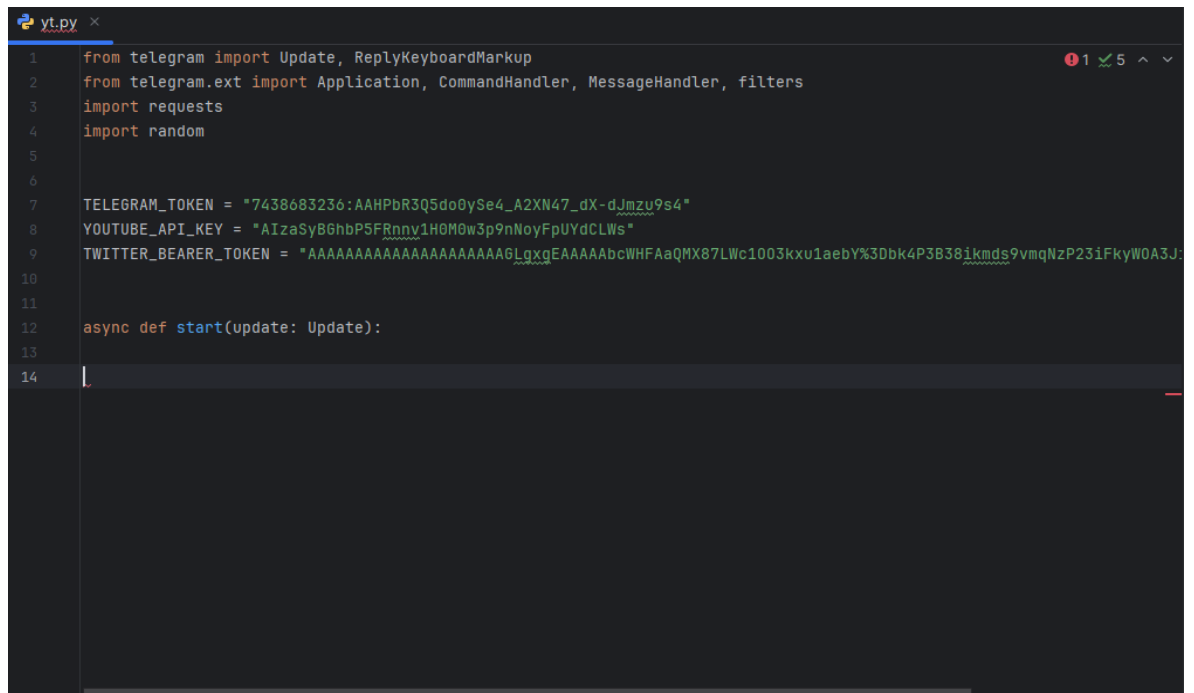


```

1 from telegram import Update, ReplyKeyboardMarkup
2 from telegram.ext import Application, CommandHandler, MessageHandler, filters
3 import requests
4 import random
5
6
7 TELEGRAM_TOKEN = "7438683236:AAHPbR3Q5do0ySe4_A2XN47_dX-dJmzu9s4"
8 YOUTUBE_API_KEY = "AIzaSyB6hbP5FRnnv1H0M0w3p9nNoyFpUYdCLWs"
9 TWITTER_BEARER_TOKEN = "AAAAAAAAAAAAAAAAAAGLgXqEAAAAAbcWHFAaQMX87LWc1003kxu1aebY%3Dbk4P3B38ikm9vmsqNzP23iFkyW0A3J:"
10
11

```

Рисунок 3.18 – Команда на додавання токенів у програму



```

1 from telegram import Update, ReplyKeyboardMarkup
2 from telegram.ext import Application, CommandHandler, MessageHandler, filters
3 import requests
4 import random
5
6
7 TELEGRAM_TOKEN = "7438683236:AAHPbR3Q5do0ySe4_A2XN47_dX-dJmzu9s4"
8 YOUTUBE_API_KEY = "AIzaSyB6hbP5FRnnv1H0M0w3p9nNoyFpUYdCLWs"
9 TWITTER_BEARER_TOKEN = "AAAAAAAAAAAAAAAAAAGLgXqEAAAAAbcWHFAaQMX87LWc1003kxu1aebY%3Dbk4P3B38ikm9vmsqNzP23iFkyW0A3J:"
10
11
12 async def start(update: Update):
13
14

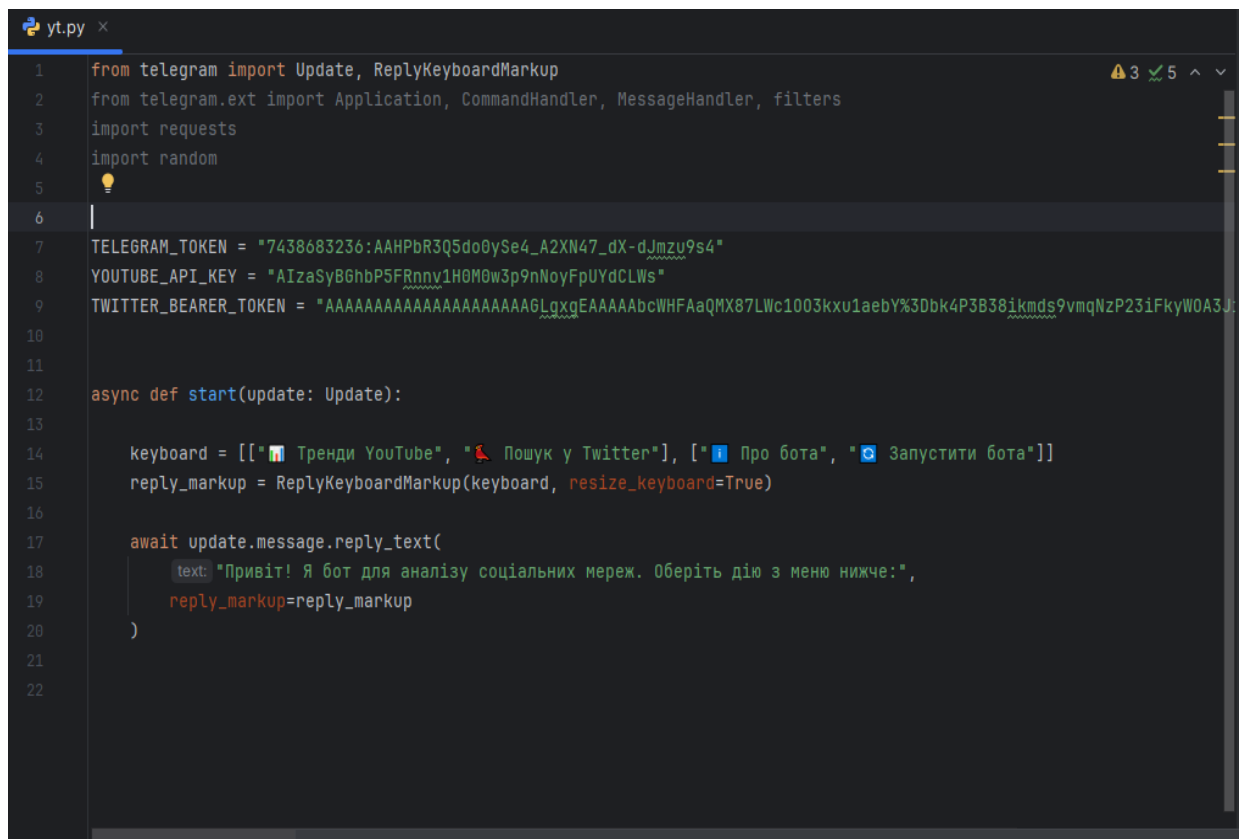
```

Рисунок 3.19 – Функція команди `/start`

Для аутентифікації бота використовується API-ключ, отриманий від @BotFather. Цей ключ вказує на конкретного бота і дозволяє взаємодіяти з ним. Для створення інтерактивного інтерфейсу користувача буде використано клас ReplyKeyboardMarkup, який дозволяє створювати кастомні клавіатури. Детальний приклад використання цього класу наведено на рисунку 3.19.

Клас ReplyKeyboardMarkup з бібліотеки python-telegram-bot дозволяє нам створювати кастомні клавіатури, які відображаються під полем введення тексту в чаті. Це спрощує взаємодію користувача з ботом, оскільки користувач може вибрати потрібну опцію з готового списку.

Використовуючи клас ReplyKeyboardMarkup, ми можемо створити інтерактивне меню, яке дозволить користувачам легко взаємодіяти з ботом. Наприклад, ми можемо запропонувати користувачам вибрати, яку інформацію вони хочуть отримати: тренди YouTube, новини чи щось інше (рис. 3.20).

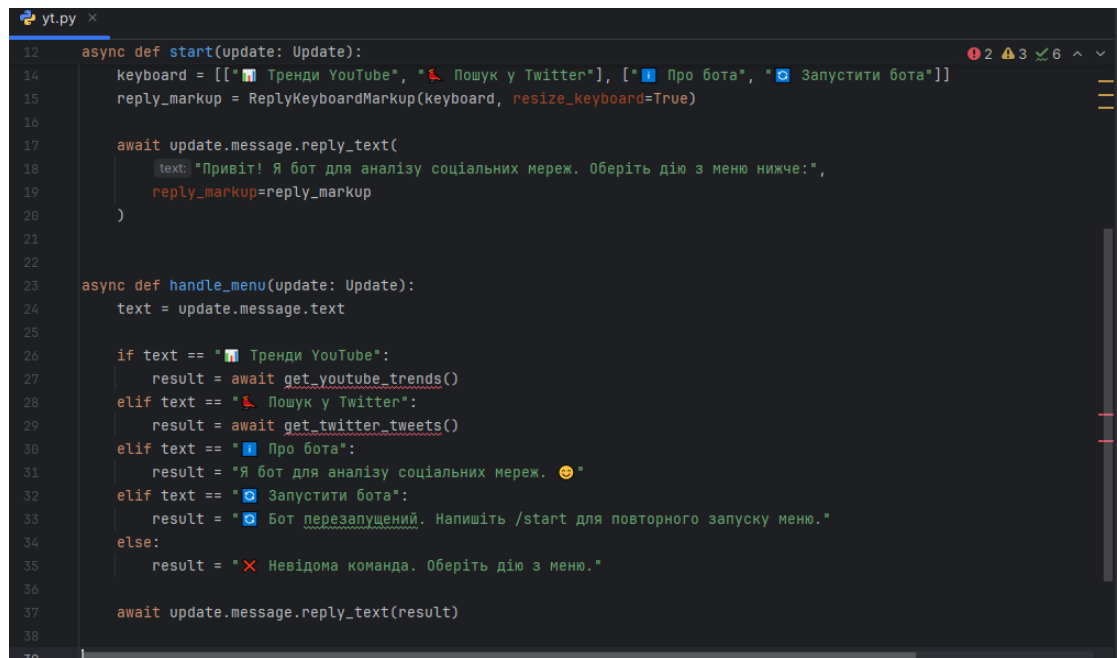


```

1  from telegram import Update, ReplyKeyboardMarkup
2  from telegram.ext import Application, CommandHandler, MessageHandler, filters
3  import requests
4  import random
5
6
7  TELEGRAM_TOKEN = "7438683236:AAHPbR3Q5do0ySe4_A2XN47_dX-dJmzu9s4"
8  YOUTUBE_API_KEY = "AIzaSyB6hbP5FRnnv1H0M0w3p9nNoyFpUYdCLWs"
9  TWITTER_BEARER_TOKEN = "AAAAAAAAAAAAAAAAAAGLqxqEAAAAAbcWHFaaQMX87LWc1003kxu1aebY%3Dbk4P3B38ikmds9vmqNzP231FkyWOA3J"
10
11
12  async def start(update: Update):
13
14      keyboard = [{"📺 Тренди YouTube", "🔍 Пошук у Twitter"}, {"ℹ️ Про бота", "🚀 Запустити бота"}]
15      reply_markup = ReplyKeyboardMarkup(keyboard, resize_keyboard=True)
16
17      await update.message.reply_text(
18          text="Привіт! Я бот для аналізу соціальних мереж. Оберіть дію з меню нижче:",
19          reply_markup=reply_markup
20      )
21
22
  
```

Рисунок 3.20 – Написання шаблонних кнопок

Щоб бот міг реагувати на дії користувача, які виконуються за допомогою кнопок, створимо спеціальну функцію. Для цього скористаємося можливостями бібліотеки `python-telegram-bot`. Основна мета цієї функції — реагувати на різні кнопки, які користувач може натискати в меню, і надавати відповідні відповіді (рис. 3.21).



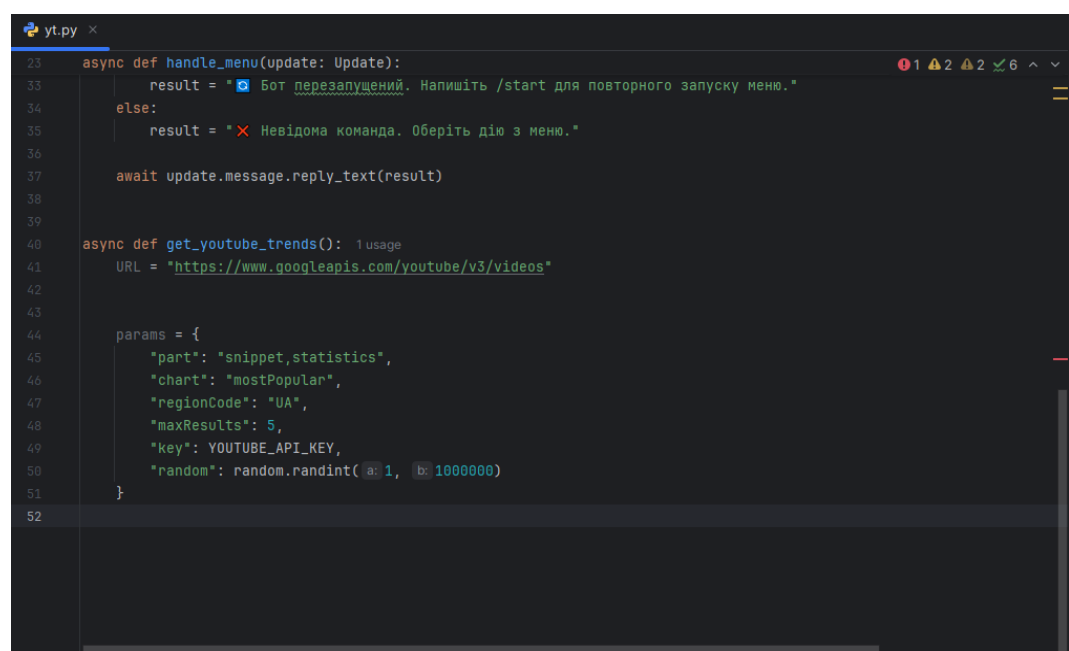
```

12 async def start(update: Update):
13     keyboard = [{"Тренди YouTube", "Пошук у Twitter"}, {"Про бота", "Запустити бота"}]
14     reply_markup = ReplyKeyboardMarkup(keyboard, resize_keyboard=True)
15
16     await update.message.reply_text(
17         text="Привіт! Я бот для аналізу соціальних мереж. Оберіть дію з меню нижче:",
18         reply_markup=reply_markup
19     )
20
21
22
23 async def handle_menu(update: Update):
24     text = update.message.text
25
26     if text == "Тренди YouTube":
27         result = await get_youtube_trends()
28     elif text == "Пошук у Twitter":
29         result = await get_twitter_tweets()
30     elif text == "Про бота":
31         result = "Я бот для аналізу соціальних мереж. 😊"
32     elif text == "Запустити бота":
33         result = "Бот перезапущений. Напишіть /start для повторного запуску меню."
34     else:
35         result = "❌ Невідома команда. Оберіть дію з меню."
36
37     await update.message.reply_text(result)
38
39

```

Рисунок 3.21 - Оброблення повідомлень з кнопками

Після цього, потрібна функція для отримання трендів YouTube. (рис. 3.22).



```

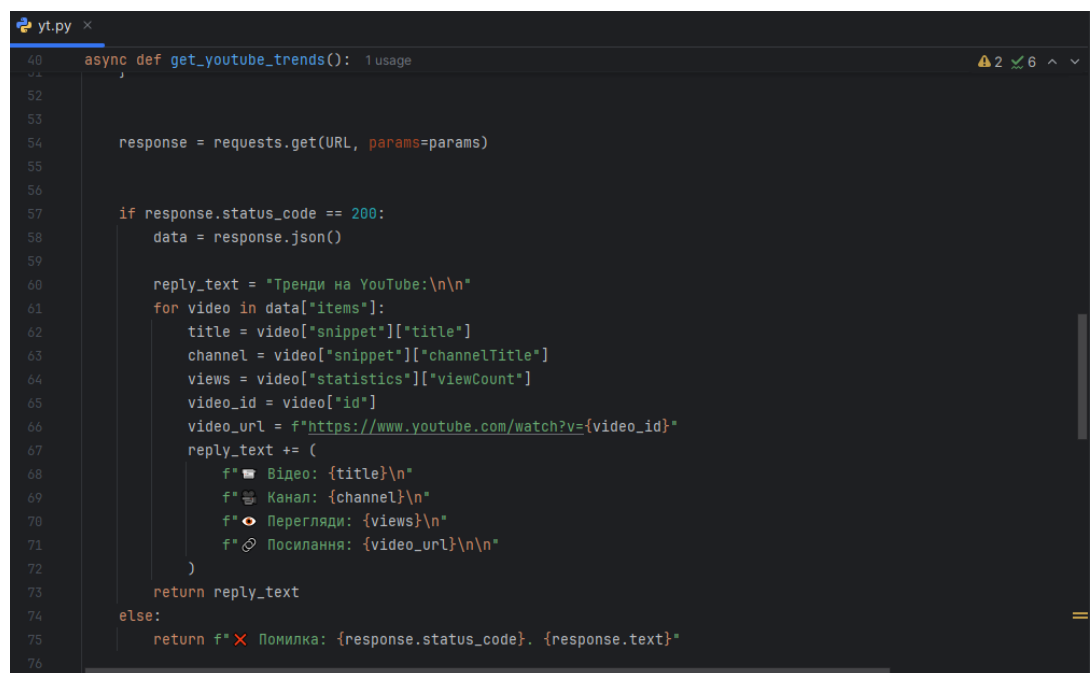
23 async def handle_menu(update: Update):
24     result = "Бот перезапущений. Напишіть /start для повторного запуску меню."
25     else:
26         result = "❌ Невідома команда. Оберіть дію з меню."
27
28     await update.message.reply_text(result)
29
30
31
32 async def get_youtube_trends():
33     URL = "https://www.googleapis.com/youtube/v3/videos"
34
35     params = {
36         "part": "snippet,statistics",
37         "chart": "mostPopular",
38         "regionCode": "UA",
39         "maxResults": 5,
40         "key": YOUTUBE_API_KEY,
41         "random": random.randint(1, 1000000)
42     }
43
44
45

```

Рисунок 3.22 - Функція для отримання трендів YouTube

Функція реалізує асинхронний виклик YouTube Data API для отримання списку популярних відео. Запит формується з використанням таких параметрів, як регіон (наприклад, "ua"), тип відео (наприклад, "music"), максимальна кількість результатів та API-ключ. Для уникнення кешування результатів до запиту додається випадковий параметр.

Після відправки запиту функція перевірить, чи буде він успішним. Якщо відповідь матиме статус 200 (успішно), функція обробить отримані дані, формуючи текст відповіді, який містить інформацію про кожне відео: назву відео, назву каналу, кількість переглядів та посилання на відео. Цей текст потім повернеться як результат функції (рис. 3.23). Якщо запит не вдасться, функція поверне повідомлення про помилку, яке включатиме статусний код та текст помилки.



```

40  async def get_youtube_trends():
41      ,
42
43
44
45
46
47
48
49
50
51
52
53
54      response = requests.get(URL, params=params)
55
56
57
58      if response.status_code == 200:
59          data = response.json()
60
61          reply_text = "Тренди на YouTube:\n\n"
62          for video in data["items"]:
63              title = video["snippet"]["title"]
64              channel = video["snippet"]["channelTitle"]
65              views = video["statistics"]["viewCount"]
66              video_id = video["id"]
67              video_url = f"https://www.youtube.com/watch?v={video_id}"
68              reply_text += (
69                  f"📺 Відео: {title}\n"
70                  f"📺 Канал: {channel}\n"
71                  f"👁 Перегляди: {views}\n"
72                  f"🔗 Посилання: {video_url}\n\n"
73              )
74          return reply_text
75      else:
76          return f"❌ Помилка: {response.status_code}. {response.text}"

```

Рисунок 3.23 – Обробка результатів та формування тексту відповіді

Функція виконує асинхронний запит до ендпоінту "recent search" Twitter API v2. Запит включає в себе заголовок Authorization з Bearer Token для аутентифікації та параметри query і max_results для визначення критеріїв пошуку. Результатом виконання запиту є список твітів, що відповідають заданим критеріям. Детальна структура запиту та відповіді продемонстрована на рисунку 3.24.

```

40  async def get_youtube_trends(): 1 usage
62      title = video["snippet"]["title"]
63      channel = video["snippet"]["channelTitle"]
64      views = video["statistics"]["viewCount"]
65      video_id = video["id"]
66      video_url = f"https://www.youtube.com/watch?v={video_id}"
67      reply_text += (
68          f"📺 Відео: {title}\n"
69          f"📺 Канал: {channel}\n"
70          f"👁 Перегляди: {views}\n"
71          f"🔗 Посилання: {video_url}\n\n"
72      )
73      return reply_text
74  else:
75      return f"❌ Помилка: {response.status_code}. {response.text}"
76
77
78  async def get_twitter_tweets(): 1 usage
79      URL = "https://api.twitter.com/2/tweets/search/recent"
80      headers = {
81          "Authorization": f"Bearer {TWITTER_BEARER_TOKEN}"
82      }
83      params = {
84          "query": "новини",
85          "max_results": 10,
86      }

```

Рисунок 3.24 - Функція для пошуку твітів через Twitter API v2

Функція аналізує HTTP-код відповіді на запит. При коді 200 вважається, що запит виконаний успішно, і функція продовжує обробку отриманих даних. З тіла відповіді витягується масив твітів, з яких формується текстовий результат. Логіка обробки відповіді представлена на рисунку 3.25.

```

78  async def get_twitter_tweets(): 1 usage
82  }
83  params = {
84      "query": "новини",
85      "max_results": 10,
86  }
87
88
89  response = requests.get(URL, headers=headers, params=params)
90
91
92  if response.status_code == 200:
93      data = response.json()
94
95      reply_text = "Останні популярні твіти:\n\n"
96      for tweet in data["data"]:
97          text = tweet["text"]
98          reply_text += f"🐦 {text}\n\n"
99      return reply_text
100  elif response.status_code == 401:
101      return f"❌ Помилка: Невірний Bearer Token. Перевірте ваш токен у Twitter Developer Platform."
102  elif response.status_code == 403:
103      return f"❌ Помилка: Доступ заборонено. Можливо, ваш обліковий запис не має прав доступу до цього ресурсу."
104  else:
105      return f"❌ Помилка: Не вдалося отримати дані з Twitter API. Код помилки: {response.status_code}, Деталі: {res
106
107

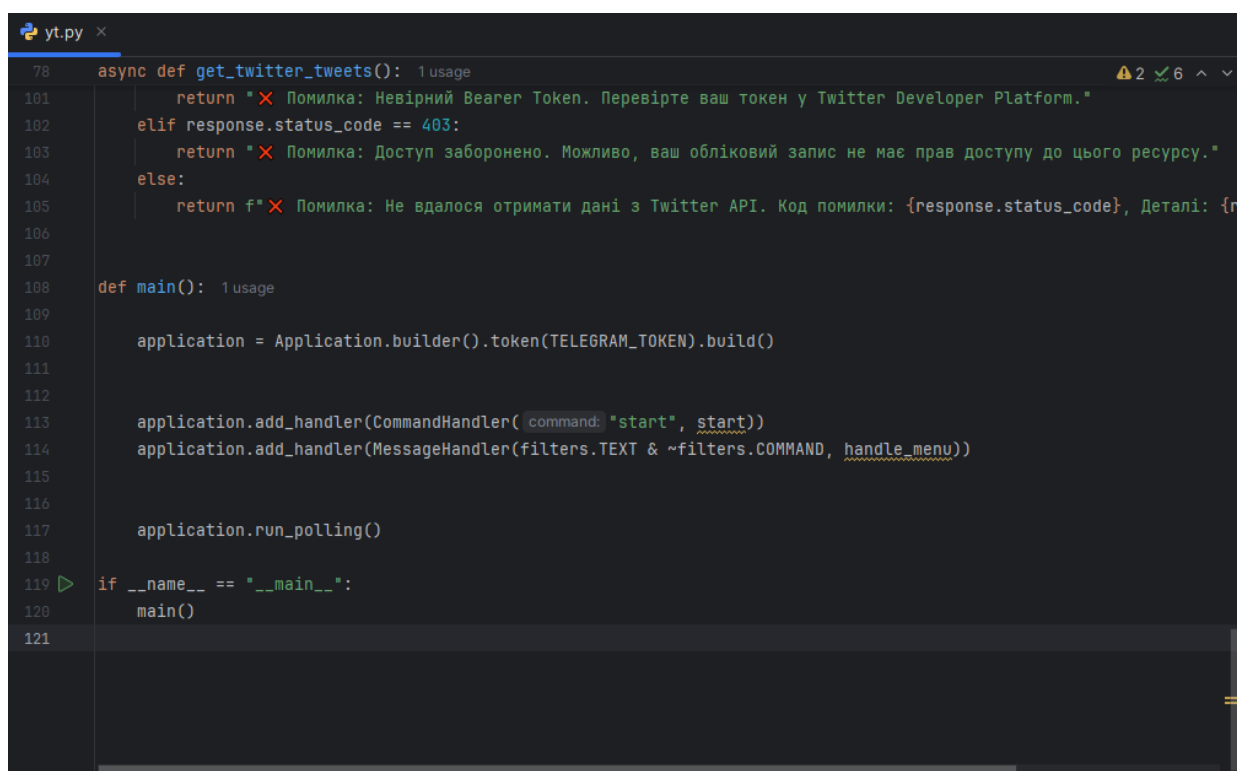
```

Рисунок 3.25 - Обробка результатів та формування тексту відповіді

Функція передбачає обробку різних типів помилок, пов'язаних з виконанням запиту до Twitter API. У разі невірному Bearer Token, відсутності дозволів або інших проблем функція повертає відповідне повідомлення про помилку, що дозволяє швидко виявити та усунути причину проблеми.

Функція `main` є точкою запуску програми. Вона створює екземпляр класу, який відповідає за взаємодію з Telegram API. Для аутентифікації бота використовується API-токен, який отримують при створенні бота в BotFather.

Створюємо функції-обробники для команд `"/start"` та текстових повідомлень. Функція для `"/start"` відповідає за ініціалізацію діалогу та відображення меню. Функція для обробки текстових повідомлень аналізує вміст повідомлення і виконує відповідні дії в залежності від контексту взаємодії (рис. 3.26).



```

78  async def get_twitter_tweets(): 1 usage
101      return "❌ Помилка: Невірний Bearer Token. Перевірте ваш токен у Twitter Developer Platform."
102  elif response.status_code == 403:
103      return "❌ Помилка: Доступ заборонено. Можливо, ваш обліковий запис не має прав доступу до цього ресурсу."
104  else:
105      return f"❌ Помилка: Не вдалося отримати дані з Twitter API. Код помилки: {response.status_code}, Деталі: {r
106
107
108  def main(): 1 usage
109
110      application = Application.builder().token(TELEGRAM_TOKEN).build()
111
112
113      application.add_handler(CommandHandler(command="start", start))
114      application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND, handle_menu))
115
116
117      application.run_polling()
118
119  if __name__ == "__main__":
120      main()
121

```

Рисунок 3.26 - Основна функція для запуску бота

Після додавання обробників команд ми запускаємо бота за допомогою методу, який починає опитування Telegram-сервера для отримання нових оновлень. Це означає, що бот постійно перевіряє нові повідомлення або команди від користувачів і реагує на них у реальному часі.

Основна функція слугує точкою входу в нашу програму. Вона виконується лише тоді, коли ми запускаємо скрипт напряму, а не підключаємо його як бібліотеку до іншого проекту. Це дозволяє нам чітко розділити логіку запуску бота та його внутрішню роботу."

Механізм запуску гарантує, що наш бот буде запущений лише тоді, коли ми цього хочемо. Це важливо для того, щоб уникнути непередбачених ситуацій, коли бот може запуснитися випадково..

Було розроблено архітектуру телеграм-боту, яка забезпечує ефективну взаємодію з користувачами, підтримку основних функціональних можливостей і гнучкість для подальшого масштабування.

Результати проектування включають визначення ключових компонентів системи, таких як інтерфейс взаємодії з Telegram API, модуль обробки команд, база даних для збереження інформації, а також логіка серверної частини. Було враховано принципи модульності та розподілу обов'язків між компонентами, що дозволяє підвищити продуктивність та спрощує підтримку системи.

Особливу увагу приділено аспектам безпеки, зокрема захисту даних користувачів та стабільності під час обробки великої кількості запитів. Архітектура передбачає використання асинхронних технологій, що забезпечує високу швидкість реакції бота.

Таким чином, створена архітектура відповідає вимогам сучасних телеграм-ботів, що дозволяє розробити функціональний, масштабований та зручний для користувачів інструмент, який можна адаптувати до різних сценаріїв використання.

3.2 Реалізація функціоналу телеграм-боту для збору та обробки даних

Цей розділ присвячений практичній реалізації Telegram-бота, який буде виконувати завдання збору та обробки даних з різних соціальних мереж.

Telegram-бот для моніторингу соцмереж – це потужний інструмент, який дозволяє економити час і підвищити ефективність маркетингових активностей. Він особливо корисний для компаній, які хочуть бути в курсі останніх трендів і швидко реагувати на зміни в онлайн-просторі.

Щоб розпочати роботу бота, необхідно знайти його в пошуку в Telegram. Для цього запустіть додаток Telegram на своєму пристрої. У верхній частині екрана знайдіть і натисніть на іконку лупи, яка відповідає за пошук. У полі пошуку введіть назву бота @socialmediabot (Рис. 3.27).

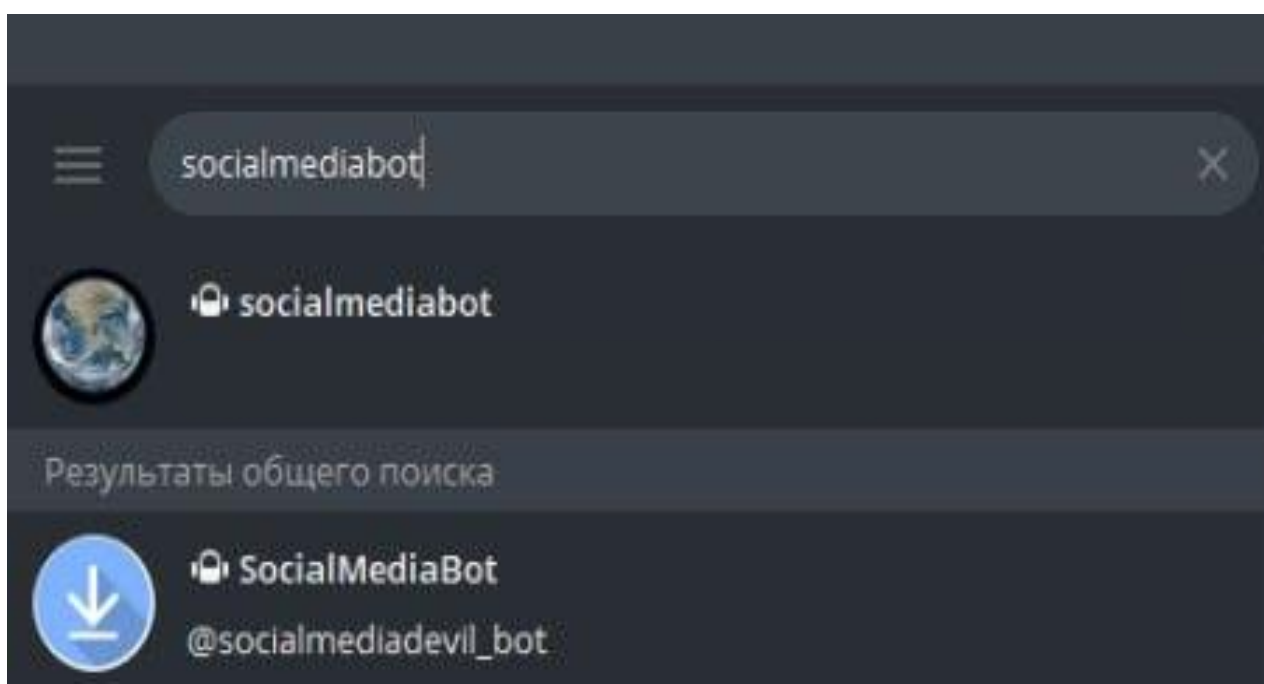


Рисунок 3.27 – Пошук телеграм-бота @socialmediabot

Натиснувши «Запустити» або «/start» (Рис. 3.28), бот привітається з користувачем, опише свою функцію та запропонує обрати дію з меню (Рис. 3.29).

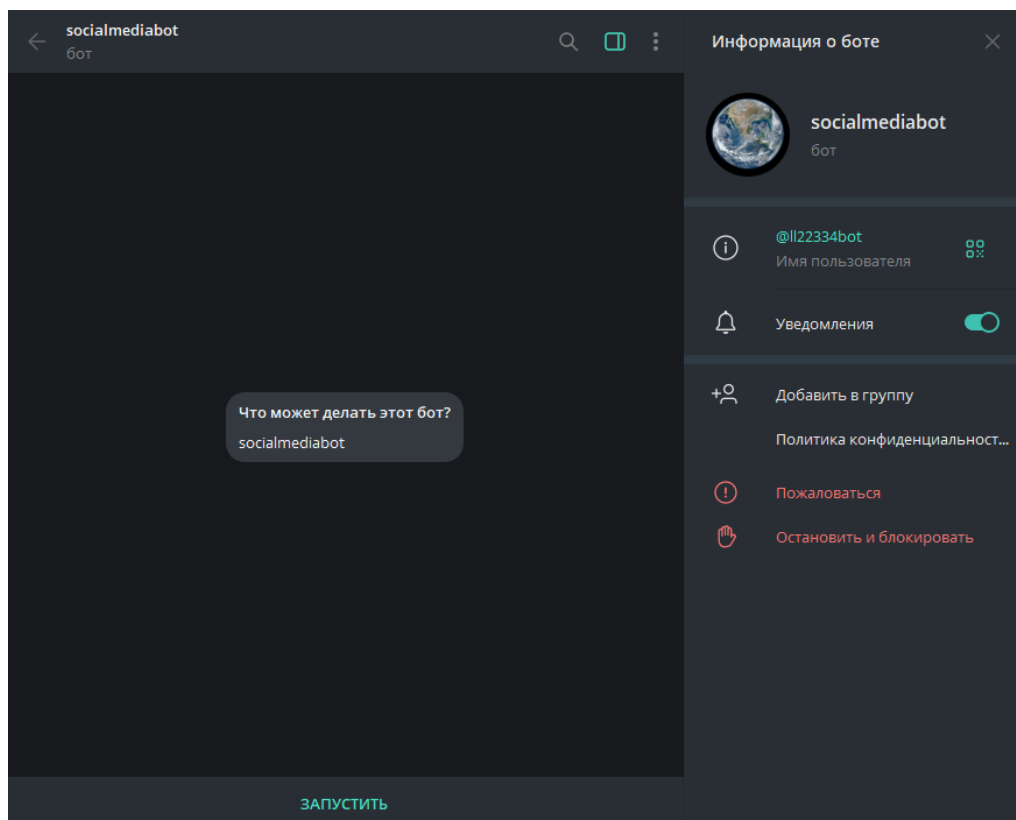


Рисунок 3.28 – Запуск телеграм-бота

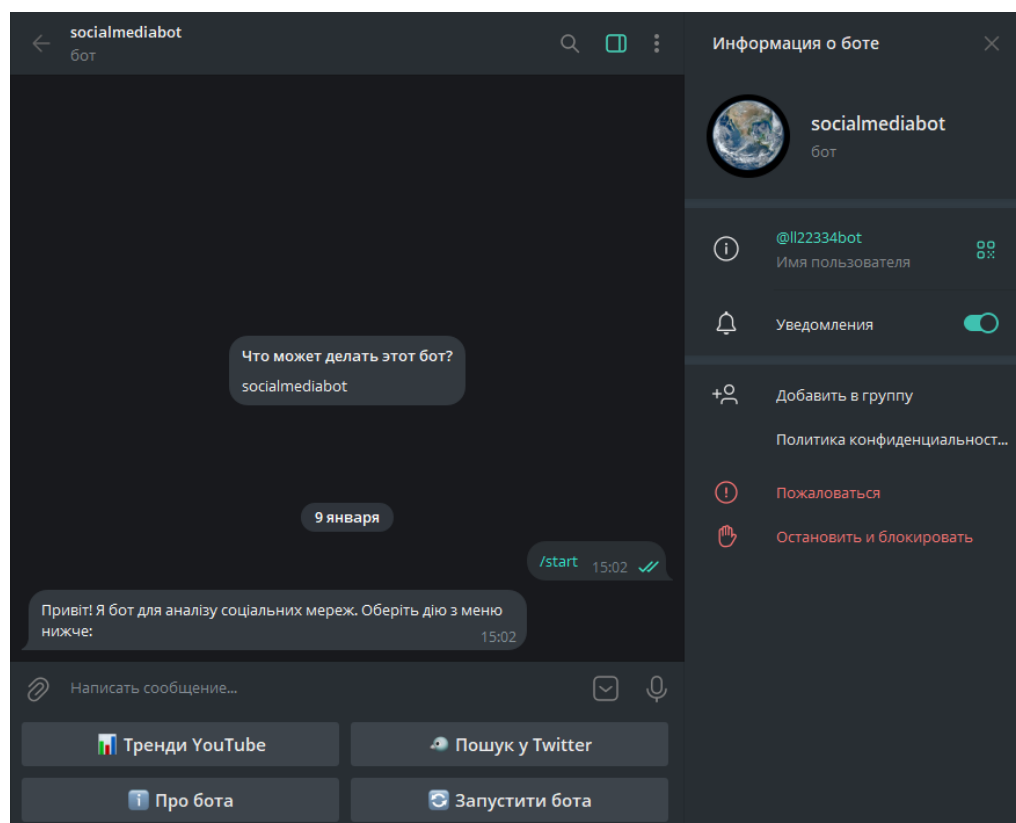


Рисунок 3.29 – Перше повідомлення від телеграм-бота

Команда «Про бота» дає можливість користувачу ознайомитись з ботом детальніше. Оскільки в одному повідомленні описується найважливіша інформація (Рис. 3.30). А саме: для яких конкретних цілей створений, які доступні команди наявні для взаємодії, наприклад, «Аналіз трендів YouTube» – презентує перелік найпопулярніших відео в вибраному регіоні на даний момент часу, «Пошук у Twitter» – видає останні популярні твіти, а також посилання на них, «Перезапуск бота» – дає можливість розпочати все з чистого аркушу. Окрім цього, відповідь бота містить інформацію про додаткові ресурси: FAQ (відповіді на часті питання), контактні дані для зв'язку з розробниками, політика конфіденційності, правила використання бота.

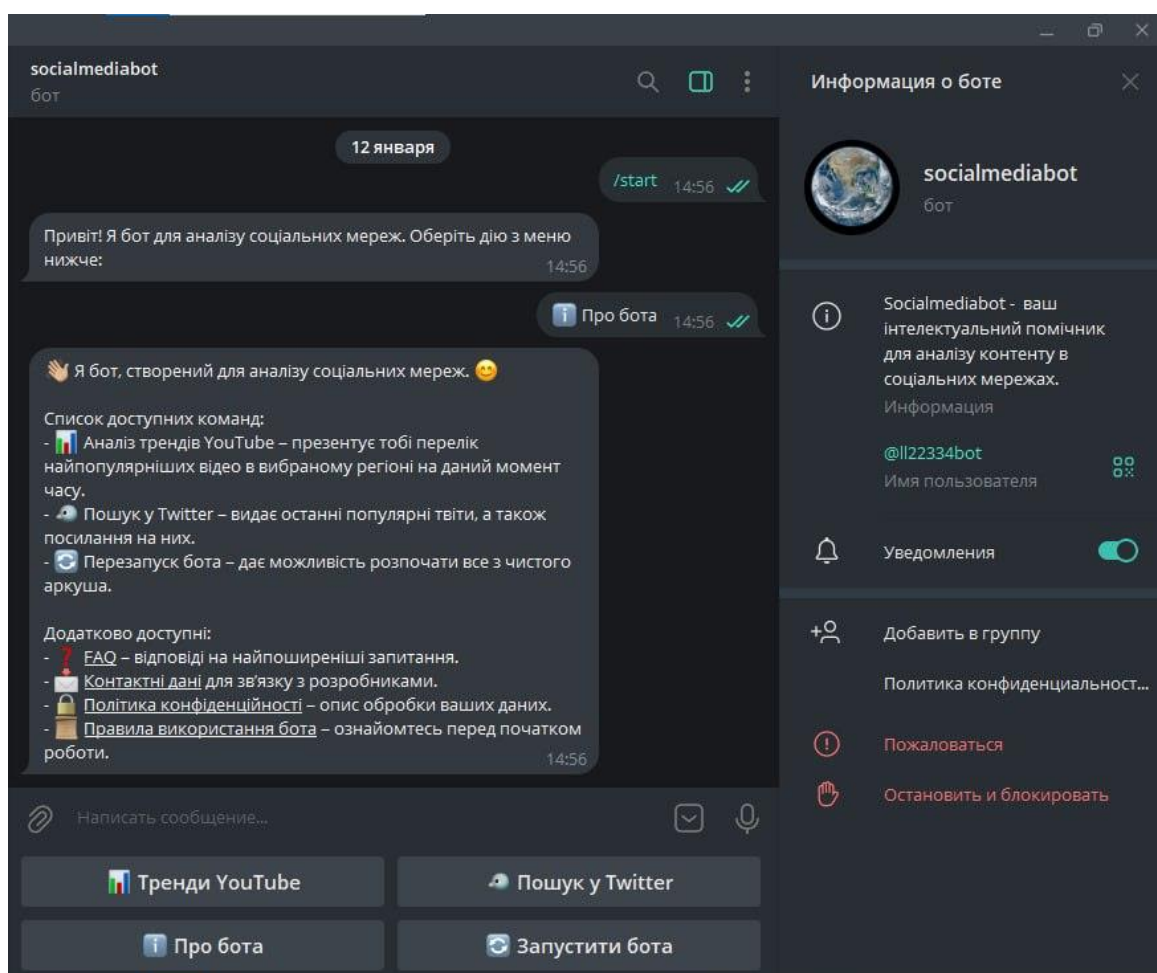


Рисунок 3.30 – Функціонал команди «Про бота»

Це повідомлення важливе, оскільки воно допомагає користувачеві зрозуміти, що він може робити з ботом. Це перший крок до взаємодії з ботом.

Рисунок 3.31 демонструє результат роботи Telegram-бота SocialMediaBot, який спеціалізується на аналізі трендів у соціальних мережах, включно у відеохостингу YouTube.

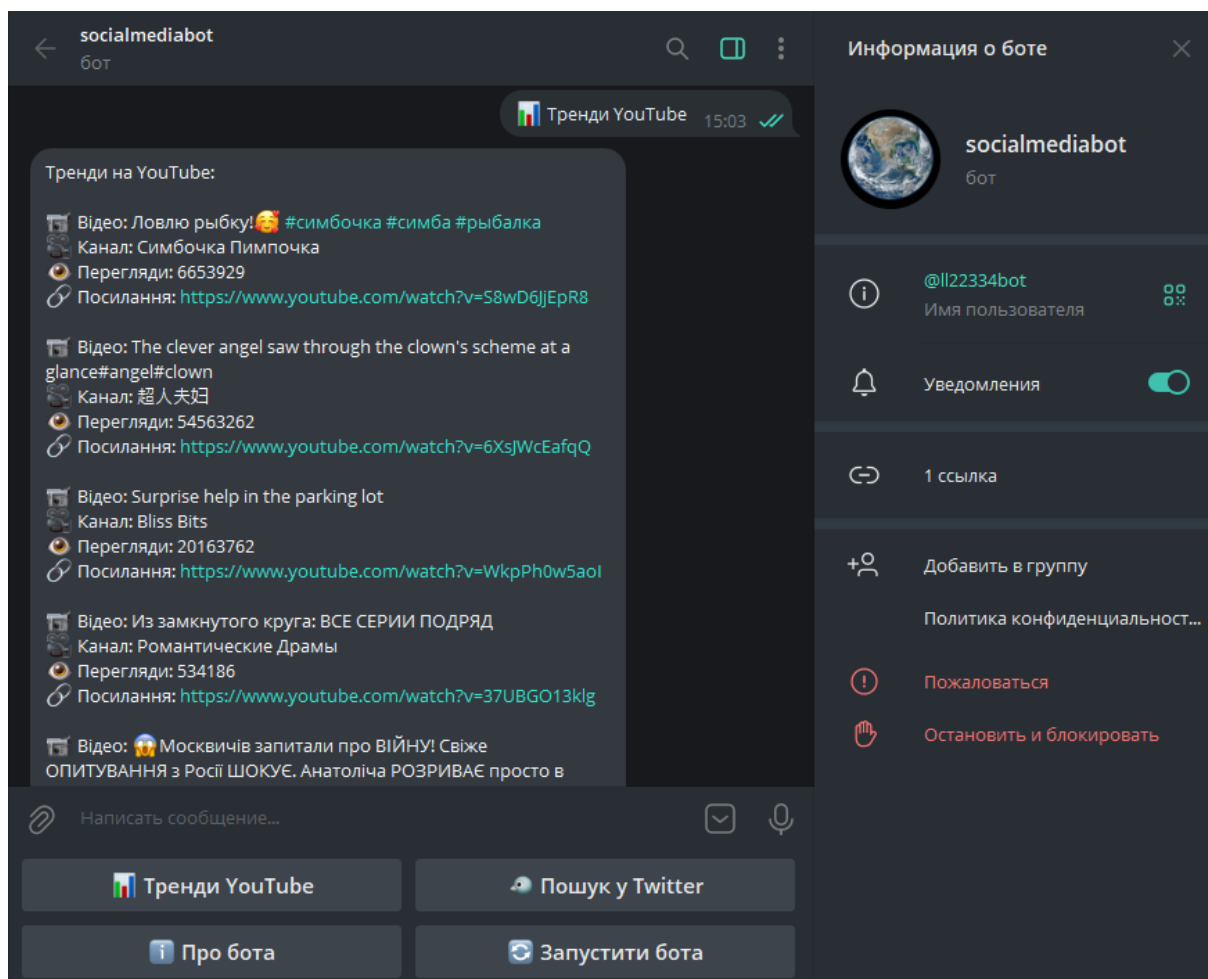


Рисунок 3.31 – Команда «Тренди YouTube»

Завдяки API-токену, бот здійснює пошук і виводить список відео на YouTube, які за короткий проміжок часу набрали значну кількість переглядів. Це означає, що бот шукає відео, які стали популярними дуже швидко, тобто набрали багато переглядів за відносно короткий час. Це можуть бути вірусні відео, трендові кліпи, новини тощо. Кожне відео в списку супроводжується прямим посиланням на нього. Це дозволяє користувачеві одразу перейти до перегляду відео, не виконуючи додаткових пошуків.

За допомогою бота користувач зможе легко знаходити найактуальніші відео на YouTube. Він сканує платформу та формує список найпопулярніших роликів,

враховуючи кількість переглядів та релевантність до ваших пошукових запитів. Для кожного відео ви отримаєте детальну інформацію: назву, канал, кількість переглядів тощо. Бот підтримує різні мови, що дозволяє вам розширити свій кругозір та дізнаватися про тренди з усього світу. Інтуїтивно зрозумілий інтерфейс забезпечує зручний доступ до потрібної інформації.

Переваги використання цієї команди:

- Економія часу – бот автоматично збирає і аналізує великі обсяги даних;
- Актуальність – інформація про тренди завжди актуальна;
- Зручність – інтерфейс бота простий і зрозумілий.

Рисунок 3.32 демонструє роботу Telegram-бота SocialMediaBot, який використовується для пошуку актуальних новин у Twitter. На прикладі показано, що бот знайшов кілька твітів, найпопулярніших за останню добу.

Бот використовує API Twitter для формування спеціального запиту, який вказує, які саме твіти він хоче знайти. Він видає список твітів, які були опубліковані за останню добу і мають найбільшу кількість ретвітів або лайків. Після відправки запиту, Twitter обробляє його і повертає боту відповідь у форматі, який він може зрозуміти. Бот отримує отримані дані, аналізує їх і вибирає найрелевантніші твіти відповідно до заданих критеріїв (наприклад, найпопулярніші за останню добу). В результаті, бот формує повідомлення для користувача, яке містить список знайдених твітів, в якому зазначений автор допису, пряме посилання та безпосередньо текст твітів.

Основні функції команди:

- Виведення результатів у зручному форматі. Знайдені твіти відображаються у вигляді списку, де для кожного твіта вказано автора, текст і посилання на оригінальний твіт в Twitter.

- Підтримка різних мов. Судячи з наявності російськомовних і англomовних твітів, бот підтримує кілька мов.
- Актуальність інформації. Виведені твіти є досить свіжими, що свідчить про ефективність пошуку.

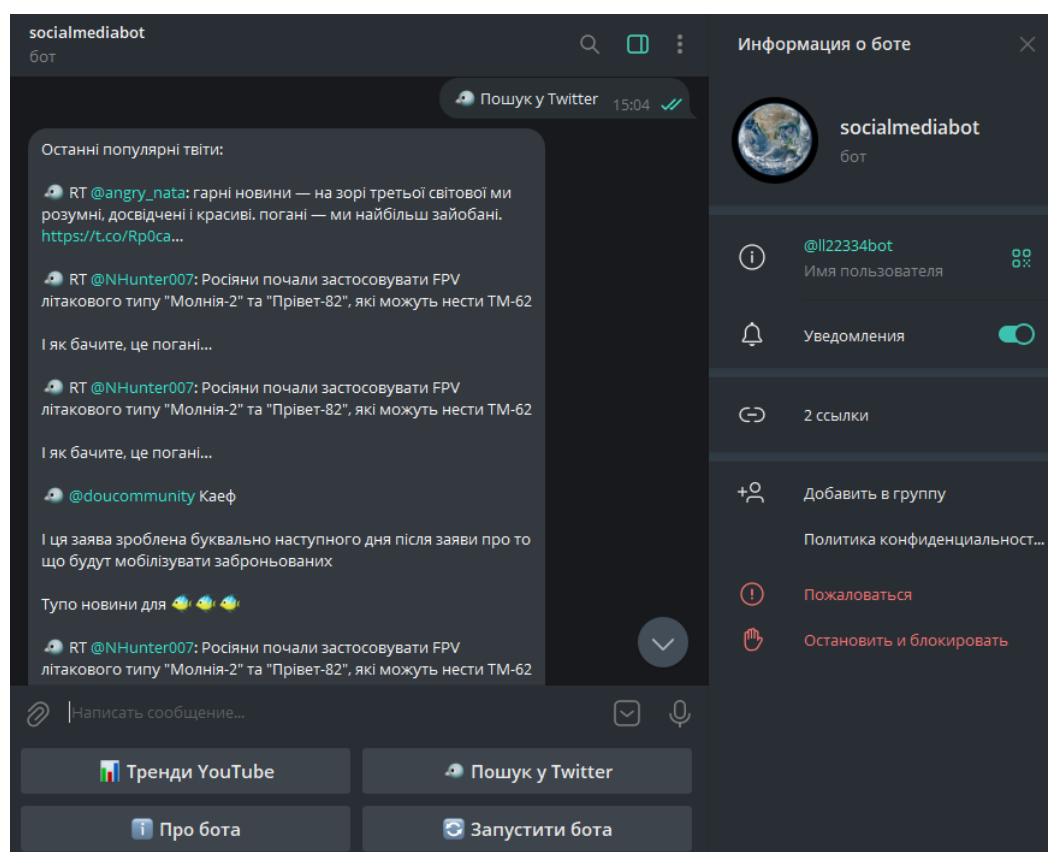


Рисунок 3.32 – «Пошук у Twitter»

Кнопка "Перезапустити бота" у контексті Рисунок 3.33 виконує функцію повного оновлення роботи бота. Перезапуск бота (після натискання цієї кнопки) дозволить розпочати роботу з чистого аркуша. Всі поточні процеси будуть завершені, а тимчасові дані видалені. Це може бути необхідно у випадку зависання бота, виконання ним неправильних дій або для оновлення інформації. Розробники можуть вносити зміни в роботу бота, і для того, щоб користувач міг скористатися новими функціями, потрібно перезапустити бота. Іноді перезапуск може знадобитися через технічні проблеми на сервері, де розміщений бот.

Після перезапуску бот буде готовий до виконання нових команд, таких як пошук новин чи аналіз даних.

Кнопка "Запустити бота" є важливим інструментом для користувача, оскільки дозволяє вирішити різноманітні проблеми, пов'язані з роботою бота. Однак, перед тим як натиснути цю кнопку, варто переконатися, що інші способи вирішення проблеми не допомогли.

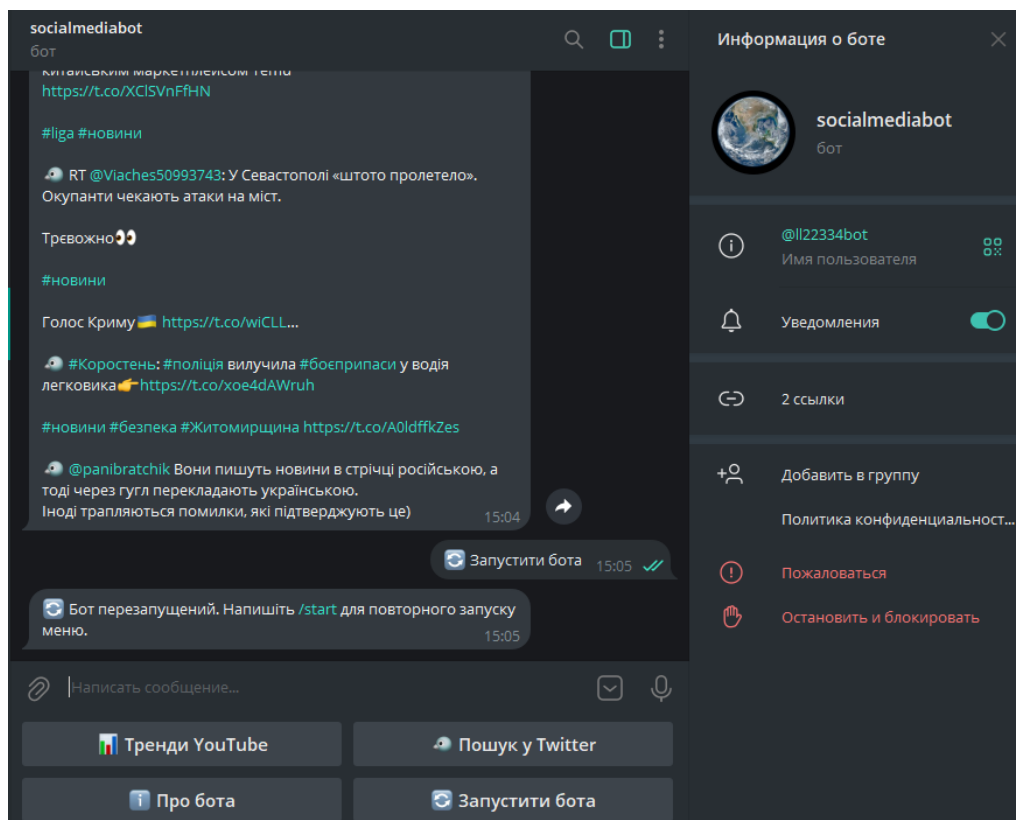


Рисунок 3.33 – Функція повного оновлення команди «Запустити бота»

3.3. Аналіз ефективності телеграм-боту порівняно з іншими інструментами

Telegram-боти є популярним інструментом, який широко використовується для автоматизації різних завдань, зокрема, для збору та аналізу даних із соціальних

мереж. Їх ефективність у порівнянні зі спеціалізованими інструментами варта детального розгляду з урахуванням переваг та недоліків.

Telegram є одним із найбільш популярних месенджерів, а створення ботів для нього відносно просте завдяки наявності багатих бібліотек та інструментів розробки. Це робить його доступним та простим у використанні. Крім того, Telegram-боти можуть бути адаптовані під широкий спектр завдань аналізу соціальних медіа, від простого пошуку згадок до складного аналізу настроїв. Вони також можуть легко інтегруватися з іншими сервісами та системами, такими як бази даних, системи аналізу даних та інші боти. Можливість створення персоналізованих ботів, які відповідають конкретним потребам користувача, додає ще одну значну перевагу. Багато бібліотек та інструментів для розробки Telegram-ботів є безкоштовними, що робить їх доступними для широкого кола користувачів.

Однак, Telegram-боти мають і певні недоліки. Наприклад, можливості API Telegram можуть бути недостатніми для деяких складних завдань аналізу даних. Для обробки великих обсягів даних Telegram-боти можуть бути менш ефективними порівняно зі спеціалізованими інструментами. Вбудовані можливості для візуалізації даних у Telegram-ботах обмежені, що може бути значним недоліком для користувачів, які потребують наочних графічних представлень даних. Крім того, для обробки великих обсягів даних та виконання складних аналітичних завдань може знадобитися додаткове налаштування та масштабування, що може бути не так просто здійснити.

Таким чином, Telegram-боти мають ряд вагомих переваг, що робить їх зручними та доступними для широкого кола користувачів. Однак, вони також мають певні обмеження, які можуть впливати на їхню ефективність у порівнянні з іншими спеціалізованими інструментами для аналізу контенту соціальних мереж.

Існує багато спеціалізованих інструментів, що допомагають автоматизувати цей процес та надають детальну інформацію. Вони виконують різні функції, такі як збір даних, включаючи моніторинг згадок бренду, ключових слів і хештегів у різних соціальних мережах. Інструменти також аналізують настрої, визначаючи тон згадок як позитивний, негативний або нейтральний. Вони можуть виявляти інфлюенсерів,

знаходячи лідерів думок у певній ніші. Крім того, вони здійснюють порівняльний аналіз, допомагаючи порівнювати ефективність різних кампаній та конкурентів. Спеціалізовані інструменти можуть генерувати візуальні звіти для презентації результатів.

Популярні інструменти для аналізу контенту в соціальних мережах включають:

Hootsuite, який є одним з найвідоміших інструментів для управління соціальними мережами. Він дозволяє планувати публікації, відстежувати згадки та аналізувати результати, пропонуючи зручний інтерфейс та широкий функціонал.

Buffer фокусується на плануванні публікацій та аналізі їхньої ефективності. Цей інструмент простий у використанні та ідеально підходить для малого та середнього бізнесу.

Sprout Social є потужним інструментом для управління соціальними мережами з широкими можливостями аналізу. Він дозволяє відстежувати конкурентів, аналізувати аудиторію та генерувати детальні звіти.

Завдяки своїй доступності та простоті використання, Telegram -боти швидко набули популярності серед розробників та користувачів. Однак, на ринку існує багато спеціалізованих інструментів, які також можуть виконувати подібні завдання. Для кращого розуміння ефективності Telegram-ботів в порівнянні з цими інструментами, розглянемо їхні характеристики за кількома ключовими параметрами в таблиці 3.1.

Таблиця 3.1 – Порівняння Telegram-ботів з іншими інструментами

Характеристика	Telegram-боти	Спеціалізовані інструменти
Доступність	Висока	Відносно низька
Гнучкість	Середня	Висока
Продуктивність	Середня	Висока
Функціональність	Широкий спектр	Спеціалізовані функції
Вартість	Низька	Висока

Таблиця порівняння Telegram-ботів зі спеціалізованими інструментами демонструє як сильні, так і слабкі сторони обох підходів. Telegram-боти мають високу доступність завдяки простоті налаштування та використання, що робить їх привабливими для широкого кола користувачів. У той же час, спеціалізовані інструменти часто мають відносно низьку доступність через складність налаштування та використання, але пропонують вищу гнучкість та продуктивність, що дозволяє ефективно обробляти великі обсяги даних і виконувати складні аналізи. Хоча Telegram-боти забезпечують широкий спектр функцій, ці функції загальні і можуть не відповідати вузько спеціалізованим потребам, які задовольняють спеціалізовані інструменти. Вартість Telegram-ботів є низькою, що робить їх доступними для багатьох користувачів, тоді як спеціалізовані інструменти можуть мати високу вартість, що обмежує їх використання у невеликих компаніях або серед індивідуальних користувачів.

Telegram-боти є відмінним вибором для простих завдань з аналізу соціальних медіа, таких як моніторинг згадок, збір даних за певними ключовими словами. Однак, для більш складних завдань, що вимагають високої продуктивності та широкого спектру функцій, краще використовувати спеціалізовані інструменти.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи на тему "Дослідження контенту соціальних мереж на основі телеграм-боту" були досягнуті значні результати, що підтверджують ефективність та доцільність використання телеграм-ботів для аналізу та моніторингу даних із соціальних мереж.

Були проаналізовані ключові концепції та підходи до вивчення соціальних мереж. Були розглянуті основні характеристики соціальних мереж, їхні функції та роль у сучасному суспільстві. Визначено, що соціальні мережі є важливим інструментом для комунікації, обміну інформацією та формування громадської думки. Також було досліджено теоретичні моделі, що пояснюють поведінку користувачів у соціальних мережах, такі як теорія соціального капіталу, теорія використання та задоволення потреб, а також теорія дифузії інновацій. Ці моделі дозволяють краще розуміти мотивацію користувачів та їхню активність у соціальних мережах.

Основними методами дослідження контенту є контент-аналіз, дискурс-аналіз та семантичний аналіз. Контент-аналіз дозволяє кількісно оцінювати вміст повідомлень у соціальних мережах, визначати основні теми та тенденції. Дискурс-аналіз фокусується на вивченні мовних практик та способів конструювання значень у текстах соціальних мереж. Семантичний аналіз дозволяє виявляти приховані смисли та структури у текстах.

Виявлено, що одним із ключових напрямів подальших досліджень є вивчення впливу алгоритмів рекомендацій соціальних мереж на формування інформаційних бульбашок та поляризацію громадської думки. Також важливим є аналіз візуального контенту, такого як зображення та відео, який стає все більш популярним у соціальних мережах. Іншим перспективним напрямом є дослідження емоційного впливу контенту та його здатності викликати різні емоційні реакції у користувачів.

Мова програмування Python. Було висвітлено переваги та особливості цієї мови програмування, які роблять її ідеальним вибором для розробки телеграм-

ботів. Python відомий своєю простотою та зручністю у використанні, що дозволяє швидко писати та налагоджувати код. Крім того, його багата екосистема бібліотек та модулів надає розробникам потужні інструменти для роботи з даними, обробки природної мови, машинного навчання та іншого. Це забезпечує високу гнучкість та масштабованість розробленого рішення.

Розглянуті можливості інтегрованого середовища розробки PyCharm, яке забезпечує зручність та ефективність процесу розробки. PyCharm надає потужні інструменти для написання, редагування та налагодження коду, а також інтеграцію з системами контролю версій, такими як Git. Важливою перевагою PyCharm є його можливість керувати залежностями проекту та інтеграція з віртуальними середовищами Python, що дозволяє легко керувати різними бібліотеками та пакетами. Це значно спрощує процес розробки та підтримки проекту.

Проведено огляд основних бібліотек, які використовуються у проекті для реалізації функціоналу телеграм-боту. Зокрема, було розглянуто бібліотеки для роботи з API Telegram, такі як `python-telegram-bot`, які забезпечують зручний інтерфейс для взаємодії з Telegram API. Ці бібліотеки надають потужні інструменти для роботи з великими обсягами даних, їхньої обробки та аналізу, що є ключовим для ефективного функціонування телеграм-боту.

Розроблено та впроваджено телеграм-бот, який автоматизує процес збору, аналізу та візуалізації даних з різних соціальних мереж. Це дозволяє значно скоротити час та ресурси, необхідні для проведення досліджень у цій сфері, а також підвищити точність і достовірність отриманих результатів. Впровадження телеграм-боту в аналітичні процеси дозволяє дослідникам оперативно отримувати актуальні дані, що є важливим для прийняття обґрунтованих рішень у сфері цифрового маркетингу, PR, соціології та інших галузях.

Отже, результати дослідження зробили значний внесок у розвиток методології аналізу контенту соціальних мереж та продемонстрували перспективи використання телеграм-ботів для цих цілей. Використання телеграм-ботів дозволяє значно підвищити якість та оперативність досліджень, що сприяє успішній адаптації стратегій комунікації та маркетингу в умовах швидких змін

інформаційного простору. Нові методи аналізу контенту, засновані на машинному навчанні та обробці природної мови, дозволяють отримати більш точні та релевантні результати, що сприяє кращому розумінню поведінки користувачів та ефективності різних видів контенту.

Таким чином, результати дослідження підтвердили ефективність та перспективність використання телеграм-ботів для аналізу контенту соціальних мереж, що робить значний внесок у розвиток цієї галузі та відкриває нові можливості для подальших досліджень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Плеханова, Т. М. "Соціальні мережі як новітня платформа для промоції медійного контенту." URL: <http://baltijapublishing.lv/omp/index.php/bp/catalog/download/154/4614/9697-1> (2021).
2. Пугач, А. В. "ПРОФОРІЄНТАЦІЙНИЙ ПОТЕНЦІАЛ СОЦІАЛЬНИХ МЕРЕЖ У ФОРМУВАННІ ЖИТТЄВОЇ УСПІШНОСТІ СТАРШОКЛАСНИКІВ." ББК 74.200 С 91 Рекомендовано до друку Вченою радою Інституту проблем виховання НАПН України (протокол № 5 від 25 травня 2023 року): 204.
3. Степанець, І., Гринюк, Д., & Савицька, Н. (2022). СОЦІАЛЬНІ МЕРЕЖІ ЯК ІНСТРУМЕНТ ЕФЕКТИВНОЇ КОМУНІКАЦІЇ МІЖ СТЕЙКХОЛДЕРАМИ ТУРИСТИЧНОЇ ІНДУСТРІЇ В КРИЗОВИЙ ПЕРІОД . Економіка та суспільство, (44). URL: <https://doi.org/10.32782/2524-0072/2022-44-36>
4. КОРОЛЬ Сергій Костянтинович. Вплив соціальних медіа на формування політичної культури сучасної української молоді. 2022. - 57 с.
5. Галушка Зоя. Вплив діджиталізації на нагромадження соціального капіталу економіки. International Science Journal of Management, Economics & Finance. Vol. 3, No. 2, 2024, pp. 1-10. doi: 10.46299/j.isjmef.20240302.01.
6. Гаврилюк, О. П. (2023). СОЦІАЛЬНІ МЕРЕЖІ ЯК ЧИННИК КУЛЬТУРНОЇ АДАПТАЦІЇ І СОЦІАЛІЗАЦІЇ. Культурологічний альманах, (2), 225–231.
7. Мостова, Анастасія. 2022. «ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ ПРОСУВАННЯ БІЗНЕСУ В СОЦІАЛЬНИХ МЕРЕЖАХ». Економіка та суспільство, вип. 43 (Вересень). <https://doi.org/10.32782/2524-0072/2022-43-27>.
8. Гарматій О. Соціальні мережі як джерело інформації в журналістиці / О. Гарматій, С. Онуфрив // Вісник Національного університету "Львівська політехніка". Журналістика. - 2021. - № 2. - С. 45-52. - Режим доступу: http://nbuv.gov.ua/UJRN/vnulpjn_2021_2_9.

9. Олейник, Анастасія. "Етичні феномени соціальних мереж: мова ворожнечі, кіберненависть та кібербулінг." Вісник Львівського університету.«Серія філософія-політологія (2022): 118-124.
10. Асєєва Ю. О. Кіберкомунікативна залежність від соціальної мережі Instagram як проблема сучасної молоді // Габітус. — 2020. — Вип. 14. — С. 152–157.
11. Сіткар Т. В. Розробка OSINT інструменту для пошуку профілів в соціальних мережах : кваліфікаційна робота на здобуття освітнього ступеня магістр за спеціальністю „125 — кібербезпека“ / Т. В. Сіткар. — Тернопіль: ТНТУ, 2023. — 68 с.
12. Teoh, T.T., Rong, Z. (2022). Python for Data Analysis. In: Artificial Intelligence with Python. Machine Learning: Foundations, Methodologies, and Applications. Springer, Singapore. https://doi.org/10.1007/978-981-16-8615-3_7
13. Зеленська Л. Д. Теоретичні основи застосування контент-аналізу в історико-педагогічних дослідженнях / Л. Д. Зеленська, А. Д. Балацинова // Наукові записки Бердянського державного педагогічного університету. Сер. : Педагогічні науки / Бердян. держ. пед. ун-т ; [редкол.: І. Т. Богданов (голов. ред.) та ін.]. — Бердянськ, 2020. — Вип. 1. — С. 29–42.
14. Fielding M. REST: Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation. University of California. Irvine, 2000. URL: https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (Дата звернення 25.10.2024).
15. Web scraping. Wikipedia. [Електронний ресурс] Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Web_scraping
16. Scrapy, загальна документація [Електронний ресурс]. — 2024. — Режим доступу до ресурсу: <https://docs.scrapy.org/en/latest/>
17. Pyhon [Електронний ресурс]. — 2024. — Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Python>

18. Лазоренко, А. Р. Повнотекстовий пошук інформації в системі Odoo : дипломна робота ... бакалавра : 122 Комп'ютерні науки / Лазоренко Анна Русланівна. – Київ, 2024. – 60 с.
19. JetBrains. Офіційний сайт. URL: <https://www.jetbrains.com/>
20. ЩО ТАКЕ МОВА PYTHON ТА ЯКІ СФЕРИ ЇЇ ЗАСТОСУВАННЯ. Sigma software. 2023. URI: <https://university.sigma.software/what-is-python/>.
21. Михайляк М. О., Січко Т. В. Використання Django Rest framework для створення API: переваги та можливості // Комп'ютерні технології обробки даних. — 2023. — Секція 2: Інтелектуальні системи та мережі.
22. НЕСКОРОДЄВА, А. Р.; ДАНИЛЬЧУК, О. М. Використання бібліотеки Matplotlib для візуалізації даних. Прикладні інформаційні технології, 2022, 115-119.
23. Федорчук, Ігор Ігорович. "Окреслення сигналів електрокардіограм з використанням автокодувальної нейромережі." (2023).

Дипломна робота на тему:

СИСТЕМА АНАЛІЗУ КОНТЕНТУ СОЦІАЛЬНИХ МЕРЕЖ НА ОСНОВІ ТЕЛЕГРАМ-БОТУ

Виконав:

студент групи КНДМ-62

Шпінь Вадим Анатолійович

Керівник роботи:

Прокопов Сергій Васильович



Слайд 1

Актуальність теми:



Зростання обсягів інформації в соціальних мережах створює виклики для дослідників у забезпеченні швидкості, якості та точності аналізу даних. Telegram як платформа пропонує унікальні можливості для автоматизації збору та аналізу даних через боти, що значно знижує витрати часу та ресурсів. Ця тема є особливо актуальною для аналізу інформаційних кампаній, виявлення трендів та моніторингу громадської думки.



Слайд 2

Мета роботи:

Створення телеграм-боту, який автоматично аналізує контент із соціальних мереж і надає користувачам інформацію про тренди, популярні публікації та статистику

Об'єкт дослідження:

Процеси аналізу контенту соціальних мереж

Предмет дослідження:

Телеграм-бот як інструмент автоматизації дослідження контенту соціальних мереж.

Слайд 3

Теоретичні основи

Теоретичні засади дослідження соціальних мереж базуються на комплексній сукупності принципів та моделей, які прагнуть пояснити механізми створення, поширення та сприйняття контенту в цифровому середовищі. Ці засади охоплюють широкий спектр дисциплін, зокрема соціологію, комунікативістику та психологію. Зокрема, соціальні теорії дозволяють зрозуміти, як суспільні структури, норми та взаємовідносини впливають на контент, що створюється та поширюється в мережі.

Слайд 4

Програмне забезпечення для розробки

Python



Python є однією з найпопулярніших мов програмування завдяки своїй простоті, читабельності та потужним можливостям. Вона забезпечує широкий спектр бібліотек і фреймворків для розробки різноманітних додатків, включаючи телеграм ботів, роботу з API, обробку даних та візуалізацію результатів.

Python є універсальним інструментом для розробки як невеликих скриптів, так і великих проектів.

Слайд 5

Програмне забезпечення для розробки

PyCharm — інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, графічний зневаджувач, інструмент для запуску юніт-тестів.

PyCharm надає користувачу інтегроване середовище розробки (ICP; англ. IDE — Integrated Development Environment) із такими корисними функціями, як підсвічування синтаксису та помилок, навігація серед проектів і в коді програми, відображення файлової структури проекту, швидкий перехід між файлами, класами, методами тощо.



Слайд 6

Бібліотеки

Для обробки тексту:

- 1.NLTK (Natural Language Toolkit)
- 2.SpaCy
- 3.Gensim

Для веб-розробки:

- 1.Django
- 2.Flask
- 3.FastAPI

Бібліотека **python-telegram-bot** дозволяє легко створювати ботів для Telegram за допомогою Python, забезпечуючи зручний інтерфейс для роботи з Telegram Bot API.

Вона підтримує всі основні функції, включаючи обробку повідомлень, медіафайлів, клавіатур і опитувань. Завдяки простоті використання, якісній документації та підтримці асинхронного програмування бібліотека забезпечує високу продуктивність і дозволяє створювати як прості, так і складні автоматизовані системи.

Слайд 7

Архітектура

При проектуванні архітектури Telegram-бота були виконані такі кроки:

- 1.Створення нового бота для аналізу контенту в соціальних мережах за допомогою **@BotFather**. Вибір для нього назви, оформлення інтерфейсу і додавання опису, який приверне увагу користувачів.
- 2.Імпорт необхідних бібліотек в код проекту.
- 3.Додавання API-токенів.
- 4.Розробка функції **start**.
- 5.Створення шаблонних кнопок для навігації та обробка їх дії.
- 6.Реалізація функції для отримання трендів YouTube і пошуку твітів за заданими параметрами.
- 7.Обробка отриманих результатів, формуючи текст відповіді.
- 8.Створення основної функції для запуску бота.

Слайд 8



1. Створення нового бота для аналізу контенту в соціальних мережах за допомогою [@BotFather](#). Вибір для нього назви, оформлення інтерфейсу і додавання опису, який приверне увагу користувачів.

2. Імпорт необхідних бібліотек в код проекту.

```
1 from telegram import Update, ReplyKeyboardMarkup
2 from telegram.ext import Application, CommandHandler, MessageHandler, filters
3 import requests
4 import random
```

```
5
6 TELEGRAM_TOKEN = "313863236:AAHPrR3Q5o0y8e4_A2XN47_dX-dJmzu9s4"
7 YOUTUBE_API_KEY = "AIzaSyB6hP5F8nyYH0M0n43p9n0yFpuYdCLW8"
8 TWITTER_BEARER_TOKEN = "AAAAAAAAAAAAAAAAAGLxvGEAAAAAbCMFaaQMKB7LWc1003kxu1aebYn30bK4P3B38jKmds9vmqNzP23fKxW0A3J"
9
10
11
```

3. Додавання API-токенів.

4. Розробка функції `start`.

```
11
12 async def start(update: Update):
13
```

Слайд 9

```
14 keyboards = [{"text": "YouTube", "x": 1, "y": 1}, {"text": "Twitter", "x": 2, "y": 1}, {"text": "App Store", "x": 3, "y": 1}, {"text": "Telegram Data", "x": 4, "y": 1}]
15 reply_markup = ReplyKeyboardMarkup(keyboards, resize_keyboard=True)
16
17 await update.message.reply_text(
18     "Привіт! Я бот для аналізу соціальних мереж. Обери для мене меню:",
19     reply_markup=reply_markup
20 )
21
22 async def handle_menu(update: Update):
23     text = update.message.text
24
25     if text == "YouTube":
26         result = await get_youtube_trends()
27         await update.message.reply_text(result)
28     elif text == "Twitter":
29         result = await get_twitter_tweets()
30         await update.message.reply_text(result)
31     elif text == "App Store":
32         result = "Я бот для аналізу соціальних мереж. Обери для мене меню:"
33         await update.message.reply_text(result)
34     elif text == "Telegram Data":
35         result = "Я бот для аналізу соціальних мереж. Обери для мене меню:"
36         await update.message.reply_text(result)
37     else:
38         result = "Невідома команда. Обери для мене меню:"
39         await update.message.reply_text(result)
```

5. Створення шаблонних кнопок для навігації та обробка їх дії.

6. Реалізація функції для отримання трендів YouTube і пошуку твітів за заданими параметрами

```
40 async def get_youtube_trends():
41     """usage"""
42     URL = "https://www.googleapis.com/youtube/v3/videos"
43
44     params = {
45         "part": "snippet,statistics",
46         "chart": "mostPopular",
47         "regionCode": "UA",
48         "maxResults": 5,
49         "key": YOUTUBE_API_KEY,
50         "random": random.randint(1, 1000000)
51     }
```

```
10 async def get_youtube_trends():
11     """usage"""
12     title = video["snippet"]["title"]
13     channel = video["snippet"]["channelTitle"]
14     views = video["statistics"]["viewCount"]
15     video_id = video["id"]
16     video_url = f"https://www.youtube.com/watch?v={video_id}"
17     reply_text = (
18         f"■ Title: {title}\n"
19         f"■ Channel: {channel}\n"
20         f"■ Views: {views}\n"
21         f"■ URL: {video_url}\n"
22     )
23     return reply_text
24
25 if __name__ == "__main__":
26     (response.status_code, response.text)
27
28 async def get_twitter_tweets():
29     """usage"""
30     URL = "https://api.twitter.com/2/tweets/search/recent"
31     headers = {
32         "Authorization": f"Bearer {TWITTER_BEARER_TOKEN}"
33     }
34     params = {
35         "query": "python",
36         "max_results": 10,
37     }
```

Слайд 10

7. Обробка отриманих результатів, формуючи текст відповіді.

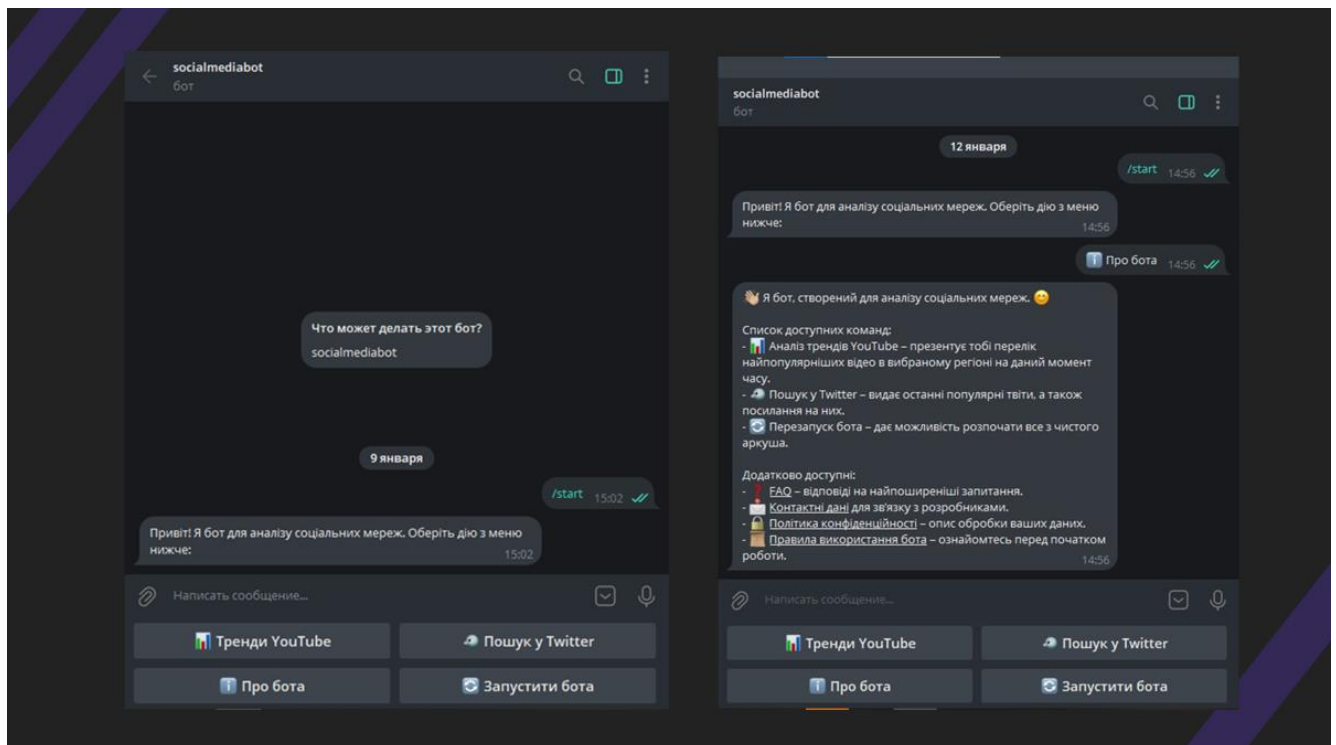
```
70 async def get_youtube_trends(): usage
71
72
73
74 response = requests.get(URL, params=params)
75
76
77 if response.status_code == 200:
78     data = response.json()
79
80     reply_text = "Топові відео на YouTube:\n\n"
81     for video in data['items']:
82         title = video['snippet']['title']
83         channel = video['snippet']['channelTitle']
84         views = video['statistics']['viewCount']
85         video_id = video['id']
86         video_url = f"https://www.youtube.com/watch?v={video_id}"
87         reply_text += (
88             f"📺 Назва: {title}\n"
89             f"👤 Канал: {channel}\n"
90             f"👁️ Перегляди: {views}\n"
91             f"🔗 Посилання: {video_url}\n\n"
92         )
93     return reply_text
94 else:
95     return f"❌ Помилка: {response.status_code}. {response.text}"
```

```
70 async def get_twitter_tweets(): usage
71
72
73
74 params = {
75     'query': 'москва',
76     'max_results': 10,
77 }
78
79 response = requests.get(URL, headers=headers, params=params)
80
81
82 if response.status_code == 200:
83     data = response.json()
84
85     reply_text = "Останні популярні твіти:\n\n"
86     for tweet in data['data']:
87         text = tweet['text']
88         reply_text += f"🐦 {text}\n\n"
89     return reply_text
90 elif response.status_code == 401:
91     return f"❌ Помилка: Недійсний Bearer Token. Перезапустіть бота у Telegram."
92 elif response.status_code == 403:
93     return f"❌ Помилка: Доступ заборонено. Можливо, ваш обліковий запис не має прав доступу до цього ресурсу."
94 else:
95     return f"❌ Помилка: Не вдалося отримати дані з Twitter API. Код помилки: {response.status_code}. Деталі: {response.text}"
```

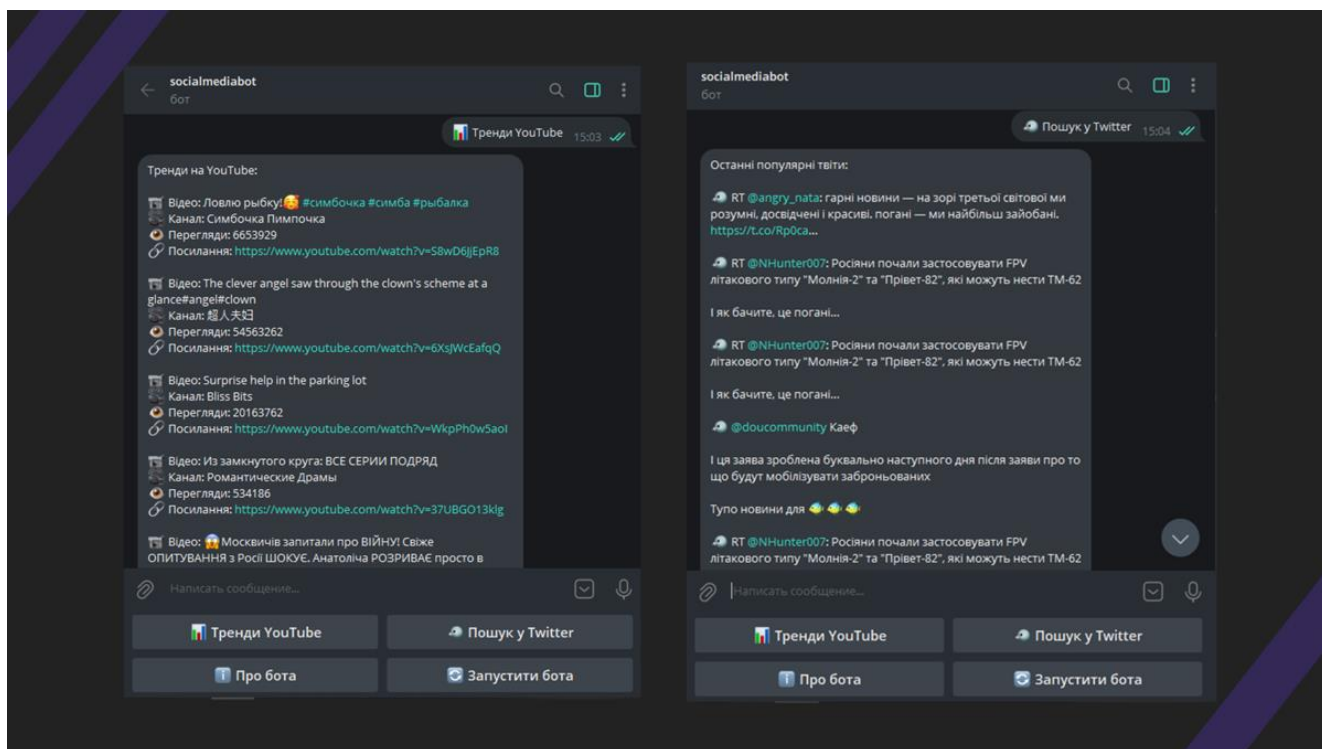
8. Створення основної функції для запуску бота.

```
107
108 def main(): usage
109
110
111 application = Application.builder().token(TELEGRAM_TOKEN).build()
112
113 application.add_handler(CommandHandler('start', start))
114 application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND, handle_menu))
115
116
117 application.run_polling()
118
119 if __name__ == '__main__':
120     main()
121
```

Слайд 11



Слайд 12



Слайд 13

Висновки

У ході виконання роботи було створено Telegram-бота, здатного аналізувати контент соціальних мереж, надаючи користувачам актуальну інформацію у зручному форматі.

Реалізовані функції, такі як отримання трендів YouTube та пошук твітів, демонструють потенціал автоматизації збору та обробки даних у сучасному цифровому середовищі.

Проект поєднує ефективність Telegram API, можливості аналізу даних і зручність користувацького інтерфейсу, що робить його корисним інструментом для моніторингу популярних тем і трендів.

Ця робота не лише підкреслює важливість інтеграції технологій у повсякденні завдання, але й відкриває перспективи для подальшого розвитку, таких як додавання нових платформ, удосконалення аналізу настроїв і створення більш складних алгоритмів обробки даних.

Результати підтверджують актуальність і практичну цінність використання Telegram-ботів у сфері аналізу контенту соціальних мереж.

Слайд 14

The slide has a dark gray background. In the center is a large, dark gray circle containing a lighter gray play button icon. The text "ДЯКУЮ ЗА УВАГУ!" is written in a bold, white, italicized sans-serif font across the middle of the circle. There are purple decorative elements: a triangle in the top-left corner, a triangle in the bottom-right corner, and two vertical columns of three small squares each, one on the left and one on the right.

ДЯКУЮ ЗА УВАГУ!

Слайд 15