

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки**

«На правах рукопису»
УДК 004.4

До захисту допущено:
В.О. Завідувача кафедри
_____ Михайло НОВОТАРСЬКИЙ
«__»_____ 2024 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою «Комп'ютерні системи та мережі»
зі спеціальності 123 «Комп'ютерна інженерія»
на тему: «Спосіб автоматичної генерації контенту в соцмережах»**

Виконав:

студент II курсу, групи ІО-31мп
Кучеренко Іван Дмитрович _____

Науковий керівник:

Доцент каф. ОТ, к.т.н.
Русанова Ольга Веніамінівна _____

Консультант:

Проф. каф. ОТ д.т.н.
Кулаков Юрій Олексійович _____

Рецензент:

Доцент кафедри ІСТ к.т.н.
Шимкович Володимир Миколайович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2024 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Інформатики та обчислювальної техніки
(повна назва)

Кафедра Обчислювальної техніки
(повна назва)

Рівень вищої освіти – другий (магістерський)

Спеціальність 123. Комп'ютерна інженерія
(код і назва)

Освітньо-професійна програма Комп'ютерні системи та мережі
(код і назва)

ЗАТВЕРДЖУЮ

В.О. Завідувача кафедри

Михайло НОВАТОРСЬКИЙ
(підпис)

« ____ » _____ 2024 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Кучеренку Івану Дмитровичу

(прізвище, ім'я, по батькові)

1. Тема дисертації «Спосіб автоматичної генерації контенту в соцмержах», науковий керівник дисертації доцент каф. ОТ, к.т.н. Русанова О.В., затверджені наказом по університету від «07» листопада 2024 р. № 50016-с
2. Строк подання студентом дисертації _____
3. Об'єкт дослідження автоматизація збору даних, генерації публікацій та постингу
4. Предмет дослідження алгоритми парсингу інтернет-сторінок і методи генерації контенту та постингу, які використовуються для автоматичної генерації та постингу контенту.
5. Перелік завдань, які потрібно розробити: провести аналіз існуючих рішень для автоматичного парсингу інформації, генерації публікацій та алгоритмів для автоматичної публікації контенту в соцмережах, розробити оптимізований алгоритм генерації публікацій, розширити алгоритм парсингу для певних інтернет-сторінок та покращити алгоритм постингу, створити прототип системи на базі оптимізованих алгоритмів, провести аналіз результатів та оцінити ефективність роботи прототипу, описати рекомендації для подальшого використання та масштабування системи.

6. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1.	<u>доцент каф. ОТ, к.т.н. Русанова О.В.</u>		
2.	<u>доцент каф. ОТ, к.т.н. Русанова О.В.</u>		
3.	<u>доцент каф. ОТ, к.т.н. Русанова О.В.</u>		
4.	<u>доцент каф. ОТ, к.т.н. Русанова О.В.</u>		
5.	<u>доцент каф. ОТ, к.т.н. Русанова О.В.</u>		

7. Дата видачі завдання 01.09.2024

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Затвердження теми роботи	1.09.24.-5.09.24	
2	Вивчення та аналіз завдання	6.09.24-21.09.24	
3	Дослідження існуючих рішень	22.09.24-28.09.24	
4	Адаптація обраних алгоритмів до розроблюваної системи	29.09.24-05.10.24	
5	Розробка архітектури та загальної структури програми	06.10.24-10.10.24	
6	Програмна реалізація	10.10.24-19.10.24	
7	Тестування рішення та аналіз ефективності.	10.11.24-13.11.24	
8	Оформлення пояснювальної записки	18.11.24-25.11.24	
9	Захист	18.12.2024	

Студент

Іван КУЧЕРЕНКО

(підпис)

Науковий керівник дисертації

Ольга РУСАНОВА

РЕФЕРАТ

на магістерську дисертацію

виконану на тему: Спосіб автоматичної генерації контенту в соцмережах

студентом: Кучеренко Іваном Дмитровичем

Дана робота містить вступ та п'ять розділів.

Актуальність теми. Актуальність теми дисертації зумовлена необхідністю впровадження автоматизованих систем для генерації та публікації контенту в соціальних мережах, таких як X. У сучасних умовах стрімкого розвитку цифрового маркетингу та зростання важливості онлайн-присутності бізнесів, автоматизація цих процесів стає все більш актуальною. Рішення, які дозволяють швидко збирати інформацію з різних джерел, генерувати контент та планувати його публікацію, є незамінними для ефективного управління соціальними мережами.

Крім того, у світі, де конкуренція на ринку онлайн-просування стає дедалі жорсткішою, автоматизовані системи можуть значно зменшити витрати часу на рутинні операції та підвищити точність і релевантність контенту. Такі системи, як ваша розробка, дозволяють користувачам створювати публікації за ключовими словами та планувати їх публікацію, що відкриває нові можливості для бізнесів з обмеженими ресурсами або малими командами.

Мета і задачі дослідження. Метою даного дослідження є підвищення ефективності процесу генерації та публікації контенту в соціальних мережах шляхом покращення автоматизованої системи збору даних та планування публікацій.

Задачі дослідження:

1. Проаналізувати існуючі рішення для автоматизованого збору та публікації контенту в соціальних мережах
2. Розробити стратегію інтеграції парсингу контенту з кількох джерел та генерації публікацій на основі ключових слів

3. Створити прототип системи, що забезпечує автоматизацію збору, форматування та публікації контенту в X
4. Провести тестування прототипу, оцінити його ефективність та можливості оптимізації
5. Сформулювати рекомендації щодо подальшого розвитку та вдосконалення системи

Об'єкт дослідження. Об'єктом дослідження є процес автоматизації генерації та публікації контенту в соціальних мережах.

Предмет дослідження. Предметом дослідження є алгоритми та методи, що використовуються для збору інформації з інтернет-сайтів, виконують обробку отриманої інформації, генерують публікації за ключовими словами та автоматичного планують пости у X.

Методи дослідження. Дослідження використовує різні аналітичні методи, включаючи аналіз, синтез, моделювання та тестування, щоб ретельно вивчити процес автоматизації публікацій. Це дозволяє розробити та оцінити інноваційні методи інтеграції аналізу даних і генерації контенту в соціальних мережах.

Наукова новизна одержаних результатів. Результат цього дослідження демонструє новий метод автоматизації створення та розповсюдження контенту в соціальних мережах. Дана система поєднує в собі аналіз джерел новин, створення публікацій за певними ключовими словами та автоматичне планування твітів. Використання такої комбінації методів дозволяє досягти високого рівня автоматизації без необхідності ручного втручання, що раніше не було представлено у подібних системах.

Практичне значення одержаних результатів. Результати дослідження можуть бути застосовані для автоматизованого управління соціальними мережами та маркетинговою діяльністю малих і середніх підприємств. Дана система допомагає заощадити години ручної роботи та підвищує ефективність контент-стратегій. Система також рекомендована для впровадження компаніям, які прагнуть оптимізувати свої онлайн-процеси просування.

Особистий внесок магістранта. Магістерська робота - це самостійний проект, який демонструє унікальну перспективу та активну участь автора в дослідницькому процесі.

У рамках дослідження була створена комп'ютерна програма для збору, аналізу та обміну інформацією в X, використовуючи передові методи розуміння та створення тексту. Автор модифікував існуючі методи відповідно до вимог завдання, забезпечивши практичні результати в сфері автоматизації контенту для платформ соціальних мереж. Основна ідея, мета і завдання роботи розроблені у співпраці з науковим керівником.

Ключові слова: автоматизація контенту, соціальні мережі, парсинг даних, генерація публікацій, X.

ABSTRACT

for the master's thesis

completed on the topic: Bot for automatic content generation in social networks

by student: Ivan Dmytrovych Kucherenko

This paper contains an introduction and five chapters.

Relevance of the topic. The relevance of the dissertation topic is due to the need to implement automated systems for generating and publishing content on social networks such as X. In today's context of rapid development of digital marketing and the growing importance of the online presence of businesses, automation of these processes is becoming increasingly important. Solutions that allow you to quickly collect information from various sources, generate content and schedule its publication are indispensable for effective social media management.

In addition, in a world where competition in the online promotion market is becoming increasingly fierce, automated systems can significantly reduce time spent on routine operations and increase the accuracy and relevance of content. Systems such as yours allow users to create keyword-based posts and schedule them for publication, which opens up new opportunities for businesses with limited resources or small teams.

Purpose and objectives of the study. The purpose of this study is to increase the efficiency of the process of generating and publishing content on social networks by improving the automated system of data collection and publication planning.

Research objectives:

1. Analyze existing solutions for automated collection and publication of content in social networks
2. Develop a strategy for integrating content parsing from multiple sources and generating keyword-based publications
3. Create a prototype of a system that automates the collection, formatting and publication of content on X
4. To test the prototype, evaluate its effectiveness and optimization possibilities
5. Formulate recommendations for further development and improvement of the system

Object of research. The object of research is the process of automating the generation and publication of content in social networks.

The subject of research. The subject of the study is algorithms and methods used to collect information from websites, process the information received, generate publications by keywords and automatically schedule posts on X.

Research methods. The study uses various analytical methods, including analysis, synthesis, modeling, and testing, to thoroughly examine the process of automating publications. This allows us to develop and evaluate innovative methods for integrating data analysis and content generation on social media.

Scientific novelty of the results. The result of this study demonstrates a new method for automating the creation and distribution of content on social media. This system combines the analysis of news sources, the creation of posts based on certain keywords, and the automatic scheduling of tweets. The use of this combination of methods allows to achieve a high level of automation without the need for manual intervention, which has not been previously presented in similar systems.

Practical significance of the results. The results of the study can be used for automated management of social networks and marketing activities of small and medium-sized enterprises. This system helps to save hours of manual work and increases the effectiveness of content strategies. The system is also recommended for implementation by companies seeking to optimize their online promotion processes.

Personal contribution of the master's student. The master's thesis is an independent project that demonstrates the author's unique perspective and active participation in the research process. As part of the research, a computer program was created to collect, analyze and share information on X using advanced methods of understanding and creating text. The author modified existing methods in accordance with the requirements of the task, providing practical results in the field of content automation for social media platforms. The main idea, goal and objectives of the work were developed in collaboration with the supervisor.

Key words: content automation, social networks, data parsing, post generation,

X.

Пояснювальна записка до магістерської дисертації

на тему: «Спосіб автоматичної генерації контенту в
соцмережах»

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	13
ВСТУП	13
РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ НАУКОВОГО, ІННОВАЦІЙНОГО ТА ПРАКТИЧНОГО ЗДОБУТКУ В ОБЛАСТІ АВТОМАТИЗАЦІЇ ПУБЛІКАЦІЙ В СОЦІАЛЬНИХ МЕРЕЖАХ.	16
1.1 Проблеми автоматизації публікацій у соціальних мережах	16
1.2 Інноваційні досягнення у сфері автоматизації публікацій у соціальних мережах	18
1.3 Огляд існуючих технологій та систем	24
1.4 Невирішені проблеми та можливі напрямки для подальшого дослідження	29
1.5 Постановка задачі	30
ВИСНОВКИ ДО РОЗДІЛУ 1	35
РОЗДІЛ 2. МЕТОДОЛОГІЯ ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДІВ	36
2.1 Загальна методика процесу дослідження	36
2.2 Опис методів та засобів	38
2.3 Оцінка методів автоматизованого постингу та парсингу	41
2.4 Поєднання обраних методів між собою	46
ВИСНОВКИ ДО РОЗДІЛУ 2	50
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ	51
3.1 Розробка моделі системи	51
3.2 Безпосередня розробка застосунку	53

	12
ВИСНОВКИ ДО РОЗДІЛУ 3	62
РОЗДІЛ 4. ТЕСТУВАННЯ РОЗРОБКИ.....	76
4.1 Створення алгоритму тестування	76
4.2 Тестування застосунку та його аналогів	77
4.3 Огляд користувацького інтерфейсу розробки	79
4.4 Оцінка результатів експериментів тестування	82
4.5 Стратегія вдосконалення та рекомендації по використанню.....	83
ВИСНОВОК ДО РОЗДІЛУ 4	88
РОЗДІЛ 5. РОЗРОБКА СТАРТАП-ПРОЕКТУ	89
5.1 Інформаційна картка проекту.....	90
5.2 Склад команди стартап-проекту	93
5.3 Про основну ідею та мету проекту	95
5.4 Аналіз технологій розробки.....	97
5.5 Аналіз реальних шансів випуску продукту на ринок розробок.....	99
5.6 Написання виробничого плану.....	102
ВИСНОВОК ДО РОЗДІЛУ 5	105
ВИСНОВКИ	106
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	107

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- IFTTT (If This Then That) – Інструмент автоматизації
- APP (application software) - Користувачка комп'ютерна програма
- X (X) - Соціальна мережа
- Постинг - Процес написання і розміщення постів у соціальних мережах або блогах
- Парсер - Метод швидкої обробки інформації, розміщеної на веб-сторінках
- ПЗ (Програмне забезпечення) – Software
- ПП (Програмний продукт) - Software Product
- DOM-структури (Document Object Model) - Специфікація прикладного програмного інтерфейсу для роботи зі структурованими документами
- HTML (HyperText Markup Language) - Стандартизована мова розмітки документів для перегляду веб-сторінок у браузері
- XPath (XML Path Language) - Мова запитів, для вибору вузлів з XML документів, або для обчислення величин
- XML (EXtensible Markup Language) - Розширювана мова розмітки
- CSS (Cascading Style Sheets) - Спеціальна мова стилю сторінок
- API (Application Programming Interface) - Прикладний програмний інтерфейс
- JSON (JavaScript Object Notation) - Формат обміну даними
- POST-запит – запит для відправки даних на сервер
- JSON (JavaScript Object Notation) - Формат обміну даними
- IDE (Integrated Development Environment) - Інтегроване середовище розробки
- URL (Uniform Resource Locator) - Єдиний локатор ресурсів
- RAM (Random Access Memory) - Оперативна пам'ять
- CPU (Central Processing Unit) - Центральний процесор
- GPU (Graphics Processing Unit) - Графічний процесор
- AI (Artificial Intelligence) - Штучний інтелект
- DB (Data Base) - База даних

ВСТУП

В сьогоднішніх умовах стрімкого розвитку цифрового маркетингу та швидкого зростання необхідності онлайн-присутності бізнесів, автоматизація таких процесів набирає все більшу популярність. Такі рішення, які дають змогу оперативно збирати інформацію з різних джерел, створювати унікальний контент та розміщувати публікації з його вмістом за певним графіком автоматично, є досить актуальним для вдалого управління соціальними мережами. Тому представлена магістерська робота спрямована на дослідження та розробку технології, яка буде працювати як вдосконалений та оптимізований процес автоматичного пошуку, генерації та публікації контенту на сторінках в соціальних мережах. Мета такого дослідження це покращення ефективності роботи автоматизованої програми, шляхом вдосконалення методів та алгоритмів генерації та публікації публікацій та написання унікального парсингу інтернет сторінок, що має на меті підвищення якості та оптимізації роботи маркетингу та процесів розвитку бізнесу.

Перший розділ представляє собою докладний огляд та аналіз інноваційного, наукового та практичного здобутку в сфері автоматизації процесів для розвитку бізнесу. Крім того, було проаналізовано існуючі рішення, їх специфіку, переваги та обмеження та можливі проблеми.

У другому розділі розглянуто методи дослідження та пояснено вибір інструментів для розробки системи, яка автоматизує процес ведення бізнес-сторінок або новинних каналів з метою їх просування.

В третьому розділі описано процес планування та розробки описуваної системи, а саме, визначено вимоги до майбутньої розробки, модернізовано алгоритми та створено архітектуру системи.

Четвертий розділ це процес тестування реалізованого застосунку, також була проведена оцінка ефективності роботи програми та точність роботи на основі тестових даних.

У кінцевому п'ятому розділі спроектовано розробку стартап-проекту, дані для якого взято з попередніх розділів, результатів тестування та інформації про саму розробку.

Дана розробка містить не тільки теоретичний опис для вирішення проблематики завдання, але й має практичну реалізацію розглянутої теми, яка надалі може стати повноцінним корисним продуктом в сфері бізнесу та маркетингу.

Як результат даної магістерської роботи це створений застосунок, оптимізований для пошуку релевантної інформації, генерації унікального контенту та автоматизованої публікації отриманих даних, що дозволить мінімізувати втручання людини в процес ведення бізнес-сторінок або новинних сайтів, починаючи з пошуку інформації. Це сприятиме більш ефективному використанню соціальних мереж, оптимізує роботу з контентом та виключатиме людський фактор.

РОЗДІЛ 1

ОГЛЯД ТА АНАЛІЗ НАУКОВОГО, ІННОВАЦІЙНОГО ТА ПРАКТИЧНОГО ЗДОБУТКУ В ОБЛАСТІ АВТОМАТИЗАЦІЇ ПУБЛІКАЦІЙ В СОЦІАЛЬНИХ МЕРЕЖАХ.

1.1. Проблеми автоматизації публікацій у соціальних мережах

1.1.1. Проблематика у сфері автоматизації

Автоматизація публікацій у соціальних мережах, наприклад у X, стає все більш актуальною в сучасному світі цифрових комунікацій. Зростання обсягу інформації та швидкості її розповсюдження вимагають від світу впровадження нових рішень для ефективного управління контентом. Проблема полягає в тому, що звичайні методи публікації зазвичай не враховують унікальні вимоги користувачів та специфіку їхньої діяльності. Часто користувачі, наприклад малі підприємства, стикаються з труднощами, коли справа доходить до організації та створення високоякісного контенту. Відсутність автоматизованих рішень у цій сфері призводить до неефективного використання часу та ресурсів. Користувачі змушені витратити значні зусилля на створення публікацій, що часто заважає їм зосередитися на інших аспектах їхньої діяльності. Сучасні інструменти для автоматизації постингу, загалом, мають обмежену функціональність і не можуть запропонувати необхідний рівень налаштування для задоволення унікальних вимог користувачів. Часто вони не враховують широкий спектр і форматів контенту, що призводить до одноманітності публікацій і, зрештою, до зменшення їх ефективності. Отже, існує безпосередня потреба у створенні передових рішень для автоматизації публікацій на сторінках в соціальних мережах, які дозволятимуть користувачам легко та швидко створювати якісний контент, адаптуючи його до своїх потреб та специфіки діяльності.

1.1.2. Потенціал автоматизації процесу публікації на платформах соціальних мереж

Автоматизація публікацій на платформах соціальних медіа, особливо X, відкриває численні можливості для підвищення ефективності цифрового маркетингу та управління контентом. Постійний прогрес у технологіях постійно вдосконалює існуючі інструменти, надаючи користувачам більш ефективні, гнучкі та результативні способи створення та планування публікацій.

1. Індивідуальний підхід: Системи автоматизації публікацій можуть бути налаштовані відповідно до індивідуальних вимог кожного користувача, враховуючи особливості бізнесу, цільову аудиторію та ключові слова, які використовуються в маркетингових цілях. Пристосовуючи контент до конкретних інтересів аудиторії, можна створити персоналізований контент, який резонуватиме з інтересами читачів та підвищить успіх маркетингових кампаній.
2. Доступність: Автоматизацію вмісту можна легко інтегрувати у веб-додатки та мобільні платформи, забезпечуючи доступність для широкого кола користувачів. Ці рішення не вимагають великого технічного досвіду чи спеціалізованого обладнання, що робить технології автоматизації доступними як для малого бізнесу, так і для особистих брендів.
3. У режимі реального часу: передові автоматизовані системи дозволяють швидко створювати та поширювати контент у режимі реального часу. Це особливо важливо для своєчасного реагування на поточні події чи зміни ринку, що дозволяє користувачам залишатися на зв'язку з аудиторією та підтримувати релевантність в інформаційному просторі.

1.1.3. Необхідність нових досліджень

Основною метою майбутніх досліджень автоматизації публікацій є усунення поточних обмежень існуючих систем і підвищення їх ефективності для користувачів із різними вимогами. Серед напрямків, які потребують додаткової уваги, слід виділити наступні:

1. Точність автоматизації: головною метою має бути створення систем, які можуть виробляти більш точний і релевантний контент, враховуючи специфіку ключових слів і тем. Це гарантує, що автоматично згенеровані

публікації більш точно відповідають уподобанням і очікуванням цільової аудиторії.

2. Інтерфейс користувача: надзвичайно важливо розробити інтерфейс, який буде простим у використанні та навігації, щоб користувачі різного рівня кваліфікації могли легко з ним взаємодіяти. Це підвищить зручність і ефективність розробки для різних груп користувачів.
3. Безпека даних: зважаючи на важливість конфіденційності, одним із найважливіших аспектів є саме захист наших персональних та інформації про бізнес-користувачів. Для забезпечення безпеки систем автоматизації вкрай важливо мати надійні механізми зберігання та обробки даних, які запобігають несанкціонованому доступу.
4. Гнучкість системи: гнучкість системи має вирішальне значення, оскільки вона повинна бути здатна враховувати різні категорії користувачів і типи вмісту, гарантуючи універсальну доступність. Це дозволить використовувати автоматизацію для різних бізнес-моделей і маркетингових стратегій.
5. Доступність: необхідно гарантувати, що системи автоматизації є фінансово прийнятними для широкого кола користувачів, особливо малих підприємств і незалежних підприємців. Це посилить впровадження та вплив стратегій цифрового маркетингу, що призведе до підвищення загальної ефективності.

Отже, подальші дослідження у сфері автоматизації публікацій повинні сприяти вдосконаленню технологій з метою забезпечення більшої точності, зручності та економічної ефективності для різних категорій користувачів.

1.2. Інноваційні досягнення у сфері автоматизації публікацій у соціальних мережах

1.2.1. Відкриття нових можливостей для маркетологів та бізнесів

Удосконалення технологій для автоматизації публікацій на платформах соціальних мереж відкрило нові можливості для маркетологів і підприємців, дозволяючи їм оптимізувати створення та розповсюдження контенту.

Завдяки передовим технологіям, які використовують штучний інтелект і алгоритми машинного навчання, маркетологи отримують доступ до потужних інструментів для точного планування та створення персоналізованого контенту. Ці системи можуть досліджувати поточні тенденції, вибирати найбільш прийнятні ключові слова та пропонувати пропозиції щодо найкращого часу для публікації, що покращує залучення аудиторії та успіх маркетингових кампаній.[2]

Автоматизовані системи публікацій також забезпечують послідовність та стабільність контенту на різних платформах. Це дозволяє мінімізувати людський фактор у створенні й плануванні постів, гарантуючи, що публікації відповідають заданим вимогам бренду і регулярності виходу. Крім того, автоматизація прискорює процес аналізу ефективності публікацій, дозволяючи маркетологам отримувати миттєвий зворотний зв'язок і вносити необхідні корективи у свої стратегії на основі зібраних даних.

Завдяки останнім досягненням у технологіях автоматизації, компанії можуть ефективніше взаємодіяти з аудиторією на різних платформах, заощаджуючи час і ресурси. Це особливо корисно для малих і середніх підприємств, які можуть не мати великих команд для ручного керування соціальними мережами.

1.2.2. Парсинг кількох джерел новин

1. Важливість багатоджерельного парсингу.

Парсинг різноманітних інформаційних видань є вирішальним кроком у сфері автоматизації публікацій. Агрегація вмісту з багатьох джерел новин дозволяє системам збирати відповідну інформацію, яку потім можна використовувати в автоматизованих публікаціях. Це дає змогу отримувати широкий спектр матеріалів, відслідковувати основні тенденції та розробляти більш різноманітний та цікавий контент для цільової аудиторії.

2. Технологічні можливості.

Сучасні системи автоматизації використовують методи веб-аналізу для збору заголовків, зображень, описів і посилань з різних веб-сайтів. Ці системи можна адаптувати до певних категорій новин, дозволяючи збирати лише ті матеріали, які відповідають вибраним темам. Використовуючи алгоритми синтаксичного аналізу, системи можуть упорядковувати зібрані дані в більш керований формат, полегшуючи їх використання для публікацій у соціальних мережах

3. Практичне використання.

Парсинг різноманітних новинних видань є важливою навичкою для компаній і маркетологів, які створюють контент, пов'язаний з поточними подіями. Наприклад, автоматизовані системи можуть збирати новини з популярних веб-сайтів і автоматично створювати пости для соціальних мереж з урахуванням запланованих часових рамок і вподобань аудиторії. Це дає вам змогу бути в курсі останніх новин і реагувати на них швидше, ніж конкуренти.

4. Перешкоди та можливості.

Отримання новин із різних джерел стикається з труднощами через необхідність налаштування систем відповідно до унікальних характеристик кожного сайту та способу подання інформації. Також важливо вирішити питання дотримання авторських прав на контент, що вимагає впровадження додаткових механізмів перевірки зібраних даних. Тим не менш, розвиток технологій синтаксичного аналізу дозволить створювати більш універсальні рішення для збору інформації з різних платформ, забезпечуючи тим самим ще більшу гнучкість у створенні та публікації контенту.

1.2.3. Генерація публікацій за ключовими словами

Генерація дописів за ключовими словами – це передовий метод створення вмісту, який дозволяє користувачам отримувати дописи, адаптовані до їхніх конкретних інтересів або сфер діяльності. Використовуючи ШІ та метод обробки природної мови, розробка може автоматично генерувати тексти, які відповідають вибраним ключовим словам і темам.

Технічні аспекти цього підходу передбачають:

- Аналіз ключових слів: алгоритми обробляють ключові слова, введені користувачем, визначаючи основні теми та генеруючи на їх основі контент.
- Генерація тексту: моделі, які використовують штучний інтелект, використовуються для створення текстів, які відповідають структурі та стилю вибраної теми.
- Оптимізація вмісту: система точно налаштована для доставки високорелевантного вмісту обраній аудиторії, забезпечуючи максимальний вплив на соціальні мережі чи інші публічні платформи.

Практичне застосування:

- Маркетинг: процес автоматичної генерації публікацій у соціальних мережах, який сприяє активно знаходити підприємствам та залучати клієнтів, створюючи контент, який відповідає їхнім потребам і вподобанням.
- Новини та інформаційні майданчики: створення публікацій на певну тематику, що сприяє оперативному висвітленню інформаційних подій.
- Блоги та інші ЗМІ: оптимізація процесу генерації текстів для видань, які часто оновлюють свій контент.

Основною перевагою цього методу є ефективність створення контенту та його відповідність з ключовими словами, що дозволяє користувачам економити час на написання статей.

1.2.4. Автоматичне розміщення публікацій

Одним із останніх досягнень у сфері керування вмістом є функція автоматичного опублікування, яка дає змогу оптимізувати процес обміну вмістом на різних платформах, включаючи соціальні мережі, блоги та веб-сайти новин. Автоматизовані системи публікацій легко підключаються до API різних платформ, дозволяючи вам планувати публікації заздалегідь і гарантувати їх своєчасну публікацію без постійної участі користувача.

Переваги автоматичного постингу:

- **Ефективність:** можливість планувати та планувати публікації заздалегідь, щоб забезпечити своєчасну публікацію.
- **Постійність:** автоматичне розміщення гарантує регулярне оновлення контенту, що позитивно впливає на активність і залучення аудиторії.
- **Оптимізація часу:** системи дозволяють обходити публікування контенту вручну, мінімізуючи ймовірність помилок і заощаджуючи час для користувачів, особливо при роботі зі значним обсягом вмісту.

Автоматичні системи постингу стають незамінними інструментами для контент-маркетологів, блогерів та компаній, що працюють у сфері медіа, забезпечуючи зручне та ефективне керування публікаціями в соціальних мережах і на різних платформах.

1.2.5. Перспективи розвитку технологій автоматичної генерації та постингу публікацій

Розвиток технологій автоматичної генерації та розміщення публікацій відкриває нові можливості у сфері цифрового контенту. Завдяки використанню штучного інтелекту, машинного навчання та обробки природної мови майбутні системи можуть стати ще потужнішими, гнучкішими та персоналізованими для користувачів.

Основні напрями розвитку охоплюють:

- **Персоналізація контенту:** У майбутньому технології автоматичної генерації зможуть ще точніше враховувати інтереси користувачів та їх аудиторії. Завдяки впровадженню цієї зміни стане можливим створювати ще більше адаптованих і персоналізованих публікацій, що посилить їхній вплив і залучення читачів.
- **Інтеграція з аналітичними системами:** Наступні етапи розробки – інтеграція автоматизованих генераторів публікацій з аналітичними системами. Аналізуючи минулі дані та ринкові тенденції, можна буде передбачити вплив публікацій і змінити контент відповідно до вподобань і реакції аудиторії.

- Розширення платформ: З удосконаленням технологій автоматичного розміщення публікацій платформи зможуть розширити свій список охоплення до нових соціальних мереж, блогів, сайтів з джерелами новин та схожих платформ обміну інформацією, що призведе до більш ефективного розповсюдження контенту.
- Покращення мовних моделей: З розвитком обробки природної мови, алгоритми генерування текстів стануть ще більш точними, здатними створювати контент, який важко відрізнити від текстів, написаних людиною. Використовуючи цю функцію, автоматизовані системи зможуть створювати більш природні та змістовні публікації.
- Автоматизація контент-стратегій: У майбутньому системи матимуть можливість автономно створювати та поширювати контент, а також розробляти стратегії для публікацій, аналізуючи дані про конкурентів, тенденції сучасного та дії аудиторії. Це дозволить користувачам отримувати найкращі результати від публікацій без необхідності активно брати участь у процесі.

Незважаючи на стрімкий розвиток технологій, все ще існують деякі проблеми, які потрібно вирішити. Наприклад, захист даних користувачів, підвищення ефективності алгоритмів для роботи в режимі реального часу та налаштування систем відповідно до мінливих вимог і вподобань користувачів.

Тим не менш, прогрес у цих технологіях свідчить про те, що автоматизація створення та розповсюдження контенту стане ще більш бездоганною та поширеною. Це дозволить компаніям, маркетологам та іншим користувачам покращити свою роботу та значно підвищити її ефективність, адаптуючись до мінливого ландшафту цифрового світу.

1.3. Огляд існуючих технологій та систем

1.3.1. Існуючі системи та їхні обмеження

Сучасні системи автоматизації публікацій пропонують різні методи для генерації та і розміщення контенту, але їх використання часто пов'язане з кількома суттєвими недоліками, які зменшують їх ефективність і зручність для широкого кола користувачів.

1. Професійні платформи для управління контентом. Багато компаній використовують професійні рішення для автоматизації публікацій, такі як Hootsuite або Buffer. Тим не менш, ці платформи мають кілька недоліків[1]:
 - Висока вартість: Висока вартість підписки на ці послуги може стати перешкодою для малих підприємств або окремих осіб, роблячи їх недоступними.
 - Складність використання: Багато користувачів вважають інтерфейси цих систем надто складними, що потребує часу для ознайомлення з технологією.
2. Мобільні додатки. Існує кілька мобільних програм, доступних для публікації контенту, але їх функціональність часто обмежена:
 - Обмежений функціонал: Багато програм надають лише базові функції планування, не маючи можливості ефективно аналізувати та оптимізувати вміст контенту.
 - Вузька спеціалізація: Деякі з них розроблені для конкретного типу контенту, що обмежує їх адаптивність.
3. Загальнодоступні інструменти. Серед рішень у вільному доступі також можна знайти ті, які забезпечують базовий аналіз контенту. Однак вони мають низку недоліків:
 - Обмежені можливості аналізу: їм часто не вистачає всебічного аналізу ефективності публікацій, що має вирішальне значення для користувачів, які прагнуть покращити свої стратегії контенту.

- Неповноцінність: Ці системи часто не можуть забезпечити необхідні функції для комплексного керування вмістом публікацій.

Незважаючи на наявність багатьох доступних рішень для автоматизації публікацій, більшість із них мають значні недоліки, які обмежують їхню ефективність і доступність для широкого кола користувачів.

Це створює можливості для просування нових, більш точних і економічно ефективних технологій у цій галузі.

1.3.2. Hootsuite:

Hootsuite — це платформа керування соціальними медіа, яка дозволяє компаніям і користувачам планувати публікації, відстежувати взаємодію та оцінювати ефективність публікацій у режимі реального часу. Система легко підключається до багатьох соціальних платформ, таких як X, Facebook, Instagram, LinkedIn, що дає змогу ефективно керувати кількома обліковими записами в одному місці. Опис технології:

Принцип роботи:

Hootsuite поєднує всі акаунти користувача в соціальних медіа у єдиний інтерфейс, який дає змогу користувачеві створювати майбутній контент, редагувати планувати майбутні публікації. Платформа дозволяє автоматично публікувати вміст на основі заздалегідь визначеного розкладу та дозволяє створювати черги публікацій. Крім того, hootsuite пропонує докладні звіти, які допомагають оцінити вплив і успіх вмісту.[3]

Застосування:

Багато компаній використовують hootsuite для керування своїми акаунтами на сторінках соц. мереж. Застосунок особливо корисний для великих організацій, яким потрібна координація між різними обліковими записами та автоматизація публікацій на різних платформах.

Переваги:

- Центральне управління: одночасне управління кількома обліковими записами.

- Мульти-публікування: Можливість створення кількох публікацій одночасно на різних платформах.
- Вбудована аналітика: Платформа пропонує вбудовану аналітику, яка дозволяє відстежувати ефективність ваших публікацій у соц. мережах.

Можливі недоліки:

- Вартість: висока вартість послуг, пов'язані з впровадженням системи CRM, можуть бути досить високими, особливо для великих команд або організацій.
- Обмежений доступ: безкоштовна версія платформи має обмеження на кількість використовуваних акантів в соціальних мережах, з якими можна працювати одночасно.
- Складність інтерфейсу: деякі користувачі згадують складність у налаштуванні інтеграції.

Hootsuite — це сервіс для керування соціальними мережами, який допомагає автоматично публікувати контент, стежити за активністю користувачів і аналізувати результативність постів у режимі реального часу. Вона об'єднує всі акаунти в єдиний інтерфейс, де можна створювати, редагувати та планувати контент для кількох платформ одночасно (X, Facebook, Instagram, LinkedIn). Платформа також підтримує аналітику, щоб оцінювати ефективність публікацій.

1.3.3. Buffer:

Buffer — широко використовувана платформа для автоматизації публікацій в соц. мережах. Застосунок дає змогу користувачам заздалегідь планувати публікації, організовувати вміст у черги та контролювати ефективність опублікованого вмісту через інтуїтивно зрозумілий і зручний інтерфейс[4]. Buffer сумісний із такими популярними платформами, як X, Facebook, Instagram і LinkedIn, що забезпечує бездоганну інтеграцію.

Принцип роботи:

Користувач може додати контент до черги, яка автоматично опублікує його на основі заздалегідь визначеного розкладу. Buffer також надає базову

аналітику, яка дозволяє оцінити кількість взаємодій, поширень та переглядів для кожної публікації.

Застосування:

Buffer широко використовується як малими підприємствами, так і великими організаціями для оптимізації процесів публікації у соціальних мережах. Такий ресурс досить практичний в використанні компаніями, які прагнуть підтримувати регулярний графік публікацій.

Переваги:

- Простота використання: зручний дизайн та інтуїтивно зрозумілий інтерфейс програмного забезпечення полегшили навігацію та використання будь-кому.
- Планування: можливість ефективного планування та публікації контенту на різних платформах.
- Інтегрованість: інтеграція з різними соціальними мережами.

Обмеження:

- Обмежений доступ: безкоштовна версія програмного забезпечення має обмежені можливості.
- Відсутність єдиного режиму: базовий пакет не включає розширену аналітику.
- Складність підтримки: не підтримує деякі складні форми інтеграції з платформами.

Buffer є однією з лідерів платформ для автоматизації публікацій за допомогою соціальних мереж, яка дозволяє користувачам планувати пости, організовувати контент у черги та відстежувати ефективність через простий інтерфейс. Сумісний з платформами, такими як X, Facebook, Instagram і LinkedIn, Buffer дозволяє автоматично публікувати контент за розкладом і надає базову аналітику для оцінки взаємодій. Він підходить для малого бізнесу та великих компаній, які прагнуть підтримувати регулярний графік публікацій. Серед переваг — зручність використання, планування та інтеграція, але безкоштовна

версія має обмеження, а розширені функції аналітики доступні тільки в платних планах.[4]

1.3.4. IFTTT:

IFTTT — онлайн-застосунок, який дозволяє користувачам автоматизувати робочі процеси між різними веб-сервісами. Це дає змогу встановлювати правила (аплети), які диктують, як різні служби можуть автономно виконувати певні дії у відповідь на події чи тригери, такі як публікація нової статті чи зміна статусу.

Принцип роботи: Ifttt працює за принципом «якщо це, то те», саме це надає змогу юзеру розробляти аплети, які автоматично запускають дії. Наприклад, ви можете налаштувати аплет для автоматичного обміну дописами в X, коли на веб-сайт додається нова стаття або оновлюється стрічка RSS.[5]

Застосування: IFTTT використовується для автоматизації публікацій для сучасних соціальних мереж, роботи з новинними сайтами, вилучення вмісту з веб-ресурсів та інших подібних завдань. Це універсальний інструмент для автоматизації рутинної діяльності.

Переваги:

- Інтеграція: можливість інтеграції з великою кількістю платформ та сервісів.
- Доступний інтерфейс: застосунок відомий своєю простотою встановлення правил автоматизації.
- Підтримка: компанія забезпечує підтримку широкого спектру робочих процесів.

Обмеження:

- Базова аналітика і обмежені можливості для аналізу результатів.
- Менш інтуїтивний інтерфейс для новачків.
- Для встановлення деяких інтеграцій може знадобитися технічний досвід.

IFTTT використовується для автоматизації публікацій в соц. мережах та управління контентом на веб-ресурсах. Основними перевагами є інтеграція з багатьма платформами, простий інтерфейс та універсальність, але платформа

має обмежені аналітичні можливості та може потребувати технічного досвіду для складних інтеграцій.

1.4. Невирішені проблеми та можливі напрямки для подальшого дослідження

1.4.1. Невирішені Проблеми

- Точність і релевантність контенту.

Автоматизовані системи для генерації та публікації контенту можуть стикатися з проблемами у створенні постів, які точно відображають поточні тенденції або відповідають потребам користувачів. Вибір неактуальних ключових слів або застарілих новин може знизити залученість аудиторії.

- Автоматизація взаємодії з користувачами.

Автоматичні системи публікацій здебільшого орієнтовані лише на односторонню взаємодію (публікацію), без врахування потреби у взаємодії з аудиторією. Відсутність можливостей для адаптивної відповіді на коментарі або реакції ускладнює побудову якісного діалогу з користувачами.

- Інтеграція з різними платформами.

Попри наявність ряду рішень, інтеграція з різними соціальними мережами та специфічними API (інтерфейсами програмування) для збору та публікації даних все ще є проблемою.

1.4.2. Можливі напрямки дослідження

- Покращення релевантності контенту через штучний інтелект.

Дослідження можуть зосередитися на розробці алгоритмів, які використовують глибоке навчання для покращення релевантності та точності автоматично згенерованих постів. Врахування контексту та актуальних тенденцій може значно підвищити якість контенту.

- Розвиток двосторонньої автоматизованої взаємодії.

Дослідження можуть бути спрямовані на розробку алгоритмів, які дозволять системам автоматизовано відповідати на коментарі або реакції користувачів, враховуючи специфіку комунікації в різних соціальних мережах.

- Інтеграція зі зростаючими платформами та інструментами.

Потрібні дослідження для покращення інтеграції автоматизованих публікацій із новими та менш відомими соціальними мережами. Створення універсальних рішень з відкритою архітектурою, які легко підключаються до нових платформ, є важливим кроком для розширення функціональності.

- Аналітика та рекомендації на основі поведінкових даних.

Потенційним напрямком дослідження є вдосконалення аналітичних інструментів, які б дозволяли автоматизованим системам використовувати поведінкові дані аудиторії для коригування стратегії публікацій. Це дозволить оптимізувати графіки постів, тематику і формат залежно від уподобань користувачів.

- Гнучка кастомізація контенту.

Варто розробити рішення, які б дозволяли більш тонко налаштовувати автоматизовані публікації під конкретні потреби брендів або індивідуальних користувачів. Такі системи мають підтримувати кастомізацію тональності, мови, стилю та контексту залежно від заданих параметрів.

Дані проблеми та можливі шляхи їх вирішення за даним дослідженням підкреслюють потенціал розвитку систем автоматизації генерації та публікації контенту. Покращення точності, кастомізації та інтерактивності допоможе не тільки підвищити залученість аудиторії, а й поліпшити ефективність комунікацій в соціальних мережах.

1.5. Постановка задачі

Основною метою даної теми є підвищення ефективності автоматизованих систем генерації та публікації контенту, за допомогою оптимізації алгоритмів парсингу, обробки та автоматичної публікації в сучасних соціальних мережах

отриманих даних. Виходячи з цього виникає необхідність створення інноваційної розробки. Дана розробка має бути спрямована на покращення існуючих методів та алгоритмів, які будуть автоматизувати процес пошуку або генерації контенту, автоматичного планування та постингу публікацій в соцмережах.

Задачі дослідження:

1. Аналіз існуючих методів та алгоритмів парсингу та автоматичного постингу:
 - Розгляд і вивчення методів та технологій автоматичної публікації контенту в соцмережах.
 - Оцінка ефективності, точності та доступності технологій для кінцевих користувачів.
2. Проектування плану вдосконалення методів парсингу та генерації контенту:
 - Пошук існуючих рішень, вдосконалення та адаптація методу парсингу під відповідні інтернет сторінки.
 - Написання свого парсингу для пошуку конкретних даних на сторінках новин для генерації контенту.
3. Розробка повноцінного застосунку з використанням створених рішень:
 - Створення програми з інтерфейсом, яка дозволяє автоматично виконувати збір інформації на задану тему, генерувати публікації за ключовими словами та планувати публікації без втручання людини в процес.
 - Тестування розробленої програми в реальному часі для розвитку бізнесу.
4. Аналіз результатів тестування і оцінювання ефективності результатів розробки:
 - Збір інформації про результат роботи застосунку, аналіз отриманих даних.

- Оцінювання ефективності роботи та зручності і доцільності в використанні.

5. Написання інструкції та рекомендацій для використання та розвитку розроблюваного продукту:

- Перелік можливих шляхів розвитку та рекомендації щодо доцільного використання.

Під час виконання поставлених завдань, найбільшу увагу було зосереджено на розробці модифікованих алгоритмів та застосунку, який буде не просто засобом автоматичної публікації контенту, а стане розробкою за допомогою якого можна легко і повноцінно збирати дані та створювати унікальний контент з подальшим його використанням.

Далі наведено порівняльну таблицю описаних вище технологій в порівнянні з розроблюваним продуктом. У цій таблиці 1.1 перелічені основні характеристики кожного з аналогів та переваги створеної розробки.

Порівняння існуючих технологій та розробки

	Функціональність	Доступність	Вартість	Вимоги до Обладнання	Персоналізація застосунку	Доступність для Індивідуальних Користувачів
Buffer	Публікація контенту, можливість планування, аналітика	Підтримка соціальних мереж	Висока	Мобільні пристрої та персональний комп'ютер	Ні	Для малого бізнесу та особистого використання
IFTTT	Автоматизована публікація на кількох сервісах	Підтримка різних платформ	Висока	Браузер та мобільний додаток	Так, обмежено	Для індивідуальних користувачів
Zapier	Автоматизована публікація, можлива інтеграція різних платформ	Підтримка інтегрованих додатків	Середня	Мобільний пристрій	Ні	Для малого та середнього бізнесу
Dlvr.it	Автоматизований постинг в соціальних мережах	Підтримка соціальних мереж та блог-платформ	Висока	Мобільна або десктоп-платформа	Ні	Для індивідуальних користувачів
Розробка	Збір даних, генерація контенту, автоматизований постинг та планування публікацій	Доступ до інтернет сторінок з новинами, до сторінок в соц. мережах та присутня можливість генерувати контент за ключовими словами	Відносно низька вартість при високому рівні автоматизації	Десктоп-платформа, можлива інтеграція на мобільні пристрої	Наявна функція персоналізації контенту на основі ключових слів	Для малого та середнього бізнесу та індивідуальних користувачів

Отже, розроблюваний застосунок забезпечує наступні можливості:

- Унікальність: результат генерації контенту за ключовими словами є завжди унікальним, навіть, якщо генерація відбувається за однаковими

ключовими словами, що дозволяє уникнути повторів в різних користувачів та конфліктних ситуацій пов'язаних з плагіатом.

- Актуальність: доступ до інтернет-новини дозволяє публікувати найактуальнішу інформацію.
- Висока ефективність створених публікацій: генерація особистих публікацій сприяє підвищувати ефективність використання соціальних мереж.
- Об'ємний функціонал: збір актуальної інформації, генерація власного унікального контенту, створення публікацій на основі отриманих даних, автоматизований постинг за власним розкладом.
- Доступна база даних: у користувача є доступ до таблиці, де можна підкорегувати контент або розклад, або власноруч додати необхідний матеріал для постингу, а також прослідкувати за процесом виконання публікацій.
- Інтуїтивний інтерфейс: максимально простий та зрозумілий функціонал програми.

ВИСНОВКИ ДО РОЗДІЛУ 1

У даному першому розділі було проведено аналіз головних методів для автоматичного постингу контенту в соціальних мережах. Окрім цього, було описано основний функціонал ведучих застосунків-аналогів на сьогоднішній день, а саме: Hootsuite, Buffer, IFTTT, Zapier, Dlv.it. Було визначено їх основні характеристики та переваги, також було розглянуто їх обмеження.

Під час проведення аналізу було визначено основні напрямки для подальшого дослідження та розробки технологій, а саме: написання унікального методу парсингу, покращення методу обробки та генерації контенту та автоматичного постингу.

Отже, розроблюваний продукт матиме потенціал в сфері бізнесу та маркетингу, адже завдяки ньому можна автоматизувати частину роботи маркетологів, або ж самостійно просувати свій продукт чи бізнес без особливих навиків роботи з соц мережами. Даний розділ описує теоретичну основу, яка використовуватиметься в подальшому процесі розробки та створення застосунку.

РОЗДІЛ 2

МЕТОДОЛОГІЯ ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДІВ

2.1. Загальна методика процесу дослідження

Під час проведення дослідження буде проведено вивчення методів, розробка власних, їх тестування і застосування методів та технологій, які були описані вище у першому розділі. Нижче наведено послідовність плану методики дослідження.[6]

1. Аналіз існуючих технологій. На даному першому етапі потрібно зробити загальний аналіз існуючих рішень на тему парсингу різноманітних інтернет сторінок з метою пошуку та вилучення необхідних даних, а також автоматизації публікації та генерації контенту. Для визначення найкращого способу потрібно встановити переваги кожного з таких методів, також взяти до уваги їх недоліки та шляхи модифікації.

2. Модифікація існуючих та розробка нових. Виходячи з результатів попереднього пункту, потрібно розробити модифікований метод або ж написати унікальний для автоматизації конкретних процесів. Такий алгоритм буде застосовано для розробки точного пошуку необхідної інформації та її форматування, крім того потрібно покращити існуючий метод автоматизації постингу та розробити повноцінну програму з візуальним інтерфейсом.

3. Тестування і підтвердження відповідності. Написаний унікальний алгоритм парсингу кількох інтернет сторінок повинен бути протестований на предмет релевантності отриманої інформації та правильного форматування тексту. Модернізований алгоритм постингу повинен бути протестований на коректність роботи та збільшення швидкості роботи програми. Тести оцінюватимуться не лише за шкалою швидкості виконання програми, а і оцінюватиметься аспект доцільності та актуальності інформації, завдяки чому процес автоматизації приносить більше користі.

4. Реалізація та аналіз результатів. Поєднання усіх створених та адаптованих методів в цілісний продукт. Проведення аналізу результатів за вище описаним алгоритмом та визначення чи досягнута поставлена мета.

5. Висновки та можливі шляхи розвитку. Проаналізувавши отримані дані після аналізу результатів розробленого застосунку, потрібно зробити висновки про коректність роботи та ефективність покращення алгоритмів. Крім того, варто сформулювати варіанти розвитку та подальшого вдосконалення даної розробки.

Алгоритм проведення дослідження продемонстровано на рисунку 2.1.



Рисунок 2.1. Загальна методика проведення дослідження.

2.2 Опис методів та засобів

2.2.1 Огляд джерел та методів для опису існуючих рішень.

1. Літературні джерела. Ознайомлення з літературними джерелами це фундаментальний процес для початку дослідження. Завдяки вивченню літератури можна зробити докладний аналіз існуючих технологій, що допоможе в написанні публікацій та наукових робіт.

Засоби:

- Бази наукової літератури.
- Журнали з науковими статтями та тематичні презентації.

2. Технічна документація. Система графічних і текстових документів, необхідних і достатніх для безпосереднього використання на усіх стадіях життєвого циклу продукції.[7]

Засоби:

- Системна документація.
- Документація користувача.

3. Тестування розробки. Процес проведення експериментів, під час якого проводиться тестові запуски готової розробки, тобто розроблених або покращених методів в повсякденних умовах. Тестування проводиться для отримання даних для подальшого аналізу результатів та ефективності роботи розробки.

Засоби, за допомогою яких проводились експерименти та аналіз результатів:

- Технічне забезпечення: Windows OS, Intell Core i5-12600h, ОЗУ 64ГБ, SSD 300ГБ
- ПЗ, що використовувалось для виконання експериментів: Projectmaker for Zennoposter, Zennoposter, скрипти Python з використанням PyInstaller, бібліотеки requests, BeautifulSoup, gspread та ін.

- Вхідні данні: перелік тем для парсингу, стандартизовані ключові слова для генерації публікації.

2.2.2 Параметри для порівняння та відбору існуючих методів автоматизації постингу:

1. Релевантність отриманих даних під час парсингу – найголовніший критерій, від якого залежить якість фінальної публікації.
2. Персоналізація – можливість вносити свої корективи, генерувати публікації за ключовими словами. Даний показник повинен бути на високому рівні для більшого функціоналу застосування.
3. Вартість реалізації – кошти, що були втрачені на покупку ліцензій або програм для створення розробки.
4. Сумісність компонентів – окремі функції повинні без проблемно поєднуватись між собою та з платформою.
5. Універсальність тематики – обрана технологія повинна враховувати різноманітність можливих запитів від користувачів з різних сфер зайнятості.
6. Швидкість виконання – кожний компонент методу повинен раціонально та оптимально використовувати ресурси, а саме, пам'ять, енергію, час, кількість процесорів.
7. Гнучкість – вміння підлаштовуватись під незвичні умови, наприклад, відсутність картинок під час збору інформації або невиконання деяких умов під час запуску.

2.2.3 Засоби і функції для створення розробки і покращення функцій

Функцією для добування початкових даних є парсинг, алгоритм, який робить пошук певного контенту на інтернет сторінках, його форматування на заповнення бази даних.

Засоби:

- Visual Studio Code –середовище програмування[8]
- Python та PyInstaller мова програмування [9]

Наступні функції для генерації публікацій та автоматичного постингу реалізовані за допомогою фреймворку Zennoposter та Projectmaker for Zennoposter з використанням мови програмування C#.

Візуальний інтерфейс створений в середовищі програмування Visual Studio Code за допомогою засобів HTML з використанням часткових кодових блоків на мові програмування java script.

Під час етапу покращення застосовано комбінований метод – створення частини власного методу для отримання даних з інтернет сторінок в поєднанні з існуючим методом генерації публікацій – модифікованим під ключові слова, та модифікованим та пришвидшеним методом автоматичної публікації контенту.

Засоби:

- Розроблений алгоритм парсингу на мові Python, з використанням бібліотек requests, BeautifulSoup, pandas, gspread, random, datetime.
- AI для генерації персоналізованих публікацій
- Алгоритми автоматичного постингу

Для імплементації вище описаних методів було створено комунікативні зв'язки таким чином, що кожен складовий компонент був легкодоступний для редагування та вдосконалення.

Засоби:

- Створення ru.exe файлів парсингу, які легко підключаються в основну автоматизовану систему.
- API для підключення AI.
- HTTP-запити для обміну даними між веб-інтерфейсом та застосунком.

Проведення експериментів: запуск системи з різними початковими сценаріями та різноманітними запитами користувача.

- Кілька ПК з різними характеристиками.
- Набори вхідних даних, емуляція роботи користувача.

Підвищення ефективності та продуктивності забезпечують:

- Унікалізація алгоритму парсингу, форматування даних.
- Розпаралелення виконуваної програми на оптимальну кількість потоків.
- Персоналізація користувацьких запитів.

Інструменти:

- Projectmaker for Zennoposter для відслідковування кількості та швидкості процесів [10].
- Покращені алгоритми автоматизації, збору інформації.

2.3 Оцінка методів автоматизованого постингу та парсингу

2.3.1 Огляд технологій автоматизованої публікації контенту

З метою визначення найоптимальніших методів для розробки, було проведено дослідження існуючих рішень, які поділені на такі категорії:

1. Інтегровані платформи для управління соціальними мережами. Це повноцінні платформи, які пропонують загальний інтерфейс для керування кількома соціальними сторінками одночасно, планування публікацій та автоматизації повторних публікацій. За допомогою таких застосунків проводяться масові розсилки та створюються інтеграції з API соціальних мереж[11]. Части саме в таких застосунках виникають проблеми з інтеграцією з різноманітними платформами, крім того, вони залежні від зовнішніх сервісів, що може провокувати конфлікти.

2. Скрипти та користувацькі боти з використанням API. Такі системи, що використовують API соціальних мереж, дозволяють створити автономний постинг з високим рівнем гнучкості. Багато соц. мереж самостійно пропонують такі API розробникам, за допомогою яких можна програмно робити дії користувача. Проте для таких рішень потрібне додаткове налаштування, може знадобитись технічна підтримка для обробки змін в API, крім того, можуть бути встановлені ліміти зі сторони API.[12]

3. Headless-браузери. Такі методи найчастіше використовуються для взаємодії з веб-інтерфейсами, зокрема і для публікації контенту. Саме вони дозволяють програмно взаємодіяти з веб-сторінками імітуючи дії користувача, а саме, натискання клавіш, заповнення форма, скролінг. Цей метод дозволяє автоматизувати обмін даними навіть з тими платформами, які не мають свого API.[13]

Кожен з цих методів має свої переваги та недоліки і в залежності від можливих інтеграцій потрібно обрати найпідходящий.

2.3.2 Опис алгоритмів парсингу

1. Парсинг на базі DOM-структур. Даний алгоритм працює з HTML-кодом, перетворюючи його на свого роду дерево елементів (тобто DOM-дерево). При процес парсингу добуває інформацію за допомогою роботи з елементами DOM-дерева. Цей алгоритм спрямований на пошук окремих тегів, класів або інших складових дерева для отримання цільових даних[14].

Сама DOM розроблена таким чином, щоб документ, не залежно від мови програмування, був представлений як структурований документ з окремими та послідовними залежностями[15]. Це дозволяє швидко знаходити необхідну частину HTML-сторінки та відслідковувати потрібні зв'язки для отримання конкретної інформації.

2. Алгоритм парсинга на базі регулярних виразів. Даний алгоритм отримує дані за допомогою стандартних регулярних виразів. Це достатньо потужний інструмент, який проводить роботу пошуку та форматування тексту. Але алгоритм регулярних виразів є не таким гнучким, саме через це для нього обробка не шаблонного HTML-коду може бути досить складною. [16]

Перевагою такого методу є те що він ідеально підходить для обробки неструктурованих даних та для вирішення задач невеликого обсягу. Особливістю є те, що цей алгоритм не підходить для динамічних інтернет-ресурсів та має обмежене розуміння структури документу.

3. XPath та CSS-селектори. Метод парсинга з використанням XPath та CSS-селекторів також працює з структурою документу, але при цьому використовує більш новітні засоби для пошуку конкретних елементів. За допомогою XPath та CSS-селекторів елементи документа адресуються на базі їх атрибутів а вкладеності[17].

Такий алгоритм для пошуку даних посилає запити до xml-розмітки. Тобто для вдалого виконання парсингу потрібно правильно задати шлях до потрібного елементу[17]. Цей алгоритм є більш підходящим для складних сайтів з динамічними змінами та складною будовою HTML-сторінки. Але вони є більш ресурсно-затратними.

4. Алгоритм парсингу з використанням Headless-браузеру. Даний метод в основі запускає браузер без графічного інтерфейсу з метою отримання даних. В сучасних сайтах часто підключені скрипти, які реагують на будь-який рух користувача та змінюють наповнення сторінки[18]. Саме тому такий метод є дієвим при динамічних веб-сайтах, де наповнення генерується певним скриптом.

Разом з цим, алгоритм з використанням Headless-браузеру є більш затратним в плані ресурсів, але здатний добувати дані зі сторінок, на яких дані доступні тільки після виконання коду

5. API-парсинг – алгоритм добування даних не стандартним способом, за допомогою надсилання POST-запитів через API. Сучасні сайти часто відкривають доступ до своїх даних через API, що значно пришвидшує роботу отримання даних. Крім того, дані отримуються в зручній формі в форматі файлу JSON або XML. Також перевагою даного методу є те, що він не залежить від змін структури HTML-коду[19] та є достатньо економним в використанні ресурсів. Проте великим мінусом є його не універсальність, тому що не всі сайти відкривають своє API-підключення.

Нижче наведено таблицю 2.1 для порівняння методів автоматичного постингу та таблицю 2.2 для порівняння алгоритмів персингу за певними критеріями.

Таблиця 2.1

Порівняння методів автоматичного постингу

Характеристика	SMM-платформи	Користувацьке API	Headless-браузери
Гнучкість	Обмежена	Висока, при умові наявності API	Висока, навіть при відсутності API
Підтримка динамічного контенту	Ні	Ні	Так
Залежність від зовнішніх факторів	Так	Так	Ні
Швидкість	Середня	Висока	Висока
Точність	Середня	Середня	Висока
Масштабованість	Середня	Висока	Висока

По результатам порівняння Headless-браузери є найкращим рішенням для автоматизації на сайтах, які не мають або не дозволяють до свого API або там де потрібно працювати з динамічними елементами.

Таблиця порівняння алгоритмів парсингу

Алгоритм	Швидкість	Релевантність результатів	Підтримка динамічних сайтів
DOM-структура	Середня підвищена	Висока для статичних сайтів	Ні
Регулярні вирази	Висока	Середня	Ні
XPath і CSS- селектори	Середня	Висока	Так
Headless	Низька	Висока	Так
API-парсинг	Висока	Висока	Ні

Проаналізувавши отримані результати дослідження існуючих алгоритмів парсингу, можна сказати наступне. Алгоритм з використанням DOM-структури досить зручний та дає точні результати, проте працює тільки зі статичними сайтами. Алгоритм на основі регулярних виразів має високу швидкість, але зовсім не універсальний. Headless має дуже низкий показник швидкості, що при обробці великого обсягу даних може призвести до поганих результатів. API-парсинг має гарні показники швидкості та релевантності результатів, але не на всіх платформах підтримується. Алгоритм з використанням XPath і CSS-селекторів показав хороші результати під час аналізу, але не з максимальною швидкістю. Тому ідеальним варіантом є створення гібридного алгоритму з поєднанням алгоритмів DOM-структури та XPath і CSS-селекторів.

Оскільки алгоритм на основі DOM-структури дозволяє зручно шукати інформацію за допомогою дерева інтернет-сторінки, але не підтримує динамічних, а деталі з методу CSS-селекторів дозволяють модифікувати алгоритм та прив'язатись саме до елементу, а не до шляху. В результаті чого, модифікований алгоритм може знаходити потрібну інформацію на динамічних сторінках не залежно від того де вона розміщена. Інші методи були відкинуті

тому, що в регулярні вирази не зовсім практичні в даному випадку, Headless має дуже низьку швидкість, а при великому обсязі даних це буде проблемою, а API-парсинг не динамічний та не підтримується деякими платформами.

2.4. Поєднання обраних методів між собою

В ході проведення аналізу існуючих алгоритмів добування даних було прийняте рішення розробити власний гібридний алгоритм на основі алгоритмів DOM-структури та XPath і CSS-селекторів.

Оскільки алгоритм на основі парсингу DOM-структури має перевагу в зручному та швидкому пошуку інформації за допомогою HTML-дерева інтернет сторінки, його було взято за основу.

Проте даний алгоритм не підтримує пошуку на динамічних веб-сторінках, було прийнято рішення, застосувати також метод з CSS-селекторами. Саме вони дозволяють відшуковувати конкретний фрагмент за його тегами, не залежно де саме він знаходиться на сторінці.

Тобто, DOM-структура має повний контроль над навігацією документу, а CSS-селектори дають змогу гнучко оперувати динамічними інтернет-сторінками, в яких елементи можуть працювати асинхронно та мінятись в процесі роботи з ними.

Спосіб роботи даних алгоритмів разом наступний та продемонстрований на рисунку 2.2. Використовуючи DOM-структуру для парсингу спочатку первинного завантаження і відтворення макету дерева HTML-документу, алгоритм може швидко знаходити необхідні фрагменти сторінки. В результаті чого робота за допомогою CSS-селекторів дозволяє більш детально розбирати сторінку та діставати самі глибокі елементи. Саму це дозволяє уникнути помилок, які могли б виникнути під час парсингу динамічних сторінок за допомогою DOM-алгоритму.

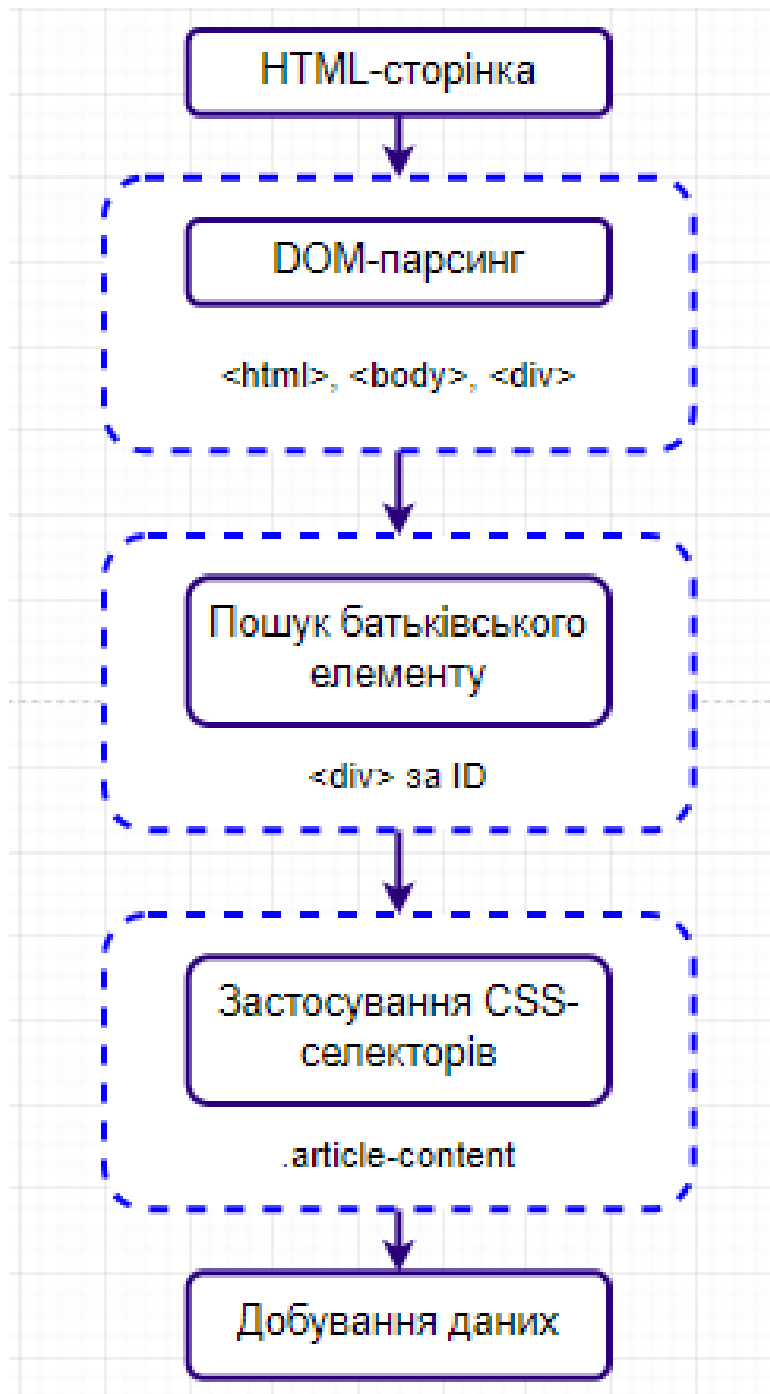


Рисунок 2.2. Поєднання методів DOM-структури та CSS-селекторів.

Отже, поєднання DOM-структури з методом CSS-селекторів дозволяє модифікувати алгоритм та прив'язатись саме до елемента, а не до шляху. В результаті чого, модифікований алгоритм може знаходити потрібну інформацію на динамічних сторінках не залежно від того де вона розміщена. А алгоритм DOM-структури дозволяє правильному виокремленню із дерева необхідних

даних. Створений алгоритм забезпечить більшу швидкість виконання алгоритму та більш високу якість отриманого тексту, який не потребує подальшої обробки.

Для модифікації нового алгоритму з поєднанням методів DOM-структури та CSS-селекторів було використано фреймворк Visual Studio Code та мова програмування Python. Visual Studio Code – не дуже ресурсно-затратний, але доволі потужний фреймворк, який має різноманітні розширення для роботи в різних напрямленнях. Даний продукт розроблений Microsoft студією та є доопрацюванням продукту Microsoft Visual Studio[20]. Він має обширний перелік розширень, які згодяться для різних мов програмування, а саме, Python, Java, C++, Go, PHP і т.д., та середовищ виконання, таких як .NET та Unity[21]. Такі розширення можуть поповнювати функції стандартного набору.

Парсинг створений за допомогою мови програмування Python, так як дана мова є найбільш підходящою для роботи з обраними методами та дозволяє використовувати окремі бібліотеки, які спрощують роботу з кодом. Це додає переваг в розробці веб-скрапінгу. Наприклад, за допомогою бібліотеки BeautifulSoup спрощується процес парсингу інтернет сторінки, при чому перевагою є те, що дана бібліотека працює з динамічними сторінками. Також за допомогою мови Python досить зручно оперувати та формувати дані та підключати бази даних, адже дана мова програмування підтримує такі інтеграції.

Інша частина створюваного продукту розроблена з використанням фреймворку Zennoposter. Zennoposter розроблений спеціально для створення автоматизаційних програм з додаванням веб-інтерфейсу. Сама автоматизація проходить завдяки емуляції дій користувача в браузері на інтернет сторінці в реальному часі. Саме це дозволяє швидко працювати з сайтами, які не мають готового відкритого API для обміну даними, або коли API недостатньо для роботи. Також даний фреймворк підтримує інтеграцію з зовнішнім API, що дозволяє розширити функціонал програми та підключити AI для генерації публікацій за ключовими словами.

Крім того, Zennoposter дозволяє розпаралелювати процеси програми, та виконувати кілька сценаріїв одночасно, це забезпечує більшу швидкість та

ефективність розробки. Завдяки повній автоматизації фреймворк знижує ризик людського фактору, це досить важливо в роботі з великими даними, складних процесах та послідовних діях.

Отже, ZennoPoster був обраний для створення програми автоматизації постингу завдяки своїй гнучкості в роботі з інтернет-сторінками, можливістю інтеграції по API з іншими методами та зручній автоматизації емуляції роботи користувача в інтернет просторі.

ВИСНОВКИ ДО РОЗДІЛУ 2

У даному другому розділі було проведено огляд алгоритмів та методів існуючих алгоритмів добування інформації з інтернет сторінок та методів автоматизації публікацій. Також, було описано загальний алгоритм, за допомогою якого здійснювалося дослідження та аналіз описуваних методів автоматизації.

Окремо було висвітлено тему критеріям, за якими порівнювались досліджувані алгоритми та методи. В результаті чого було визначено переваги та недоліки кожної з технологій парсингу та автоматизації процесу публікацій контенту в соціальних мережах.

Після проведення аналізу було визначено найефективніші технології та їх можливі поєднання. Для парсингу було обрано розробити гібридний метод на основі двох алгоритмів, а саме, алгоритму парсингу DOM-структури та алгоритму роботи з CSS-селекторами. Що дозволить зробити метод добування даних з HTML-сторінки більш швидшим та точним. В якості технологій автоматизації постингу для модифікації було обрано найефективнішу це технологій Headless-браузерів.

Крім того, в даному розділі було розглянуто послідовний процес поєднання двох технологій парсингу для модифікації алгоритму, що дозволить зробити його більш ефективнішим та отримувати більш релевантні дані. До того ж, було розглянуто методи комунікації автоматизованого методу постингу з зовнішніми компонентами. Результат повинен бути у вигляді модифікованих компонент, які працюють швидше та забезпечують більш точний результат.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1. Розробка моделі системи

3.1.1. Виявлення вимог до розроблюваного застосунку

Для розробки даної системи головною характеристикою для аналізу є майже повна автоматизованість процесів. Застосунок створюється з ціллю максимального виключення людського фактору з процесу пошуку інформації та публікації контенту у соціальних мережах.

Не менш важливим фактором є релевантність отриманих даних в результаті виконання скрипту. Розроблювана система повинна обробляти або генерувати та форматовувати отримані дані. Крім того пошук даних повинен працювати ідеально, щоб на виході отримувати достовірну інформацію. Ця характеристика є одною з найголовніших, від якої залежить правильність функціонування продукту.

Інтуїтивний інтерфейс являється ще однією важливою характеристикою даної розробки. Очікується, що додаток матиме зручний та зрозумілий функціонал. За допомогою розробленого інтерфейсу, користувачу потрібно буде налаштовувати початкові опції. Отже, результат виконання програми напряду залежить від правильних налаштувань користувача на початку роботи з веб-інтерфейсом.

Вище наведено основні вимоги до майбутньої розробки, виконання яких не тільки повинно підвищити ефективність методів, а і повністю автоматизувати частину роботи маркетологів та інших людей, хто має на меті розвивати якимось чином сторінки в соціальних мережах.

3.1.2. Проектування складових системи

Сам хід розробки складових елементів має на меті визначити поєднувані зв'язки між компонентами та самі компоненти системи, щоб підвищити достовірність та ефективність роботи розробки. Найбільш важливе завдання при цьому розподілення логічних частини системи. Наприклад, потрібно чітко

розмежувати модуль обробки даних, користувацький інтерфейс, систему вводу виводу даних, місце для зберігання даних та зовнішні функції. Нижче на рисунку 3.1 зображена схема взаємодії складових системи.

- Модуль взаємодії з користувачем. Даний модуль являє собою просте відображення певних функцій користувачеві, які тягнуть за собою складні алгоритми невидимі користувачеві. Головною ціллю під час розробки інтерфейсу було досягнути максимально інтуїтивно-зрозумілого зовнішнього вигляду веб-відображення, від якого залежатиме правильність початкових налаштувань користувачем.
- Модуль обробки даних. Модуль, який працює над обробкою даних є одним із головних компонент програми. Де обробляються усі отримані дані, як від користувача – налаштування початкових параметрів для запуску програми, так і від безпосередньо бази даних та інших зовнішніх компонент програми. Від правильної обробки цих даних залежить успішність виконання програми. Адже саме тут отримані дані перетворюються в потрібний вигляд, поєднуються між собою, форматуються та готуються до публікації.
- Модуль, який виконує процес отримання даних з веб-ресурсів. Він знаходиться всередині модуля обробки даних, та відповідає за форматування даних які надходять в результаті парсинга інших веб-джерел на різні теми. Цей модуль відповідає за роботу з великим обсягом даних та коректність її обробки.
- Модуль взаємодії з зовнішнім середовищем. Даний модуль містить підключені розроблені методи до основного елементу програми. Тут знаходиться метод парсингу інтернет-ресурсів, який виконує пошук свіжих новини на задану тему користувачем та передає отриману інформацію в базу даних. Окрім парсингу, тут є і інший імплементований метод, а саме Модуль AI, який підключений за допомогою API та відповідає за генерацію публікацій за ключовими словами наданими користувачем.
- Модуль взаємодії з базою даних та база даних. База даних являє собою таблицю з великим обсягом даних. Підключення бази даних відбувається через API та має захист за допомогою зашифрованого ключа. Тобто доступ до

даних користувача має тільки програма та сам користувач, що забезпечує захист користувацьких даних.

Маючи таку продуману структуру, програма повинна працювати злагоджено та ефективно, швидко виконуючи завдання користувача та затрачаючи на оптимальну кількість ресурсів.

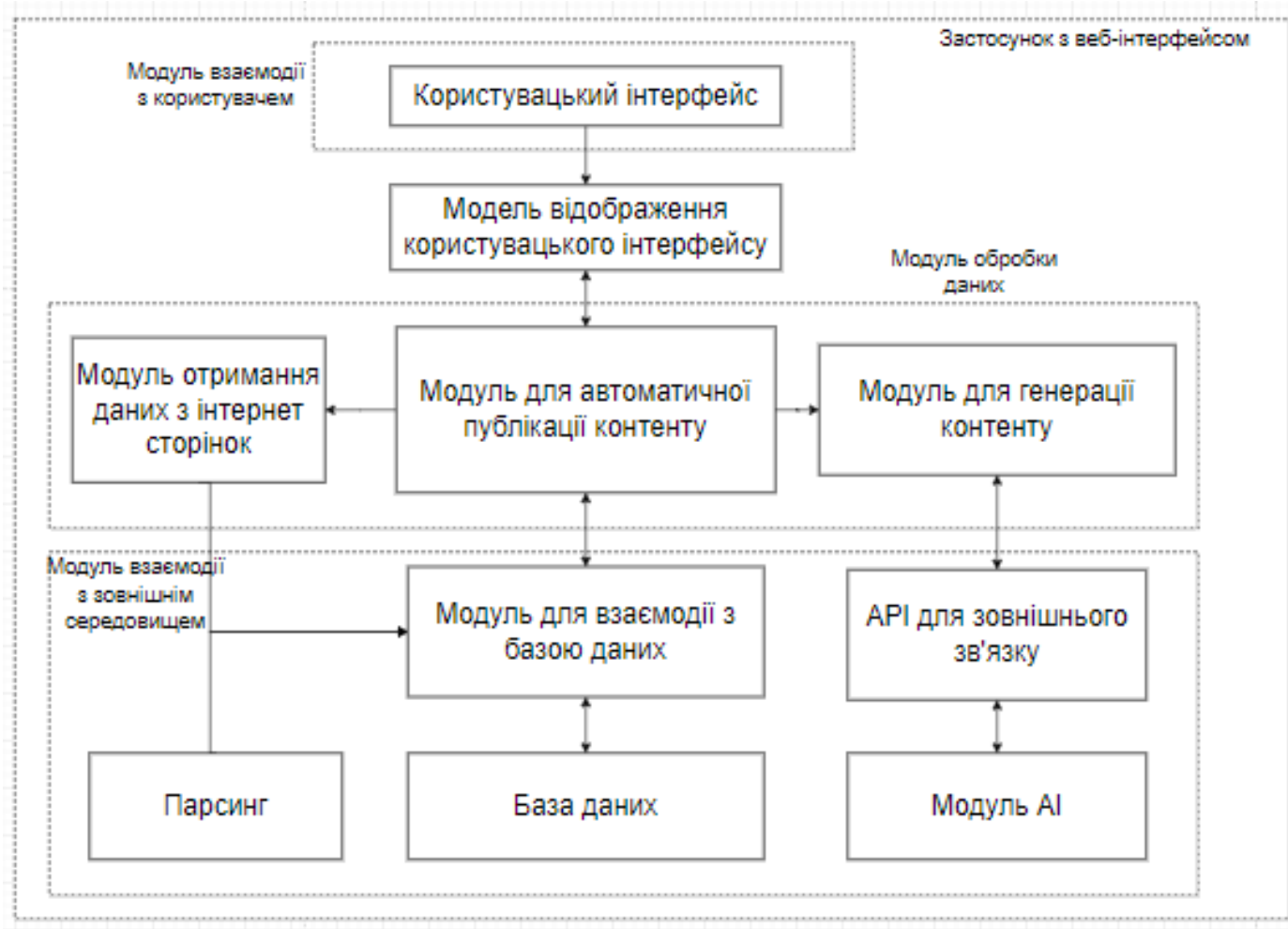


Рисунок 3.1. Схема взаємодії складових системи

3.2. Безпосередня розробка застосунку

3.2.1. Головні етапи розробки

В даному розділі поетапно описано процес створення розробки. Вона складається з кількох компонент, які разом працюють як повноцінна ефективна

розробка для автоматичної генерації та постингу контенту в соціальних мережах. Отже, розробка поділяється на такі програмовані частини:

- Пошук та отримання даних з інтернет сторінок. Це процес парсингу, який знаходить необхідні дані та відправляє їх до бази даних, роблячи форматування. Даний метод забезпечує отримання актуальних та достовірних даних з інтернет джерел новин на популярні теми.
- Генерація за ключовими словами. Даний процес відбувається за допомогою AI, яке отримує на вхід ключові слова написані користувачем, та генерує красивий текст-оголошення чи іншого формату. Далі ці дані форматовуються та передаються в базу даних, які далі використовуються під час планування публікації або відразу публікується на заданій сторінці.
- Взаємодія з базою даних. В якості місця зберігання даних в цій розробці виступають Google Sheets, яке підключене за допомогою API та захищені за допомогою шифрованого ключа для підвищення безпеки даних користувача.
- Автоматизований постинг. Це центральний компонент розробки, в якому відбувається поєднання усіх підключених методів та запуск їх в потрібній послідовності. Крім того, він отримує вхідні дані від користувача та реалізує автоматичну публікацію контенту, використовуючи усі отримані дані від користувача та API. Цей процес відбувається за допомогою емуляції роботи людини в соц.мережі.
- Веб-інтерфейс. Елемент програми, який напряду контактує з користувачем. З його допомогою користувач може задати початкові налаштування для отримання бажаного результату та запустити проект.

Завдяки цим компонентам, розробка може виконувати основне завдання даної магістерської роботи, а саме, реалізувати автоматичну генерацію та публікацію контенту в соціальних мережах.

3.2.2. Парсинг

Розроблений метод парсингу є модифікованим алгоритмом на основі методу парсингу DOM-структури та методу CSS-селекторів.

- Процес отримання даних з інтернет-сторінки, а саме з HTML коду, за допомогою DOM-парсингу продемонстровано на рисунку 3.2.

```
r = requests.get(URL_TEMPLATE)
print('\nStatus request =', r.status_code)
soup = bs(r.text, "html.parser")
links_par = soup.find_all('a', class_='row-result-body')
```

Рисунок 3.2. Застосування методу DOM-структури

Для даного фрагменту коду використовується бібліотека BeautifulSoup для аналізу DOM-структури. За допомогою цього методу, знаходяться усі посилання на новини з певної категорії. Для цього застосовується метод find-all, що дозволяє збирати дані з конкретних елементів HTML-сторінки.

- Використання CSS-селекторів для знаходження заголовків текстів на кожній с сторінок новин зображено на рисунку 3.3.

```
rs = requests.get(url_zaglavie)
root = bs(rs.content, 'html.parser')
article = root.find('h1')
```

Рисунок 3.3. Застосування алгоритму пошуку CSS-селекторів

В даному випадку знаходиться заголовок публікацій, використовуючи CSS-селектор тегу заголовків, а саме, <h1>. Тобто тут метод CSS-селекторів застосовується для знаходження певних елементів сторінки.

- Пошук ілюстрацій за допомогою розробленого комбінованого методу показано на рисунку 3.4.

Тут використовується DOM-парсинг для знаходження контейнера картинки і одночасно, за допомогою CSS-селекторів, знаходиться конкретний тег , звідки дістається посилання на картинку. Тобто, в даному алгоритмі DOM-структури застосовуються для початкового аналізу сторінки та пошуку спільних контейнерів з даними, а, в свою чергу, CSS-селектори застосовуються для чіткого знаходження необхідних елементів всередині цих контейнерів, що особливо важливо для динамічних сайтів.

```

images = soup.find('div', class_='article__content__head_img')
import time
time.sleep(1.5)
photol = []

if images:
    image_tag = images.find('img')
    if image_tag:
        img_url = image_tag['src']
        print("Image: ", img_url)
        photol.append(img_url)
    else:
        print("Image tag not found")
else:
    print("Div not found")

```

Рисунок 3.4. Застосування комбінованого методу парсингу

Такий комбінований підхід дозволяє алгоритму бути більш гнучким і стійким до динамічних змін веб-сторінок.

- Функція для парсинга заголовків текстів показана на рисунку 3.5 нижче.

```

def zaglavie(url_zaglavie):

    rs = requests.get(url_zaglavie)
    root = bs(rs.content, 'html.parser')
    article = root.find('h1')
    import time
    time.sleep(0.5)
    title = []
    title.append(article.text)
    return title

```

Рисунок 3.5. Реалізація функції пошуку заголовків

Дана функція приймає на вхід посилання, яке генерується за допомогою налаштувань користувача в веб-інтерфейсі та повертає заголовок, використовуючи алгоритм DOM-парсингу. Також тут використана затримка виконання коду для уникнення перенавантаження кількістю запитів до веб-сайту.

- Нижче на рисунку 3.6 наведено наступний цикл для отримання заголовків з усіх знайдених публікацій.

```
text_pars = []
for l in range(len(links)):
    url_zaglavie = links[l]
    text_pars.append(zaglavie(url_zaglavie))
    l += 1
for n in range(len(text_pars)):
    print(text_pars[n], '\n')
```

Рисунок 3.6. Цикл отримання заголовків

В даному циклі заголовки усіх публікацій витягуються за допомогою вище описаної функції zaglavie(), під час якого перебираються усі посилання до яких застосовується цей метод.

- Функцію для парсингу картинок зображено на рисунку 3.7.

```
def mass_for_photo(lk):
    p = requests.get(lk)
    soup = bs(p.text, 'html.parser')

    images = soup.find('div', class_='article__content__head_img')
    import time
    time.sleep(1.5)
    photol = []

    if images:
        image_tag = images.find('img')
        if image_tag:
            img_url = image_tag['src']
            print("Image: ", img_url)
            photol.append(img_url)
        else:
            print("Image tag not found")
    else:
        print("Div not found")

    div_not_f = soup.find('div', class_="longread-header-image")
    if div_not_f:
        img_tag1 = div_not_f.find('img')
        if img_tag1:
            img_url1 = img_tag1['src']
            print('Image URL found:', img_url1)
            photol.append(img_url1)
        else:
            print("Not found all")

    return photol
```

Рисунок 3.7. Парсинг фото з альтернативним варіантом

Функція `mass_for_photo()` знаходить фото на кожній публікації. Це складний метод, в якому використовується і DOM-парсинг, так як алгоритм парсингу за допомогою CSS-селекторів для пошуку зображень. Для цього дана функція спочатку знаходить основний блок з картинками застосовуючи пошук через CSS-селектор класу, і потім дістає звідти посилання. Якщо сторінка виявилась динамічною з іншим сценарієм, то CSS задіює інший селектор, щоб знайти необхідне зображення.

3.2.3. Робота з базою даних

В даній розробці в якості бази даних виступає Google Sheets, тобто необмежена таблиця. Вона підключається за допомогою API, використовуючи авторизацію, що продемонстровано на рисунку 3.8.

```
# Підготовка до авторизації
SHEET_ID = '1TN7uuQprkXBrPeXct8kPiz9LDHPLLeRGZOMrprD2G9g'
SHEET_NAME = 'List1'
# Завантаження та підключення до Google Sheets
credentials_file = 'sheet-automation-zen-c6b81725474c (3).json'
gc = gspread.service_account(credentials_file)
spreadsheet = gc.open_by_key(SHEET_ID)
worksheet = spreadsheet.worksheet(SHEET_NAME)
rows = worksheet.get_all_records()
```

Рисунок 3.8. Підключення бази даних

Авторизація проходить за допомогою зашифрованого ключа та файла `.json`. Також відразу підключається необхідний лист таблиці для роботи.

Для передачі даних до цієї таблиці потрібно їх сформувати в певний набір, це показано на рисунку 3.9. При цьому створюються відповідні набори даних, які в ході виконання коду заповнюються в правильній послідовності. А саме, тут присутні такі стовбці як заголовок тексту, текст, посилання на текст, посилання на картинку – це все витягується за допомогою парсингу та форматується. Також тут присутні такі поля як дата, яка генерується автоматично, враховуючи поточну дату, час для планування публікації, лічильник. Ці та інші поля зображені на рисунку 3.10.

```

counter = len(df['Counter'])
time_format = ['AM', 'PM']
row = []

text_pars1 = [item[0] for item in text_pars]
text_pars2 = [item.replace('\xa0', ' ') for item in text_pars1]

photo_all1 = [item[0] if item else "No image available" for item in photo_all]

for i in range(len(links)):
    if hour_now < 12:
        row = [month_today, day_today, year_today, hour_now + 1, random.randint(10, 59),
               time_format[0], text_pars2[i], photo_all1[i], links[i], '', '', counter]
        worksheet.append_row(row)
        counter += 1
    else:
        row = [month_today, day_today, year_today, hour_now - 11, random.randint(10, 59),
               time_format[1], text_pars2[i], photo_all1[i], links[i], '', '', counter]
        worksheet.append_row(row)
        counter += 1

```

Рисунок 3.9. Процес заповнення таблиці

	A	B	C	D	E	F	G	H	I	J	K	L
1	Month	Day	Year	Hour (0-12)	Minute(00-59)	AM/PM	Text for tweet	Link on picture	Link on post	Done?	Error	Counter
2	November	17	2024	5	20	PM	У напівпрозорому топі та	https://img.tsn.ua/cach	https://tsn.ua/lady/news	Done		0
3	November	17	2024	5	10	PM	"Я ледь не помер у черв"	https://img.tsn.ua/cach	https://tsn.ua/prosport/vi	Done		1
4	November	17	2024	5	47	PM	Для охорони Трампа вики	https://img.tsn.ua/cach	https://tsn.ua/tsikavinki/	Done		2
5	November	17	2024	6	23	PM	В Україні різко здешевша	https://img.tsn.ua/cach	https://tsn.ua/ukrayina/v	Done		3
6	November	17	2024	6	50	PM	Про що свідчить сьогодні	https://img.tsn.ua/cach	https://tsn.ua/ukrayina/p	Done		4
7	November	17	2024	7	26	PM	Якому знаку зодіаку пощ	https://img.tsn.ua/cach	https://tsn.ua/lady/astroj	Done		5
8	November	17	2024	7	13	PM	Інвестиції у «зелену» ене	https://img.tsn.ua/cach	https://tsn.ua/groshi/inv	Done		6
9	November	17	2024	7	48	PM	Російські дрони могли по	https://img.tsn.ua/cach	https://tsn.ua/svit/rosiysk	Done		7
10	November	17	2024	8	51	PM	Шольц виправдався за с	https://img.tsn.ua/cach	https://tsn.ua/politika/sh	Done		8
11	November	17	2024	8	12	PM	Відсвяткували річницю: п	https://img.tsn.ua/cach	https://tsn.ua/lady/news			9
12	November	17	2024	8	37	PM	Чоловік покійної Руслани	https://img.tsn.ua/cach	https://tsn.ua/glamur/ch			10
13	November	17	2024	9	56	PM	На Вовчанському напрям	https://img.tsn.ua/cach	https://tsn.ua/ato/na-vov			11
14	November	17	2024	9	19	PM	Які цифри у даті народж	https://img.tsn.ua/cach	https://tsn.ua/other/yaki			12
15	November	17	2024	9	10	PM	Як зробити волосся блис	https://img.tsn.ua/cach	https://tsn.ua/lady/kraso			13
16	November	17	2024	10	31	PM	Високі роботи в осінньом	https://img.tsn.ua/cach	https://tsn.ua/lady/news			14
17	November	17	2024	10	25	PM	"Наступні тижні будуть ви	https://img.tsn.ua/cach	https://tsn.ua/svit/nastur			15
18	November	17	2024	10	25	PM	Легендарний Холіфід ві	https://img.tsn.ua/cach	https://tsn.ua/prosport/le			16
19	November	17	2024	11	28	PM	Масований обстріл енерг	https://img.tsn.ua/cach	https://tsn.ua/ukrayina/n			17
20	November	17	2024	11	29	PM	Російські дипломати запл	https://img.tsn.ua/cach	https://tsn.ua/svit/rosiysk			18

Рисунок 3.10. Загальний вигляд бази даних

Дана база даних виконує важливу роль в розробці, оскільки саме тут зберігається вся детальна інформація про майбутні або вже виконані публікації. У користувача є доступ до даної таблиці, тому про бажанні, він може відслідковувати процес автоматичного постингу або відразу на сторінці у соціальній мережі, або в базі даних, де відстежується прогрес виконання.

3.2.4. Автоматизований постинг

Спосіб автоматизованого постингу був створений за допомогою фреймворку ZennoPoster. Завдяки цьому бот має безпосередній вихід в інтернет та може проводити емуляцію дій користувача в реальному часі при роботі з

соціальними мережами. Це є великою перевагою, оскільки такий функціонал дозволяє працювати з сайтами які навіть не мають API для взаємодій з ними.

Для отримання даних за допомогою парсингу, бот по команді користувача запускає Python-код, який виконується паралельно на кількох потоках для економії часу та збільшенню ефективності. І вже під час перших отриманих результатів паралельно запускається постинг-бот.

В даному випадку в якості інтернет-ресурсу для постингу публікацій було обрано платформу X. Для правильної роботи з боту було реалізовано авторизацію та альтернативний вхід – це зображено на рисунку 3.11.

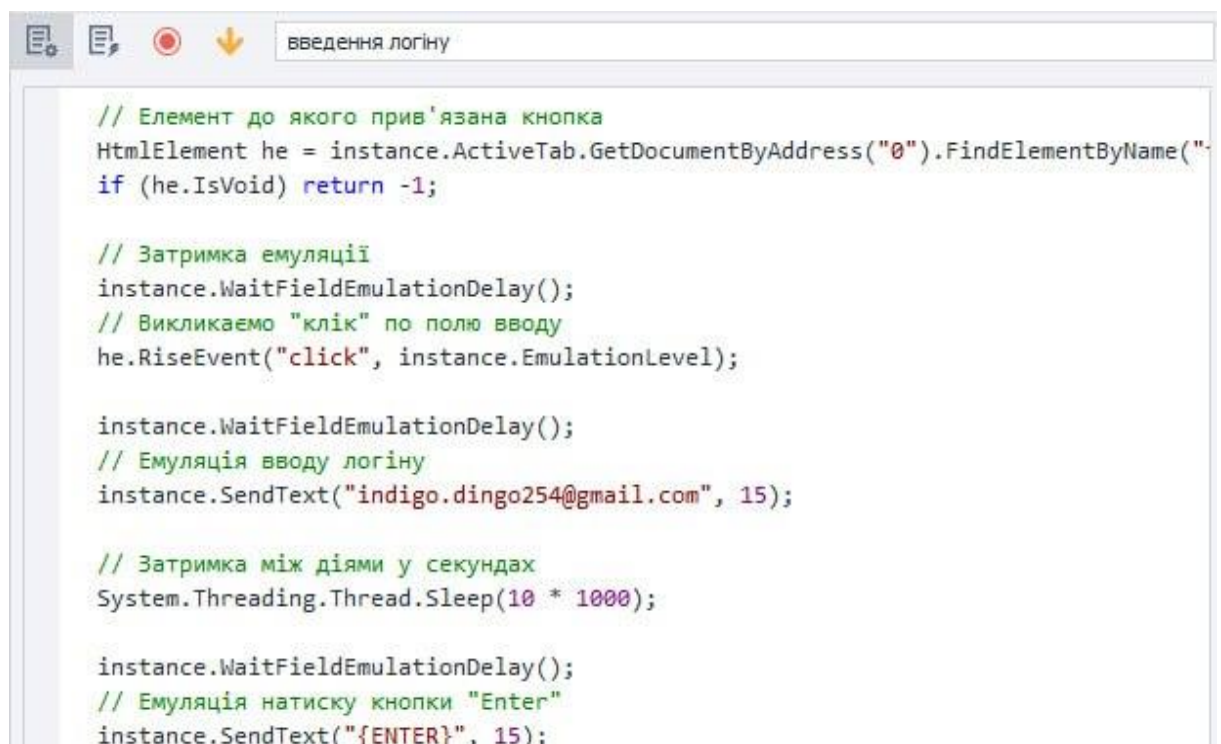


Рисунок 3.11. Вхід в X акаунт за допомогою емуляції

Під час цієї емуляції, бот виконує кроки звичайної авторизації. Тобто введення логіну та паролю, враховано також затримка для коректної роботи сторінки. Окрім цього, код виконує експорт даних в файл Netscape та зберігає його в локальній базі даних. Ця дія дозволить при майбутньому повторному використанню даної сторінки підвантажувати файли куки та пропускати авторизацію.

Далі відбувається перехід по посиланням, за допомогою натискання кнопок в режимі емуляції, до сторінки публікації посту. Під час постингу, бот

через API підтягує усі дані публікації та автоматично заповнює поля, що частково наведено на рисунку 3.12.

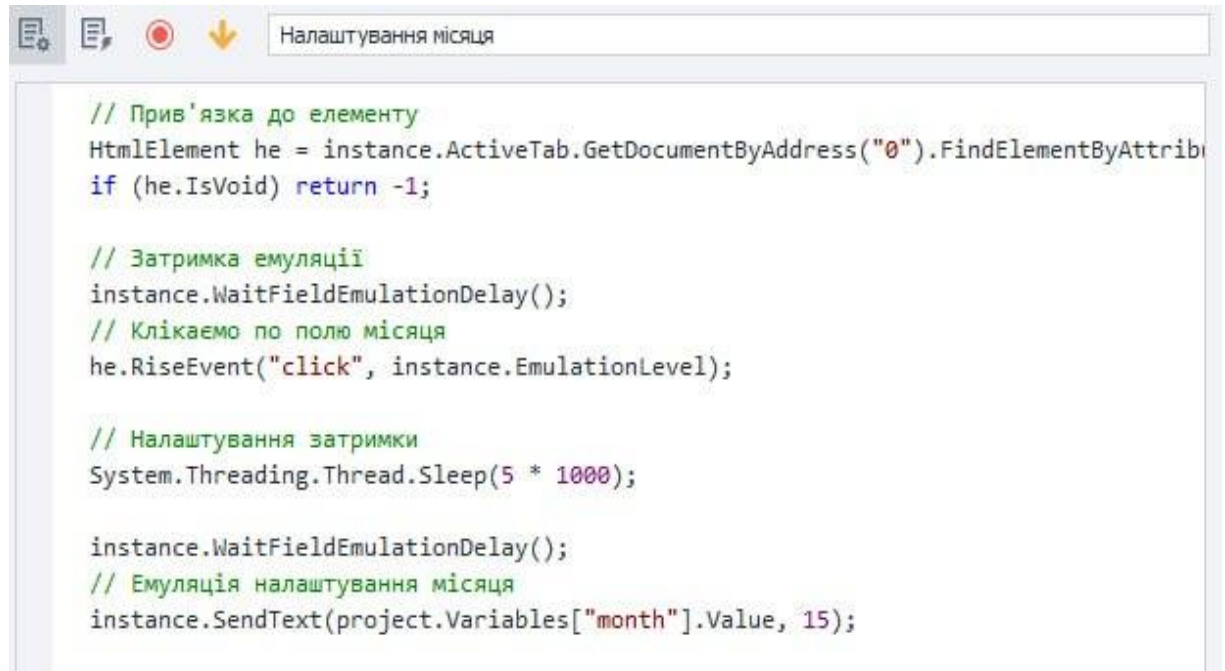


Рисунок 3.12. Автоматичне заповнення майбутньої публікації

Також бот повністю автономно налаштовує планування майбутньої публікації на дату, що вказана в базі даних.

В результаті чого, на виході отримуємо розробку, яка є автоматизованою та виключає людський фактор, вона є унікальною на сьогоднішній день з таким набором функцій.

ВИСНОВКИ ДО РОЗДІЛУ 3

У даному третьому розділі описується процес розробки та створення описаного продукту для автоматизації збору інформації, генерації контенту та автоматичного постингу публікації в соціальних мережах.

Розробка елементів системи, а саме веб-інтерфейс, модуль підключення по API, модуль обробки даних та інші всередині перелічених, створюють повноцінний застосунок для автоматизації роботи маркетолога, автоматизатора або розвитку бізнесу. Завдяки тому, що розробка складається з окремих легкопоеднуємих блоків, вона відкрита до розширення та допрацювання. Це дозволяє легко та швидко вносити корективи чи модернізувати застосунок.

У підрозділі безпосередньої розробки застосунку було описано етапи, за якими розроблювалася система, які компоненти були модернізовані, покращені або створені унікальні. Кожен з таких елементів має прямий вплив на швидкість та ефективність програми. Дана розробка була створена та модернізована для гарантування високої продуктивності та універсальності продукту.

РОЗДІЛ 4

ТЕСТУВАННЯ РОЗРОБКИ

4.1. Створення алгоритму тестування

Створення алгоритму тестування передбачає пошук тестових даних для роботи з розробкою, симуляція сценаріїв дій користувача та визначення критеріїв, за якими порівнюватимуться результати експериментів. В якості тестових даних було взято ключові слова з бази даних ключових слів SE Ranking[22]. Завдяки цьому, можна симулювати різноманітні запити користувачів до даної розробки та порівнювати релевантність результату роботи застосунку, що є одним з основних критеріїв ефективності роботи розробки.

Для проведення максимально ефективної оцінки роботи застосунку було обрано кілька сценаріїв поведінки користувача:

- Пошук даних для публікацій за кількома заданими темами
- Застосування парсингу для пошуку даних за однією конкретною темою
- Генерація публікації за малою кількістю ключових слів
- Застосування великого переліку ключових слів для генерації тексту
- Створення різної кількості публікацій

4.2. Створення алгоритму тестування

Під час тестування розробленого застосунку було проведено по 15 експериментів на кожен сценарій дій користувача. Така кількість експериментів потрібна для визначення унікальності створених публікацій та перевірки коректності та швидкості роботи програми під час різних варіацій роботи користувача з програмою.

Оскільки створена програма поєднує в собі кілька різноманітних функцій, які до цього часу не були реалізовані в одному повноцінний застосунку, то

експерименти були проведені над кожною частиною розробки. Та кожна така частина порівнюється з відповідною функцією найкращих аналогів відібраних в попередніх розділах.

Результати тестування було оцінено за такими критеріями:

- Оцінка унікальності – показує, що система генерує унікальний текст для публікацій навіть за однаковими ключовими словами
- Оцінка релевантності – показує, що система генерує текст відповідний до ключових слів чи теми
- Оцінка швидкості виконання
- Оцінка коректності перенесення даних під час симуляції роботи користувача в ролі публікатора
- Оцінка використовуваних ресурсів комп'ютера під час роботи з застосунком:
 - Пам'ять
 - Енергія
 - Використання ядер

Докладніше результати експериментів готової розробки описані нижче в таблиці 4.1.

Таблиця 4.1

Результати експериментів готової розробки

Сценарії	Унікальність	Релевантність	Швидкість	Коректність
Пошук даних для публікацій за кількома заданими темами	Дані унікальні	Створені публікації відповідають обраній темі	3,8 с	Створені публікації коректно заплановані
Застосування парсингу для пошуку даних за однією конкретною темою	Дані унікальні	Створені публікації відповідають обраній темі	3 с	Створені публікації коректно заплановані

Таблиця 4.1 (продовження)

Результати експериментів готової розробки

Сценарії	Унікальність	Релевантність	Швидкість	Коректність
Генерація публікації за малою кількістю ключових слів	Текст публікації повністю унікальний при кожній генерації	Створені публікації відповідають ключовим словам	1 с	Створені публікації коректно заплановані
Застосування великого переліку ключових слів для генерації тексту	Текст публікації повністю унікальний при кожній генерації	Створені публікації відповідають ключовим словам	1 с	Створені публікації коректно заплановані
Створення різної кількості публікацій	Текст публікації повністю унікальний при кожній генерації	Створені публікації відповідають ключовим словам	1,5 с	Створені публікації коректно заплановані

Далі в таблиці 4.2 описано результати отримані під час експериментів відповідних функцій аналогів розробленого продукту.

Також нижче описано кількість ресурсів комп'ютера, які затрачались під час проведення експериментів для тестування аналогів та готового застосунку, ці дані наведено в таблиці 4.4 та 4.3 відповідно.

Результати експериментів тестування аналогів розробки

Сценарії	Унікальність	Релевантність	Швидкість	Коректність
Пошук даних для публікацій за кількома заданими темами	Дані унікальні	3 з 15 публікацій на хибну тематику	15 с	12 з 15 публікацій створено коректно
Застосування парсингу для пошуку даних за однією конкретною темою	Дані унікальні	2 з 15 публікацій на хибну тематику	10 с	13 з 15 публікацій створено коректно
Генерація публікації за малою кількістю ключових слів	Немає унікальності	Створений текст складно зв'язаний з ключовими словами	3 с	Публікацій створено правильно
Застосування великого переліку ключових слів для генерації тексту	Трапляються повтори	Створені публікації відповідають ключовим словам	5 с	Публікацій створено правильно
Створення різної кількості публікацій	Немає унікальності	4 з 15 публікацій не релевантні	5,5 с	Публікацій створено правильно

Таблиця 4.3

Затрачені ресурси в результаті експериментів готової розробки

Сценарій	CPU	Memory	Energy
1	45%	810 MB	High (35,8% GPU)
2	33%	620 MB	High (32,4% GPU)
3	30%	311 MB	High (22,1% GPU)
4	36%	346 MB	High (26,2% GPU)
5	40%	489 MB	High (27,9% GPU)

Оскільки один сценарій виконувався 15 разів для достовірності результатів, то в таблиці вказано середнє значення використаних ресурсів за кожним сценарієм.

Таблиця 4.4

Затрачені ресурси в результаті експериментів тестування аналогів створеного продукту

Сценарій	CPU	Memory	Energy
1	80%	1,2 ГБ	High (48,6% GPU)
2	33%	620 MB	High (34,4% GPU)
3	32%	350 MB	High (25,3% GPU)
4	37%	396 MB	High (30,2% GPU)
5	44%	561 MB	High (26,8% GPU)

4.3. Огляд користувацького інтерфейсу розробки

Розглянемо інтерфейс програми, де користувач співпрацює з розробкою за допомогою веб-сторінки, що дозволяє правильну комунікацію між користувачем та розробкою.

Веб-інтерфейс має дві сторінки. Перша - це налаштування опцій для парсингу та планування публікацій, що показано на рисунку 4.1. Тут користувач може обрати потрібні теми, на які в подальшому орієнтується парсинг та формує публікації. Крім того, нижче, перейшовши по посиланню, користувач може переглянути детальніше сформовану інформацію та за необхідності відредагувати майбутні публікації або розклад до них.

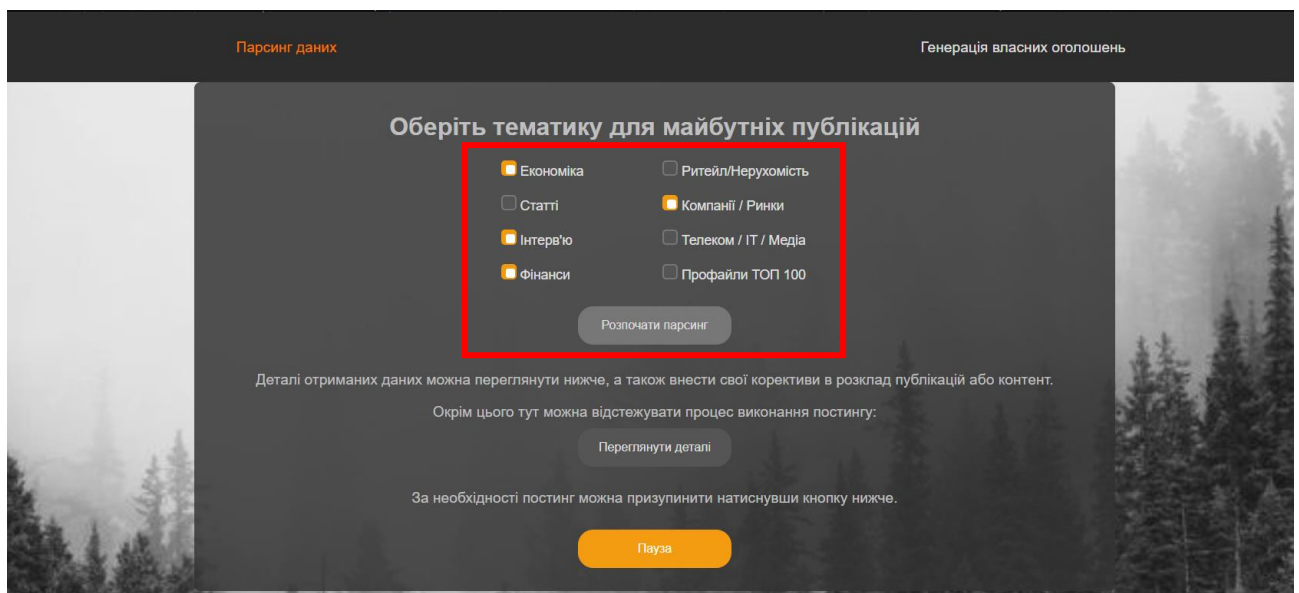


Рисунок 4.1. Сторінка налаштування парсингу та запуск постингу

Також, в даній таблиці, що зображено на рисунку 4.2, можна відстежувати процес виконання постингу в стовбці “Done”. Окрім цього, на сторінці присутня кнопка паузи, щоб призупинити процес публікування контенту, що зображено на рисунку 4.3.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Month	Day	Year	Hour (0-12)	Minute(00-59)	AM/PM	Text for tweet	Link on picture	Link on post	Done?	Error	Counter
2	November	17	2024	5	20	PM	У напівпрозорому тоні та	https://img.tsn.ua/cach	https://tsn.ua/lady/news	Done		0
3	November	17	2024	5	10	PM	"Я ледь не помер у черв"	https://img.tsn.ua/cach	https://tsn.ua/prosport/vi	Done		1
4	November	17	2024	5	47	PM	Для охорони Трампа вики	https://img.tsn.ua/cach	https://tsn.ua/tsikavinki/	Done		2
5	November	17	2024	6	23	PM	В Україні різко здешевша	https://img.tsn.ua/cach	https://tsn.ua/ukravina/v	Done		3
6	November	17	2024	6	50	PM	Про що свідчить сьогодні	https://img.tsn.ua/cach	https://tsn.ua/ukravina/p	Done		4
7	November	17	2024	7	26	PM	Якому знаку зодіаку пощ	https://img.tsn.ua/cach	https://tsn.ua/lady/astroi/	Done		5
8	November	17	2024	7	13	PM	Інвестиції у «зелену» ене	https://img.tsn.ua/cach	https://tsn.ua/groshi/inve	Done		6
9	November	17	2024	7	48	PM	Російські дрони могли по	https://img.tsn.ua/cach	https://tsn.ua/svit/rosiys/	Done		7
10	November	17	2024	8	51	PM	Шольц виправдався за с	https://img.tsn.ua/cach	https://tsn.ua/politika/sh	Done		8
11	November	17	2024	8	12	PM	Відсвяткували річницю: п	https://img.tsn.ua/cach	https://tsn.ua/lady/news/			9
12	November	17	2024	8	37	PM	Чоловік покійної Руслани	https://img.tsn.ua/cach	https://tsn.ua/qlamur/ch			10
13	November	17	2024	9	56	PM	На Вовчанському напрям	https://img.tsn.ua/cach	https://tsn.ua/ato/na-vov			11
14	November	17	2024	9	19	PM	Які цифри у даті народже	https://img.tsn.ua/cach	https://tsn.ua/other/yaki-			12
15	November	17	2024	9	10	PM	Як зробити волосся блис	https://img.tsn.ua/cach	https://tsn.ua/lady/kraso			13
16	November	17	2024	10	31	PM	Високі чоботи в осінньом	https://img.tsn.ua/cach	https://tsn.ua/lady/news/			14
17	November	17	2024	10	25	PM	"Наступні тижні будуть ви	https://img.tsn.ua/cach	https://tsn.ua/svit/nastur			15
18	November	17	2024	10	25	PM	Легендарний Холіфінд ві	https://img.tsn.ua/cach	https://tsn.ua/prosport/le			16
19	November	17	2024	11	28	PM	Масований обстріл енерг	https://img.tsn.ua/cach	https://tsn.ua/ukravina/n			17
20	November	17	2024	11	29	PM	Російські дипломати залп	https://img.tsn.ua/cach	https://tsn.ua/svit/rosiys/			18

Рисунок 4.2. Таблиця для відслідковування прогресу постингу з дательною інформацією

Парсинг даних

Процес публікації закінчено

Генерація власних оголошень

ОК

Оберіть тематику для майбутніх публікацій

☐ Економіка

☐ Ритейл/Нерухомість

☐ Статті

☐ Компанії / Ринки

☐ Інтерв'ю

☐ Телеком / IT / Медіа

☐ Фінанси

☐ Профайли ТОП 100

Розпочати парсинг

Деталі отриманих даних можна переглянути нижче, а також внести свої корективи в розклад публікацій або контент.

Окрім цього тут можна відстежувати процес виконання постингу:

Переглянути деталі

За необхідності постинг можна призупинити натиснувши кнопку нижче.

Пауза

Рисунок 4.3. Активація кнопки паузи

Друга сторінка веб-застосунку – рисунок 4.4 – це генерація власних публікацій. Тут користувачу потрібно ввести ключові слова, за якими далі буде створена публікація, за допомогою штучного інтелекту. За потреби можна згенерувати публікацію повторно, за такими ж ключовими словами. Система кожного разу створює унікальний текст, що продемонстровано на рисунках 4.5.

Рисунок 4.4. Сторінка генерації власної публікації

Далі потрібно задати дату та час за допомогою випадаючих списків, що зображено на рисунку 4.6 та 4.7 відповідно. Важливо зазначити, що система враховує поточний час і задати час публікації раніше поточного часу неможливо.

Рисунок 4.6. Вибір дати під час планування публікації

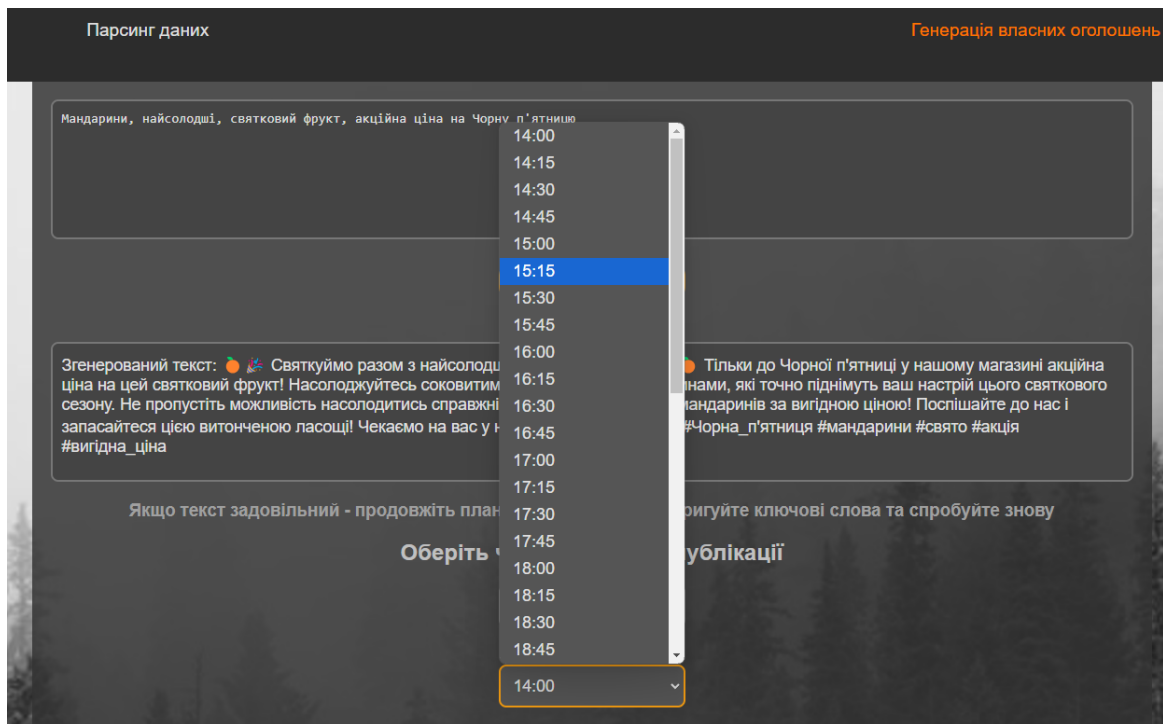


Рисунок 4.7. Вибір часу для планування публікації

Для остаточного завершення процесу створення власної публікації, потрібно натиснути кнопку «Запланувати», після чого дані публікації відправляються у таблицю та обробляються наступною частиною програми, яка автоматично планує її в X.

4.4. Оцінка результатів експериментів тестування

Проаналізувавши отримані результати під час проведення експериментів тестування даної розробки, можна зробити висновок, що завдяки застосуванню розроблених алгоритмів та модифікованих методів система працює точніше та ефективніше. Це показують експерименти з різним типом сценарію створення контенту для публікацій в соціальних мережах.

В фінальній версії застосунку експерименти показують, що при генерації власного контенту система створює повністю унікальний контент, на відмінну від аналогів застосунку, де трапляються повтори та не якісно згенеровані тексти. Крім того, в розробленому застосунку створюваний контент є релевантний при кожному експерименті, чого не було в інших експериментах варіанті. Це свідчить про точність та ефективність фінальної розробки.

Введіть через кому ключові слова майбутньої публікації

Годинник настінний, колір - золото, пропозиція тижня, механічний

Згенерувати

Згенерований текст: ✨ Відкрийте для себе неймовірний годинник настінний в кольорі золото - ідеальний акцент для вашого інтер'єру! Ця пропозиція тижня - механічний годинник з елегантним дизайном і бездоганним ходом. Придайте своєму приміщенню шарму та стилю з цим чудовим аксесуаром, який точно не залишить байдужим нікого! ✨ 📌 #годинник #золото #механічний #пропозиціядня

а)

Введіть через кому ключові слова майбутньої публікації

Годинник настінний, колір - золото, пропозиція тижня, механічний

Згенерувати

Згенерований текст: ✨ Неймовірна пропозиція тижня! ✨ 📌 Ви шукаєте найкращий годинник настінний для свого інтер'єру? Ми маємо щось особливе для вас - годинник настінний у кольорі золото! Його елегантний дизайн приверне увагу кожного, хто відвідає ваш дім. 📌 Цей годинник - не просто красивий предмет, а ще й механічний! Він не потребує батарейок і завжди буде показувати точний час. Крім того, його золотистий колір надасть вашому інтер'єру витонченості і шарму. Не втрачайте можливість отримати цей ексклюзивний годинник настінний за вигідною ціною! Поспішайте замовити його прямо зараз і насолоджуйтесь красою кожен день. 🌟 🌟 🌟

б)

Рисунок 4.3. Генерація унікальних публікацій

Не менш важливим фактором для оцінки ефективності роботи програми є її швидкість виконання. Завдяки модифікації деяких алгоритмів парсингу та форматування даних, а також розпаралелювання на оптимальну кількість потоків виконання програми, вдалось досягти набагато кращого результату, аніж був в аналогів.

Також, в порівнянні з попередніми результатами, розробка показала себе набагато ефективнішою в плані використання ресурсів комп'ютера, а саме, використовує меншу кількість пам'яті, є менш ресурсозатратним та менше споживає енергії, що на сьогоднішній день є досить актуальним, особливо під час відключень електроенергії.

Під час тестування не обійшлося і без виявлення проблем. Коли користувач намагається згенерувати власну публікацію і натискає кнопку генерації кілька разів, не чекаючи відповідь, то система генерує кілька разів відповідь, адже скрипт запускається стільки, скільки користувач натиснув кнопку. Проте, це не є критично, адже кожен раз генерується унікальний текст, і користувач в будь-якому випадку отримує відповідь. Хоча це є один із моментів, який в майбутньому повинен бути вдосконалений.

Отже, підсумовуючи результати експериментів, можна сказати, що створена система є ефективною, унікальною та практичною. Це говорить про те, що дана розробка може стати повноцінним інструментом створення контенту для маркетологів та різних бізнесів. Вона з легкістю може бути застосована в повсякденному житті як для окремих людей, так і для малого та середнього бізнесу. Дана розробка є першою, яка поєднує усі вище описані функції.

4.5. Стратегія вдосконалення та рекомендації по використанню

Після аналізу результатів тестування варто навести кілька рекомендацій для подальшого покращення розвитку на використання створеної системи. Такі рекомендації мають на меті покращити роботу системи, а саме, зробити більш зручною в використанні, підвищити швидкість та ефективність, покращити функціонал.

- План для випуску розробки на ринок:
 - Продумування та створення маркетингових ходів для відкриття розробки світу, вказуючи на її унікальність та ефективність.
 - Пошук партнерів та спонсорів для подальшого просування розробки в світ бізнесу.
 - Проведення конференцій та презентацій з демонстрацією високих можливостей створеного застосунку.
- Проектування варіантів покращення системи:

- Усунення знайдених недоліків, а саме, покращення взаємодії програми з користувачем, додавання більш чітких вказівок або створення умов для раціональної роботи програми.
- Створення авторизації для користувачів, що забезпечить приватність. Також варто додати історію сформованих запитів та відповідей, для зручності користувача, та можливість редагувати її.
- Додавання внутрішнього ком'юніті для зручності використання застосунку компаніями та передачі даних між кількома користувачами не виходячи з застосунку.
- Розвиток застосунку на ринку розробок:
 - Додавання інтеграції з іншими застосунками для бізнесу, таких як месенджери, програми для ведення аналітики та ін.
 - Створення більш різноманітного функціоналу, наприклад, генерація картинок або можливість попереднього перегляду створеної публікації прямо в застосунку.
 - Розробка можливих сповіщень чи створення нагадування для користувача про важливі моменти.

ВИСНОВКИ ДО РОЗДІЛУ 4

В даному четвертому розділі було описано процес проведення експериментів для тестування розробленого застосунку. Як результат, система показала себе достатньо ефективною в пошуку великого обсягу інформації, генерації унікальних публікацій за різними видами запитів та коректному плануванню публікацій в соціальних мережах.

Створений веб-застосунок працює відповідно до вказівок користувача. Хоча була знайдена невелика неточність при спробі користувача натиснути кнопку генерації публікації кілька разів підряд – публікація була згенерована кілька разів. Проте це ніяк не впливає на ефективність та точність роботи програми, але варто в майбутньому вдосконалити цей момент.

За практичність та ефективність програми також свідчать показники використаних ресурсів комп'ютера під час роботи з програмою, які показали хороші результати в порівнянні з іншими застосунками. Час виконання функцій розроблюваної програми в середньому у 3,7 разів менше ніж час виконання цих функцій у аналогічних застосунках, CPU в середньому використовується на 8,6% менше. В окремих випадках значно менше використовується пам'яті в даному застосунку, найбільша різниця між аналогічними експериментами становить 390 МБ. Також програма є ефективнішою ніж її аналоги за показником споживання енергії в середньому на 4,18%.

Також, в даному розділі було описано стратегії випуску даного застосунку на ринок продуктів, можливі шляхи покращення його функціоналу та алгоритм впровадження розробки в життя бізнесу та бізнес-проектів. 144,4

РОЗДІЛ 5

РОЗРОБКА СТАРТАП-ПРОЕКТУ

5.1. Інформаційна картка проекту

Таблиця 5.1, що наведена нижче описує інформаційну картку проекту.

Таблиця 5.1

Інформаційна картка проекту

№	Характеристика	Зміст характеристики
1	Назва проекту	Спосіб автоматичної генерації контенту в соцмережах
2	Автори проекту	Русанова Ольга Веніамінівна, Кучеренко Іван Дмитрович
3	Коротка анотація	Застосунок створений для повної автоматизації пошуку, генерації та постингу публікацій в соціальних мережах. Поєднавши два метод парсингу, що дозволило зробити систему більш універсальною та швидшою, та розробка власного алгоритму автоматизації, яка виконується в кілька потоків паралельно, система здатна виключити людський фактор із своєї роботи та автоматизувати весь процес.
4	Термін реалізації	Чотири квартали (12 місяців)
5	Необхідні фінансові ресурси	Зарплати співробітникам (помісячно) на етапі розробки: - Розробник = 2.700\$ кожен; - Web-дизайнер = 1.500\$; - Аналітик = 1.500\$; - Проект-менеджер = 2.900\$; - Тестувальник = 1.600\$; Всього щомісяця = 10.200\$; Інвестиції в обладнання та програмне забезпечення = 11.000\$; Ліцензії = \$70; Просування та реклама = 3.000\$. Хмарні ресурси для збереження інформації = \$120 щомісяця після запуску.
6	Необхідні матеріальні ресурси	- Потужні ПК з високою обчислювальною здатністю для тестування та безпосередньої розробки
7	Необхідні людино-ресурси	- 1 Розробник; - 1 Web-дизайнер; - 1 Аналітик; - 1 Проект-менеджер; - 1 Тестувальник.
8	Опис проблеми, яку вирішує проект	Проект зосереджений на вирішенні проблеми автоматизації пошуку інформації, генерації та публікації контенту для соціальних мереж на основі ключових слів, на тему бізнесу користувача, або на тему актуальних новин. Розробка передбачає використання алгоритмів автоматизації для збору даних з новинних ресурсів, формування змістовних постів та їх публікації в соціальних мережах з виключенням людського фактору.

Таблиця 5.1(продовження)

Інформаційна картка проекту

№	Характеристика	Зміст характеристики
9	Головні цілі та завдання	Основна мета проекту – розробка системи, за допомогою якої можна автоматично створювати публікації в соцмережі на основі парсингу новинних або аналізу ключових слів. Даний застосунок має на меті облегшити процес управління контентом та забезпечити регулярну публікацію з виключенням людського фактору. Завдання проекту: 1. Розробити системи, яка здатна автоматично збирати, аналізувати та обробляти інформацію з новинних ресурсів для формування контенту. 2. Провести тестування та валідацію створеного прототипу на реальних новинних даних або ключових словах, заданим користувачем, для перевірки точності, ефективності та надійності алгоритмів генерації публікацій. 3. Зробити оцінку результатів, проаналізувати релевантність отриманих результатів та виявити можливостей покращення та оптимізації системи для досягнення максимальної автоматизації.
10	Очікувані результати	- Кінцева версія застосунку: Розробка та тестування базової версії системи, яка демонструє основні функціональні можливості та концепцію проекту – автоматичне створення та публікація постів на соціальних платформах. - Адаптованість: Продукт має бути розроблений з урахуванням майбутньої реалізації адаптації та можливості масштабування. - Оптимізація продукту: Внесення покращень в систему на основі отриманого відгуку та результатів тестування для підвищення ефективності. ефективності. - Позитивний соціальний вплив: Проект повинен сприяти підвищенню ефективності роботи бізнесів і покращенню їх присутності в соціальних мережах, що повинно полегшити роботу людини. - Просування маркетингових стратегій: Створення початкової маркетингової стратегії для просування продукту на ринок, з метою залучення нових користувачів і поширення системи.

5.2. Склад команди стартап-проекту

У таблиці 5.2 наведено склад команди розробки.

Таблиця 5.2

Склад команди розробки

№	Роль	Спеціальність	Кількість
1	Розробник	Розробник	1 люд.
2	Web-дизайнер	Web-дизайнер	1 люд.
3	Дата-аналітик	Дата-аналітик	1 люд.
4	Менеджер	Project Manager	1 люд.
5	Тестувальник	Manual	1 люд.
Разом			5 людей

Колектив зібраний для даного стартапу містить 5 персон. Кожен з учасників колективу має свою унікальну задачу, яка спрямована на створення та розвиток розробки. Нижче в таблиці 5.3 розписано важливість роботи кожного з них.

У рамках оцінювання ефективності проекту важливо виділити ключові фактори, які впливають на його успіх. Кожен з цих факторів має свою вагу важливості, що допомагає визначити пріоритетність у процесі планування та реалізації.

Таблиця 5.3

Важливість факторів вкладення, які впливають на якість проекту

№	Фактор	Вага важливості (1-10 балів)
1	Створення та просування ідеї	6
2	Формулювання чіткого плану	9
3	Компетентність співробітників	9
4	Виконання заданих обов'язків	10
5	Гарна співпраця та комунікація	7

Останній етап оцінювання співробітників передбачає детальний аналіз їх індивідуального внеску у проект. Оцінка враховує участь кожного члена команди в різних аспектах проекту, що дозволяє зробити об'єктивні висновки про ефективність роботи команди в цілому. Використовуючи отримані дані можна графік для візуального відображення важливості внеску кожного із співробітника, що продемонстровано у таблиці 5.4.

Оцінка внеску кожного співробітника

Фактор	Розробник	WEB-дизайнер	Дата-аналітик	Менеджер	Тестувальник	Разом
Створення та просування ідеї	40	70	20	70	15	
Формулювання чіткого плану	26	15	45	88	10	
Компетентність співробітників	95	90	90	90	70	
Виконання заданих обов'язків	100	100	95	100	90	
Гарна співпраця та комунікація	70	75	50	83	40	
Разом	331	350	300	431	225	1637
Відсоток	20.2%	21.4%	18.3%	26.3%	13.7%	100%

Отже, кожен співробітник відіграє свою важливу роль в розробці продукту, тобто правильна продуктивна співпраця є головним в даному завданні.

5.3. Про основну ідею та мету проекту

Наш стартап націлений на створення інноваційної системи, що автоматизує процес генерації та публікації контенту в соціальних мережах. Основна концепція проекту полягає у розробці інструменту, який за допомогою штучного інтелекту і машинного навчання автоматично збирає, аналізує та формує пости, а також має можливість генерувати на основі ключових слів контент для користувача або новинних матеріалів.

Ця система надасть можливість компаніям і приватним користувачам автоматизувати свої маркетингові кампанії в соціальних мережах. Сервіс аналізує новини та актуальні події, створює контент відповідно до заданих параметрів, таких як тематика, цільова аудиторія та стиль. Таким чином, користувачі зможуть планувати і публікувати пости без необхідності витратити значні ресурси на ручне управління контентом.

Метою проекту є не просто автоматизація процесу публікації, а й створення інтелектуальної системи, здатної самостійно генерувати релевантний та своєчасний контент. Це рішення дозволить значно знизити витрати часу на

керування соціальними мережами, підвищити ефективність просування бізнесу та зробити процес планування постів простішим та доступнішим для будь-якого користувача, незалежно від його технічної підготовки.

Розробка орієнтована на широкий спектр користувачів, від малих підприємств до великих компаній, які прагнуть збільшити свою присутність в соціальних мережах, не витрачаючи при цьому значні ресурси на маркетингові активності.

Наш проект також спрямований на вирішення проблеми невчасного або нерелевантного контенту, який не відповідає інтересам цільової аудиторії. Система, яку ми розробляємо, забезпечить автоматичне планування публікацій, створення інформативних постів і підвищення загальної ефективності цифрового маркетингу.

Таким чином, наш продукт зможе зайняти провідні позиції на ринку автоматизованих інструментів для управління соціальними мережами, пропонуючи користувачам інноваційне рішення для оптимізації їхньої онлайн-присутності та просування.

Переваги та можливі обмеження продукту будуть детально розглянуті у таблиці 5.5 для подальшої оцінки та вдосконалення проекту.

Таблиця 5.5

Переваги та недоліки

№	Переваги	Недоліки
1	Оригінальність та інноваційний підхід	Високий рівень технічної складності
2	Орієнтованість на широку аудиторію	Необхідність постійних оновлень
3	Підвищення ефективності контент-менеджменту	Можлива конкуренція з існуючими рішеннями
4	Відповідність трендам автоматизації у цифровому маркетингу	
5	Легке налаштування для різних потреб бізнесу	
Сума	5	3

Для успішного втілення ідеї проекту необхідно провести детальний аналіз, в якому будуть виявлені як сильні сторони, так і можливі недоліки

техніко-економічних характеристик розробки. Особливу увагу слід звернути на слабкі місця, розробляючи стратегії для їх усунення або мінімізації, оскільки ці аспекти можуть вплинути на загальний успіх проекту. Аналіз характеристик у порівнянні з аналогами представлено в таблиці 5.6.

Таблиця 5.6

Аналіз характеристик у порівнянні з аналогами

№	Характеристика	Власна розробка	Аналоги			W	N	S
			1	2	3			
1	Не потребує додаткового обладнання (тільки ПК та встановлене ПЗ)	+		+				+
2	Доступна ціна для користувачів	+	+				+	
3	Технологічна перевага	+	+	+	+		+	
4	Зворотний зв'язок у реальному часі	+	+				+	
5	Широкий спектр доступних публікацій	+						+
6	Доступність як для окремих користувачів, так і для малого та середнього бізнесу	+		+				+
7	Легка та швидка масштабованість	+						+

Виходячи з результату проведеного дослідження переваг та недоліків створюваного стартапу, варто відзначити його важливі якості, такі як унікальність та інноваційність проекту, великий круг потенційних користувачів, легка масштабованість завдяки функціональній структурі та актуальність на сьогоднішній день.

Проте, звісно існують деякі особливості, в вигляді високого рівня технічної складності реалізації, необхідності постійних оновлень та можливої конкуренції зі схожими продуктами на ринку.

Результат аналізу характеристик у порівнянні з аналогами показує, що а даному стартапі є кілька вагомих переваг. Таких як, доступність, для використання та розробки даного проекту достатньо потужного ПК та певного програмного забезпечення, доступні ціни, висока технологічна функціональність, наявність широкого спектру актуальних публікацій,

доступність та корисність для окремих користувачів та бізнесів, можливість функціонального масштабування. Цей ряд переваг виділяє дану розробку серед лінійки існуючих.

5.4. Аналіз технологій розробки

Вибір технологій та інструментів є досить важливим початковим кроком для розробки проекту, від якого залежить подальша ефективність та функціональність розробки. Адже, якщо технології не будуть відповідати сучасним вимогам, то сам стартап не може бути новітнім.

Для цього оглянемо таблицю 5.7, де описано головні технології, які використовувались під час розробки проекту.

Таблиця 5.7

Технології

№	Технологія	Версія	Сфера використання	Доступність
1	Python	3.13.0	Мова програмування для написання парсингу	Безкоштовно
2	Visual Studio Code	1.94	Фреймворк для програмування на Python	Безкоштовно
3	ZennoPoster	7.18	Фреймворк для запуску проектів автоматизації	2.500 грн
4	Project Maker for ZennoPoster	Йде в комплекті з ZennoPoster	Фреймворк для написання алгоритмів автоматизації, з можливістю емуляції дій людини	2.500 грн
5	C#	9.0	Мова програмування для реалізацій функцій автоматизації	Безкоштовно
6	HTML	4.01	Гіпертекстова мова для реалізації веб-дизайна інтернет-сторінки	Безкоштовно

Дані засоби та мови програмування дозволяють розробити сучасний актуальний продукт, який може стати інноваційним рішенням автоматизації роботи людини. Дані технології дозволяють імплементувати в свою роботу інші функції або легко поєднуватися з іншими технологіями, що забезпечує простоту масштабованості розробленого продукту та його ефективність роботи.

5.5. Аналіз реальних шансів випуску продукту на ринок розробок

Варто розглянути декілька аспектів, щоб оцінити ринок та шанс продукту бути в топі серед своїх аналогів. Це дозволить оцінити перспективу та виявить особливості проекту, які варто розвивати або його унікальності. Дані характеристики описані в таблиці 5.8, 5.9 і 5.10.

Таблиця 5.8

Аналіз ринкових характеристик

№	Характеристика	Значення
1	Численність аналогів	6 проектів
2	Майбутня реклама	150 000 щомісячно (2.50\$ за 1000 показів = 375\$)
3	Динаміка ринку	Автоматизація роботи людини – передова технологія, розвиток бізнесу є також не менш важливою опцією
4	Особливості випуску розробки	Потреби в інвестиціях та оплаті праці співробітників (команд розробників)
5	Необхідні умови виконання проекту	- Висока релевантність вихідного контенту - Висока ефективність роботи програми - Унікальність контенту та оперативні оновлення
6	Рентабельність в своїй сфері	70-85% річних

Цільовою аудиторією нашого стартапу є власники бізнесів, маркетингові команди, SMM-фахівці та користувачі, які активно займаються просуванням в соціальних мережах і прагнуть спростити процес публікацій. Продукт розроблений для автоматизації створення, планування та публікації контенту, що дозволить оптимізувати час і ресурси на ведення соціальних платформ, з огляду на сучасні тенденції автоматизації та вимоги цифрового ринку.

Для грамотної реалізації стартапу, що займається автоматизацією публікацій, необхідно детально оцінити можливі ринкові ризики та перспективи, враховуючи конкурентне середовище, технологічні виклики та нові можливості для вдосконалення продукту.

Аналіз можливих загроз на ринку розробок

№	Загроза	Опис	Можливість уникнення загрози
1	Розробка новітніх технологій	Впровадження нових, більш продуктивних технологій автоматизації публікацій	Постійний моніторинг нових технологій і швидка інтеграція їх у продукт
2	Користувацькі відгуки	Негативні відгуки через збої або недостатню ефективність у плануванні постів	Оперативне реагування на фідбек та вдосконалення системи на основі відгуків
3	Юридичні обмеження	Введення нових законів, що регулюють контент у соцмережах	Аналіз законодавчих змін і своєчасна адаптація функціоналу до вимог законодавства

Таблиця 5.10

Аналіз ринкових перспектив

№	Фактор	Опис можливості	Відповідна реакція компанії
1	Пошук партнерства	Співпраця з платформами для соціальних мереж або аналітичними сервісами	Розвиток партнерських відносин для розширення можливостей інтеграції
2	Розширення функціоналу	Розробка нових функцій для автоматизації та оптимізації процесів публікацій	Планування та реалізація регулярних оновлень продукту
3	Глобалізація	Відкриття продукту на нових регіональних ринках	Створення нових стратегій глобального розширення з адаптацією до локальних потреб

Такий спосіб аналізу дозволяє чітко оцінити ринкові можливості, з якими може стикнутися стартап, що допомагає своєчасно адаптувати стратегію розвитку та впроваджувати нові можливості для утримання конкурентних переваг.

Детальний аналіз ринкових сегментів і конкурентних переваг продукту буде відображено в таблицях 5.11 та 5.12, що допоможе краще зрозуміти стратегію майбутнього зростання і розвитку продукту.

Таблиця 5.11

Аналіз ринкових сегментів

Характеристики конкуренції	Сфера впливу характеристики	Дії компанії для забезпечення конкурентоспроможності
Цінова	Залежить від готовності покупця витратити гроші на продукт чи сервіс.	Створення доступних тарифних планів та гнучкої системи ціноутворення, з урахуванням можливостей малого та великого бізнесу
Продуктова	Включає різні інструменти для автоматизації публікацій	Розширення функціоналу системи, пропонуючи інноваційні рішення для різних платформ соціальних мереж
Унікальна	Сегмент ринку, де конкуренція обмежена або нова технологія є рідкісною	Акцент на унікальність рішення, що пропонує автоматизацію та аналіз у реальному часі для ефективної роботи з контентом
Брендова	Вплив репутації та бренду на вибір користувачів	Побудова сильного бренду за рахунок якісного сервісу, постійних оновлень та підтримки користувачів
Локальна	Конкуренція на ринку локальних рішень для автоматизації контенту	Адаптація продукту до локальних потреб та специфіки ведення бізнесу в різних країнах, локалізація сервісу

Таблиця 5.12

Аналіз конкурентних переваг автоматизації публікацій

№	Перевага над конкурентами	Оцінка (1-20)	Ефективність
1	Технологічні інновації	19	Висока ефективність завдяки використанню сучасних технологій, що відповідають потребам сучасного ринку автоматизації.
2	Орієнтація на клієнтів	18	Висока, дозволяє забезпечити якісну підтримку користувачів та підвищувати їхню лояльність.
3	Гнучке ціноутворення	16	Середня, оскільки гнучка цінова політика важлива, але не є основним фактором для залучення клієнтів.

Для максимального охоплення цільової аудиторії, компанія повинна використовувати багатоканальну стратегію просування. Основний фокус – на цифрові канали, які дозволять ефективно взаємодіяти з бізнесами різного рівня.

- Соціальні мережі: Використання Facebook, LinkedIn та X для залучення бізнес-користувачів, які зацікавлені в автоматизації публікацій.
- Вебінари та онлайн-івенти: Проведення освітніх сесій щодо оптимізації контенту та автоматизації маркетингу.
- SEO та контент-маркетинг: Оптимізація сайту під пошукові системи та створення корисних статей для підвищення залученості клієнтів.
- Спеціалізовані виставки та конференції: Участь у заходах, пов'язаних із маркетингом та автоматизацією для розширення партнерської бази.

Цей аналіз дозволяє компанії адаптувати свої стратегії, щоб забезпечити конкурентоспроможність на ринку, використовуючи інноваційні підходи та зосереджуючи увагу на потребах цільової аудиторії.

5.6. Написання виробничого плану

Розробка плану виробничих робіт є майже найголовнішим аспектом для досягнення злагодженої та продуктивної роботи команди розробників. Це допоможе досягти найвищих результатів під час створення проекту та його просування. Такий план наведений у таблиці 5.13.

Таблиця 5.13

План виробничих робіт

№	Етап	Період						
		1-й рік розробки				2-й рік	3-й рік	N+ рік
		1-й кв.	2-й кв.	3-й кв.	4-й кв.			
1	Дослідження ринку	+						
2	Продумування ідеї проекту	+	+					
3	Проектування та розробка		+	+	+			
4	Тестування розробки			+	+	+		
5	Реклама та маркетинг				+	+	+	
6	Випуск на ринок					+		
7	Оптимізація та слідування за розвитком					+	+	+

Враховуючи те, що команда проекту працюватиме віддалено, то витрати на оренду офісу можна не враховувати. Під час етапу розробки передбачається

одноразова витрата на інфраструктуру — \$10,000 для забезпечення основного обладнання та річні витрати на програмне забезпечення та ліцензії для розробників — \$500 на кожного члена команди. Такий підхід дозволяє оптимізувати фінансові ресурси та направити їх на ключові етапи розвитку продукту та маркетингові активності.

Проаналізуємо витрати на залучення команди та оплату їхньої праці протягом кількох років, що відображено в таблиці 5.14.

Таблиця 5.14

Витрати на оплату праці команди

№	Категорія працівників	Чисельність	Заробітна плата за місяць, тис. \$	Податки (ФОП, 5% від ЗП)	Витрати на оплату праці, тис. \$		
					1 рік	2 рік	N+ рік
1	Розробник	1	3	150	36	36	36
2	WEB-дизайнер	1	2.3	115	27.6	27.6	27.6
3	Менеджер	1	2	100	24	24	24
4	Маркетолог	1	2.1	105	25.2	25.2	25.2
5	Аналітик	1	2.2	110	26.4	26.4	26.4
Всього		5	11.6	580	139.2	139.2	139.2

Прогнозовані витрати на оплату праці команди допоможуть ефективно розподілити фінансові ресурси та забезпечити стабільний розвиток проекту.

ВИСНОВОК ДО РОЗДІЛУ 5

В даному третьому розділі було детально проаналізовано ключові аспекти розроблюваного проекту, спрямованого на автоматизацію створення та публікації контенту для соціальних мереж. Особливу увагу приділено аспектам конкурентоспроможності, ринковій стратегії та етапам впровадження рішення.

Було проведено аналіз конкурентного середовища, зокрема оцінено існуючі рішення на ринку, їх функціональні можливості, цінову політику та рівень задоволення потреб користувачів. Основні фактори, які забезпечують конкурентоспроможність нашого проекту, включають інноваційність у використанні комп'ютерного зору, автоматизацію процесів, а також високу ефективність у створенні контенту. Водночас було визначено виклики, зокрема можливий дефіцит ресурсів на етапі розширення та ризики, пов'язані з конкуренцією у швидкозростаючій галузі.

Також створено детальну модель реалізації проекту, яка охоплює всі етапи — від початкових досліджень і розробки системи до запуску продукту на ринок. Окреслено необхідні ресурси, часові рамки для виконання кожного етапу, а також ключові завдання для кожного члена команди, що сприятиме ефективному розподілу обов'язків та ресурсів.

Таким чином, цей аналіз надає чітке розуміння структури проекту, його конкурентних переваг та плану розвитку, забезпечуючи надійну основу для успішної реалізації та виходу на ринок.

ВИСНОВКИ

Під час написання даної магістерської роботи було досліджено великий обсяг ринку бізнес-застосунків та методів автоматичного просування продукту в соціальних мережах, проаналізовано основні необхідні методи та алгоритми для розробки та проведено їх оптимізацію для створення власного застосунку.

Перший розділ був присвячений широкому аналізу існуючих рішень та методів, що вони використовують. Саме це допомогло визначити основний напрямок робіт для розробки унікального інноваційного рішення та допомогло сформулювати задачі на наступні розділи.

Другий розділ це розгляд основних методів та алгоритмів для автоматизації, парсингу та генерації публікацій. Їх порівняння допомогло визначити переваги та недоліки кожного та вибрати оптимальні для покращення та створення власного. Крім того, огляд існуючих технологій допоміг розробити концепцію та алгоритм створення окремих частин розробки, також це допомогло визначити напрямки покращення.

У третьому розділі магістерської роботи описано процес розробки власних алгоритмів на основі знайдених в попередньому розділі. А саме, розробка парсингу, автоматизованого постингу, генерації контенту за допомогою штучного інтелекту, обробка даних, розробка веб-інтерфейсу. Усі ці алгоритми були детально оглянуті та оптимізовані для досягнення високої ефективності та точності роботи розробки.

У четвертому розділі описано проведені експерименти для тестування ефективності системи. Результати експериментів показали, що дана розробка є унікальною, ефективною, точною та чітко виконує свій функціонал. Крім того, в даному розділі було описано стратегію подальшого розвитку застосунку та випуску його на ринок.

П'ятий розділ представляє собою розробку стартап-проекту, де детально описано кожен крок розвитку та кількість ресурсів, які повинні бути враховані під час його застосування. Це дозволяє прорахувати різні аспекти проекту,

наприклад, конкурентоспроможність, актуальність, вартість, алгоритм реалізації та стратегію просування, саме це, забезпечує правильність впровадження створеного продукту на ринок.

Отже, результати даної магістерської роботи показують, що розроблений застосунок має великий потенціал для випуску на ринок ІТ-продуктів в галузі автоматизації та застосування його в малому та середньому бізнесі. Майбутня модифікація та покращення даної розробки може повністю автоматизувати роботу певних співробітників в сфері бізнесу та маркетингу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. SMM-маркетинг [Електронний ресурс] / Павло Власюк // СМО. – 2024. – Режим доступу до ресурсу: <https://wedex.com.ua/blog/shho-take-smm-marketyng/>.
2. Застосування штучного інтелекту в маркетингу [Електронний ресурс] // Ranktracker. – 2024. – Режим доступу до ресурсу: <https://www.ranktracker.com/uk/blog/application-of-artificial-intelligence-to-marketing/>.
3. Hootsuite [Електронний ресурс] // Hootsuite – Режим доступу до ресурсу: <https://www.hootsuite.com/>.
4. Buffer [Електронний ресурс] // Buffer – Режим доступу до ресурсу: <https://buffer.com/>.
5. Automation for business and home [Електронний ресурс] // ifttt – Режим доступу до ресурсу: <https://ifttt.com/>.
6. МЕТОДОЛОГІЯ НАУКОВИХ ДОСЛІДЖЕНЬ У ГАЛУЗІ [Електронний ресурс] – Режим доступу до ресурсу: https://ela.kpi.ua/bitstream/123456789/45871/1/Metod_doslid.pdf
7. Technical documentation [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Technical_documentation.
8. Visual Studio Code with GitHub Copilot [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/>.
9. Python is a programming language [Електронний ресурс]. – 2001. – Режим доступу до ресурсу: <https://www.python.org/>.
10. Basic capabilities of ZennoPoster [Електронний ресурс] // ZennoLab. – 2008. – Режим доступу до ресурсу: <https://zennolab.com/en/products/zennoposter/>.
11. Most popular autoposting services [Електронний ресурс] // Юлія Канаш. – 2022. – Режим доступу до ресурсу: <https://www.getresponse.com/en/blog/servisy-avtopostinga>.

12.API testing [Электронный ресурс] – Режим доступа до ресурсу: <https://qagroup.com.ua/publications/korotko-pro-ari-ta-jogo-testuvannia/>.

13.What is a Headless Browser: Top 8 Options for 2024 [Pros vs. Cons] [Электронный ресурс] // Ismail Ajagbe. – 2024. – Режим доступа до ресурсу: <https://www.scrapingbee.com/blog/what-is-a-headless-browser-best-solutions-for-web-scraping-at-scale/>.

14.What is HTML DOM? [Электронный ресурс] // Aspose Pty Ltd. – 2001. – Режим доступа до ресурсу: <https://docs.aspose.com/html/net/tutorial/what-is-html-dom/>.

15.Introduction to the DOM [Электронный ресурс] // mozilla.org contributors. – 1998. – Режим доступа до ресурсу: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction.

16.Parsing algorithm based on regular expressions [Электронный ресурс] – Режим доступа до ресурсу: <https://jarch.donnu.edu.ua/article/view/11534/11415>.

17.How to use XPATH and CSS for free data parsing [Электронный ресурс] // iDatica. – 2022. – Режим доступа до ресурсу: <https://idatica.com/blog/xpath-css-dlaya-parsing/>.

18.Headless Browser: What is It and How to Use It for Scraping [Электронный ресурс] // Team Froxy. – 2024. – Режим доступа до ресурсу: <https://blog.froxy.com/en/headless-browser-what-is-it-and-how-to-use-it-for-scraping>.

19.Search for an open API of the site [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://habr.com/ru/articles/680320/>.

20.Customizing Visual Studio Code for the layout designer [Электронный ресурс] – Режим доступа до ресурсу: <https://liquidhub.ru/blogs/blog/nastroyka-visual-studio-code>.

21.Visual Studio Code extensions [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://learn.microsoft.com/uk-ua/power-apps/maker/portals/vs-code-extension>.

22. Research Keywords [Електронний ресурс] – Режим доступу до ресурсу:

https://online.seranking.com/research.keywords.html/?keyword=%D0%90%D0%BA%D1%86%D1%96%D1%8F&source=ua&user_group=0&landing=keyword_suggestion_tool&lang=ua.

ДОДАТОК А

Спосіб автоматичної генерації контенту в соцмережах

ЛІСТИНГ ПРОГРАМНОГО КОДУ

Аркушів 11

```

import requests
from bs4 import BeautifulSoup as bs
import pandas as pd
import gspread
import random
import time
from datetime import *

# Отримання часових даних
date_now = datetime.now()
today = date_now.strftime("%d %B %Y (%A)")
today_mass = today.split(' ')
day_today = today_mass[0]
month_today = today_mass[1]
year_today = today_mass[2]
hour_now = date_now.hour
print('Today is', today)

# Підготовка до авторизації
SHEET_ID = '1TN7uuQprkXBrPeXct8kPiz9LDHPLLeRGZOMrprD2G9g'
SHEET_NAME = 'List1'
# Завантаження та підключення до Google Sheets
credentials_file = 'sheet-automation-zen-c6b81725474c (3).json'
gc = gspread.service_account(credentials_file)
spreadsheet = gc.open_by_key(SHEET_ID)
worksheet = spreadsheet.worksheet(SHEET_NAME)
rows = worksheet.get_all_records()

print('\n===== General view of the dataframe =====')
df = pd.DataFrame(rows)
print(df.tail())

# Реалізація запиту сторінки з новинами
URL_TEMPLATE = "https://biz.nv.ua/ukr/publications.html"
r = requests.get(URL_TEMPLATE)
print('\nStatus request =', r.status_code)

print('\n\n===== Parsing of news feeds =====')
print('\nParsing links:')
soup = bs(r.text, "html.parser")
links_par = soup.find_all('a', class_='row-result-body')

links = []
for name in links_par:
    links.append(name.attrs['href'])

for n in range(len(links)):
    print(links[n], '\n')

def zaglavie(url_zaglavie):

```

```

rs = requests.get(url_zaglavie)
root = bs(rs.content, 'html.parser')
article = root.find('h1')
import time
time.sleep(0.5)
title = []
title.append(article.text)
return title

text_pars = []
for l in range(len(links)):
    url_zaglavie = links[l]
    text_pars.append(zaglavie(url_zaglavie))
    l += 1
for n in range(len(text_pars)):
    print(text_pars[n], '\n')

# Функція для парсинга фото
def mass_for_photo(lk):
    p = requests.get(lk)
    soup = bs(p.text, 'html.parser')

    images = soup.find('div', class_='article__content__head_img')
    import time
    time.sleep(1.5)
    photo1 = []

    if images:
        image_tag = images.find('img')
        if image_tag:
            img_url = image_tag['src']
            print("Image: ", img_url)
            photo1.append(img_url)
        else:
            print("Image tag not found")
    else:
        print("Div not found")

    div_not_f = soup.find('div', class_="longread-header-image")
    if div_not_f:
        img_tag1 = div_not_f.find('img')
        if img_tag1:
            img_url1 = img_tag1['src']
            print('Image URL found:', img_url1)
            photo1.append(img_url1)
        else:
            print("Not found all")

    return photo1

```

```

photo_all = []
for o in range(len(links)):
    lk = links[o]
    photo_all.append(mass_for_photo(lk))
    o += 1

for m in range(len(photo_all)):
    print(photo_all[m], '\n')

# Формування масиву та запис отриманих даних в базу даних
counter = len(df['Counter'])
time_format = ['AM', 'PM']
row = []

text_pars1 = [item[0] for item in text_pars]
text_pars2 = [item.replace('\xa0', ' ') for item in text_pars1]

# Виправлення: перевіряємо наявність елементів перед доступом до першого
photo_all1 = [item[0] if item else "No image available" for item in photo_all]

# print('text_pars-----', text_pars2)
# print('photo_all-----', photo_all1)
# print('links-----', links)

for i in range(len(links)):
    if hour_now < 12:
        row = [month_today, day_today, year_today, hour_now + 1, random.randint(10,
59),
                time_format[0], text_pars2[i], photo_all1[i], links[i], '', '',
counter]
        worksheet.append_row(row)
        counter += 1
    else:
        row = [month_today, day_today, year_today, hour_now - 11,
random.randint(10, 59),
                time_format[1], text_pars2[i], photo_all1[i], links[i], '', '',
counter]
        worksheet.append_row(row)
        counter += 1

print("Counter =", counter)
import time
time.sleep(60)

<!DOCTYPE html>
<html lang="ru">
<head>

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Генерація власних оголошень</title>
<style>
    * {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
    }

    html, body {
        height: 100%;
        width: 100%;
        font-family: Arial, sans-serif;
        overflow-x: hidden;
    }

    body {
        background-color: #222;
        background-image:
url('https://i.pinimg.com/736x/4e/63/46/4e6346accffe7bb2c2f796c546c73540.jpg');
        background-size: cover;
        background-repeat: no-repeat;
        background-position: center;
        transition: background-position 0.1s ease-in-out;
    }

    .container {
        width: 80%;
        max-width: 1200px;
        margin: 0 auto;
        position: relative;
        z-index: 2;
        background-color: #333333d7;
        padding: 20px;
        border-radius: 8px;
        display: flex;
        flex-direction: column;
        align-items: center;
        justify-content: flex-start;
    }

    .menu-bar {
        display: flex;
        width: 100%;
        background-color: #2C2C2C;
        border: none;
        position: sticky;
        top: 0;
        z-index: 10;

```

```

        padding: 0;
    }

    .menu-button {
        flex: 1;
        padding: 20px;
        background-color: #2C2C2C;
        color: #e0e0e0;
        border: none;
        cursor: pointer;
        font-size: 1.2em;
        text-align: center;
        transition: background-color 0.3s, color 0.3s, transform 0.3s, box-
shadow 0.3s;
        width: 200px;
        height: 70px;
        border-radius: 20px;
    }

    .menu-button.active {
        color: #ff7700;
    }

    .menu-button:hover {
        background-color: #555;
        transform: scale(1.1);
        box-shadow: 0 0 15px rgba(255, 255, 255, 0.3);
    }

    h1 {
        color: #c0c0c0;
        margin: 20px 0;
        text-align: center;
    }

    select {
        padding: 10px;
        margin: 20px 0;
        background-color: #444;
        color: #e0e0e0;
        font-size: 1.1em;
        border-radius: 8px;
        border: 2px solid #777;
        width: 200px;
        transition: background-color 0.3s ease, border-color 0.3s ease;
    }

    select:focus {
        border-color: #f39c12;
        background-color: #555;
    }

```

```

    }

    button {
        padding: 10px 20px;
        margin: 20px 0;
        background-color: #f39c12;
        color: #ffffff;
        border: none;
        cursor: pointer;
        font-size: 1em;
        width: 200px;
        height: 50px;
        transition: transform 0.3s ease-in-out, background-color 0.3s ease;
        border-radius: 20px;
    }

    button:hover {
        background-color: #777;
        transform: translateY(-5px);
    }

    .output-window {
        width: 100%;
        height: 150px;
        margin-top: 20px;
        padding: 10px;
        border-radius: 8px;
        border: 2px solid #777;
        background-color: #444;
        color: #e0e0e0;
        font-size: 1.1em;
        resize: none;
        overflow-y: auto;
    }

    textarea {
        width: 100%;
        height: 150px;
        margin-top: 20px;
        padding: 10px;
        border-radius: 8px;
        border: 2px solid #777;
        background-color: #444;
        color: #e0e0e0;
        font-size: 1.1em;
        resize: none;
    }

</style>
</head>

```

```

<body>

  <div class="menu-bar">
    <button class="menu-button"
onclick="window.location.href='webmorda.html'">Парсинг даних</button>
    <button class="menu-button active"
onclick="window.location.href='ownpost.html'">Генерація власних оголошень</button>
  </div>

  <div class="container">
    <h1>Створіть власне оголошення</h1>

    <h3 style="color: #a0a0a0;">Введіть через кому ключові слова майбутньої
публікації</h3>

    <textarea id="inputText" placeholder="Ключові слова (наприклад: Акція,
Диван-софа, 10000грн)"></textarea>

    <button onclick="processText()">Згенерувати</button>

    <div class="output-window" id="outputText">Текст майбутньої
публікації</div>

    <h3 style="color: #a0a0a0; margin-top: 20px;">Якщо текст задовільний -
продовжіть планування, якщо ні - скоригуйте ключові слова та спробуйте знову</h3>

    <h2 style="color: #c0c0c0; margin-top: 20px;">Оберіть час та дату для
публікації</h2>

    <select id="datePicker"></select>

    <select id="timePicker"></select>

    <button onclick="confirmDateTime()">Запланувати</button>
  </div>

  <script>
    async function processText() {
      const inputText = document.getElementById("inputText").value;
      const outputWindow = document.getElementById("outputText");

      if (inputText.trim() !== "") {
        const kontekst = "Згенеруй, будь-ласка, красивий пост за даними ключовими
словами: "; // Это можно изменить
        const fullText = kontekst + inputText;

        try {
          const response = await
fetch("https://api.openai.com/v1/chat/completions", {
            method: "POST",

```

```

        headers: {
            "Authorization": "Bearer sk-proj-3-
diaGhqjX84xPPhhG11I8iOVzArYuiJNwThgExY513_Lq7SPj22LtWX7TZE3Rdq6iZUsAxW88T3BlbkFJata
XhJAys5zP8sPxNoz0kZzrcSw98IhWZBu7Sb5ptxo2NcSILVjDpDo-pCW_OofEBBtUGln-IA", //
Замініть на свій API-ключ OpenAI
            "Content-Type": "application/json",
        },
        body: JSON.stringify({
            model: "gpt-3.5-turbo",
            messages: [{
                role: "user",
                content: fullText
            }]
        })
    });

    const result = await response.json();
    if (response.ok) {
        outputWindow.textContent = `Згенерований текст:
\n${result.choices[0].message.content}`;
    } else {
        outputWindow.textContent = result.error ? result.error.message :
"Упс, сталася помилка..";
    }
} catch (error) {
    outputWindow.textContent = "Помилка при генерації, спробуйте ще.";
}
} else {
    outputWindow.textContent = "Введіть ключові слова вище.";
}

}

function confirmDateTime() {
    const selectedDate = document.getElementById("datePicker").value;
    const selectedTime = document.getElementById("timePicker").value;
    const confirmationMessage = `Публікація запланована на ${selectedDate}
o ${selectedTime}`;
    alert(confirmationMessage);
}

function setDateTimeOptions() {
    const now = new Date();
    const currentDate = now.toISOString().split('T')[0]; // Отримуємо
поточну дату у форматі YYYY-MM-DD
    const currentTime = now.toTimeString().split(' ')[0].slice(0, 5); //
Поточний час у форматі HH:mm

    const dateSelect = document.getElementById("datePicker");
    const timeSelect = document.getElementById("timePicker");

```

```

// Генерація доступних дат
for (let i = 0; i < 100; i++) {
    let optionDate = new Date(now);
    optionDate.setDate(now.getDate() + i);
    let formattedDate = optionDate.toISOString().split('T')[0];
    let optionElement = document.createElement("option");
    optionElement.value = formattedDate;
    optionElement.innerText = formattedDate;
    dateSelect.appendChild(optionElement);
}

// Встановлюємо мінімальне значення для дати
dateSelect.value = currentDate;

// Генерація часу залежно від обраної дати
dateSelect.addEventListener("change", updateTimeOptions);

updateTimeOptions();

function updateTimeOptions() {
    const selectedDate = dateSelect.value;

    // Очищаємо список часу
    timeSelect.innerHTML = "";

    let startHour = 0; // Почнемо з 00:00 для завтрашнього дня і
    пізніше

    if (selectedDate === currentDate) {
        // Якщо сьогодні, починаємо з поточного часу
        startHour = now.getHours();
        const currentMinute = now.getMinutes();
        const nextMinute = Math.ceil(currentMinute / 15) * 15;
        if (nextMinute < currentMinute) {
            startHour++;
        }
        // Додамо тільки доступні хвилини
        for (let hour = startHour; hour < 24; hour++) {
            for (let minute = 0; minute < 60; minute += 15) {
                let formattedTime = `${String(hour).padStart(2,
'0'))}:${String(minute).padStart(2, '0'))}`;
                if (formattedTime >= currentTime) {
                    let optionElement =
document.createElement("option");
                    optionElement.value = formattedTime;
                    optionElement.innerText = formattedTime;
                    timeSelect.appendChild(optionElement);
                }
            }
        }
    }
}

```

```

    } else {
        // Для всіх наступних днів починаємо з 00:00
        for (let hour = 0; hour < 24; hour++) {
            for (let minute = 0; minute < 60; minute += 15) {
                let formattedTime = `${String(hour).padStart(2,
'0'))}:${String(minute).padStart(2, '0')}`;
                let optionElement = document.createElement("option");
                optionElement.value = formattedTime;
                optionElement.innerText = formattedTime;
                timeSelect.appendChild(optionElement);
            }
        }
    }
}

window.onload = setDateTimeOptions;

// Функція для ефекту паралакса
document.addEventListener('mousemove', function (e) {
    const x = (e.clientX / window.innerWidth) - 0.5;
    const y = (e.clientY / window.innerHeight) - 0.5;
    document.body.style.backgroundPosition = `${x * 50}px ${y * 50}px`;
});
</script>

</body>
</html>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Веб-Интерфейс</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        html, body {
            height: 100%;
            width: 100%;
            font-family: Arial, sans-serif;
            overflow-x: hidden;
        }
    </style>

```

```

body {
    background-color: #222;
    background-image:
url('https://i.pinimg.com/736x/4e/63/46/4e6346accffe7bb2c2f796c546c73540.jpg');
    background-size: cover;
    background-repeat: no-repeat;
    background-position: center;
    transition: background-position 0.1s ease-in-out;
}

.container {
    width: 80%;
    max-width: 1200px;
    margin: 0 auto;
    position: relative;
    z-index: 2;
    background-color: #333333d7;
    padding: 20px;
    border-radius: 8px;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: flex-start;
}

.menu-bar {
    display: flex;
    width: 100%;
    background-color: #2C2C2C;
    border: none;
    position: sticky;
    top: 0;
    z-index: 10;
    padding: 0;
}

.menu-button {
    flex: 1;
    padding: 20px;
    background-color: #2C2C2C;
    color: #e0e0e0;
    border: none;
    cursor: pointer;
    font-size: 1.2em;
    text-align: center;
    transition: background-color 0.3s, color 0.3s, transform 0.3s, box-
shadow 0.3s;
    width: 200px;
    height: 70px;
}

```

```

        border-radius: 20px;
    }

    .menu-button.active {
        color: #ff7700;
    }

    .menu-button:hover {
        background-color: #555;
        transform: scale(1.1);
        box-shadow: 0 0 15px rgba(255, 255, 255, 0.3);
    }

    h1 {
        color: #c0c0c0;
        margin: 20px 0;
        text-align: center;
    }

    .checkbox-group {
        display: grid;
        grid-template-columns: repeat(2, 1fr);
        column-gap: 20px;
        row-gap: 10px;
        margin-bottom: 20px;
    }

    .additional-text {
        margin-top: 20px;
        color: #c0c0c0;
        font-size: 1.2em;
        text-align: center;
    }

    label {
        display: block;
        margin: 5px 0;
        color: #e0e0e0;
        font-size: 1.1em;
    }

    /* Стили для чекбоксов */
    input[type="checkbox"] {
        appearance: none;
        -webkit-appearance: none;
        -moz-appearance: none;
        width: 20px;
        height: 20px;
        background-color: #555;
        border-radius: 5px;
    }

```

```

border: 2px solid #777;
position: relative;
cursor: pointer;
transition: background-color 0.3s ease, border-color 0.3s ease;
}

/* Подсветка для чекбокса при активации */
input[type="checkbox"]:checked {
  background-color: #f39c12; /* Оранжевый цвет */
  border-color: #f39c12;
}

input[type="checkbox"]:checked::after {
  content: "";
  position: absolute;
  top: 4px;
  left: 4px;
  width: 12px;
  height: 12px;
  background-color: white;
  border-radius: 3px;
}

button {
  padding: 10px 20px;
  margin: 10px;
  background-color: #555;
  color: #e0e0e0;
  border: none;
  cursor: pointer;
  font-size: 1em;
  width: 200px;
  height: 50px;
  transition: transform 0.3s ease-in-out, background-color 0.3s ease;
  border-radius: 20px;
}

button:hover {
  background-color: #777;
  transform: translateY(-5px);
}

.pause-button {
  margin-top: 30px;
  background-color: #f39c12;
  color: #fff;
}

.pause-button:hover {
  background-color: #e67e22;
}

```

```

        transform: translateY(-5px);
    }

</style>
</head>
<body>

    <div class="menu-bar">
        <button class="menu-button active" onclick="switchTab('page1')">Парсинг
даних</button>
        <button class="menu-button"
onclick="window.location.href='ownpost.html'">Генерація власних оголошень</button>
    </div>

    <div class="container">
        <h1>Оберіть тематику для майбутніх публікацій</h1>

        <div class="checkbox-group">
            <label><input type="checkbox" name="option1"> Економіка </label>
            <label><input type="checkbox" name="option2">
Ритейл/Нерухомість</label>
            <label><input type="checkbox" name="option3"> Статті </label>
            <label><input type="checkbox" name="option4"> Компанії / Ринки </label>
            <label><input type="checkbox" name="option5"> Інтерв'ю </label>
            <label><input type="checkbox" name="option6"> Телеком / IT / Медіа
</label>
            <label><input type="checkbox" name="option6"> Фінанси </label>
            <label><input type="checkbox" name="option6"> Профайли ТОП 100</label>
        </div>

        <button onclick="startFunction()">Розпочати парсинг</button>

        <p class="additional-text">Деталі отриманих даних можна переглянути нижче,
а також внести свої корективи в розклад публікацій або контент. </p>
        <p class="additional-text">Окрім цього тут можна відстежувати процес
виконання постингу: </p>

        <button
onclick="window.location.href='https://docs.google.com/spreadsheets/d/1BygECCWT01jf
Z_bzets3d7gxWKP1WH9IjHNRBhbcu8U/edit?gid=0#gid=0'">Переглянути деталі</button>

        <p class="additional-text">За необхідності постинг можна призупинити
натиснувши кнопку нижче.</p>

        <button class="pause-button" onclick="pausePosting()">Пауза</button>
    </div>

    <script>
        function startFunction() {
            alert('Парсинг за обраними темами запущено');

```

```

    }

    function switchTab(tab) {
        document.querySelectorAll('.menu-button').forEach(button => {
            button.classList.remove('active');
        });
        document.querySelector(`.menu-
button[onclick="switchTab('${tab}')]`).classList.add('active');
    }

    function pausePosting() {
        alert('Процес публікації зупинено');
    }

    // Функція для ефекту паралакса
    document.addEventListener('mousemove', function (e) {
        const x = (e.clientX / window.innerWidth) - 0.5;
        const y = (e.clientY / window.innerHeight) - 0.5;
        document.body.style.backgroundPosition = `${x * 50}px ${y * 50}px`;
    });
</script>

</body>
</html>

```