МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ЛЬВІВСЬКИЙ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА
БІОТЕХНОЛОГІЙ ІМ. С. З. ҐЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

перший (бакалаврський) рівень вищої освіти

на тему:

«МУЛЬТИАГЕНТНА СИСТЕМА ДЛЯ ВІДСТЕЖЕННЯ ТА АНАЛІЗУ ДАНИХ З РОЗПОВСЮДЖЕННЯ НОВИННИХ ПОДІЙ»

Виконав: студент групи КН-41
спеціальності 122 «Комп’ютерні науки»
Кунець В.О.
(прізвище та ініціали)

Керівник: Желєзняк А.М.
(прізвище та ініціали)

ДУБЛЯНИ 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Рівень вищої освіти перший (бакалаврський)
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____
(підпис)
д.т.н., професор, Тригуба А. М.
(вч. звання, прізвище, ініціали)
"____" _____202__року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Кунця Владислава Олеговича
(прізвище, ім'я, по батькові)

1. Тема роботи *«Мультиагентна система для відстеження та аналізу даних з розповсюдження новинних подій»*

керівник роботи *к. е н., доцент., Желєзняк А.М.*
(наук.ступінь, вч. звання, прізвище, ініціали)

затверджені наказом Львівського НУП № 123/к-с від 25.02.2025р.

2. Строк подання студентом роботи 12.06.2025 р.

3. Вихідні дані: *Google News, WhatsApp Cloud API, сервер Oracle Cloud, система обробки тексту (NLP, Machine Learning), вебінтерфейс керування новинами (HTML/CSS/JavaScript).*

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)
Вступ
1. Теоретичні аспекти мультиагентних систем та обробки новин
2. Розробка архітектури та реалізація мультиагентної системи
3. Інтерфейс користувача та функціональність взаємодії з ботами
4. Безпека, тестування та результати
Висновки

5. Перелік графічного матеріалу
Діаграми архітектури, системи UI, інтерфейс UML, діаграми агентів, схема взаємодії з WhatsApp API Cloud, скріншоти роботи системи

6. Консультанти розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата		Відмітка про виконання
		завдання видав	завдання прийняв	
1, 2, 3	*к. е н., доцент., Желєзняк А.М.*			
4	*к.т.н., доцент Городецький І.М*			

7. Дата видачі завдання 10.11.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Терміни виконання етапів роботи	Примітка
1.	*Обґрунтування теми, постановка мети і завдань, складання плану роботи*	*01.11.2024 – 18.11.2024*	
2.	*Теоретичне дослідження мультиагентних систем, агрегаторів новин, алгоритмів обробки текстів*	*19.11.2024 – 22.12.2024*	
3.	*Проєктування архітектури системи, вибір технологій, розробка інтерфейсу*	*08.01.2025 – 05.02.2025*	
4.	*Реалізація системи: агенти збору, обробки та доставки новин через WhatsApp API Cloud*	*06.02.2025 – 15.03.2025*	
5.	*Тестування системи, аналіз результатів, написання висновків і розділу з охорони праці*	*16.03.2025 – 20.04.2025*	
6.	*Оформлення документації, підготовка до захисту, подання пояснювальної записки*	*21.04.2025 – 06.06.2025*	

Студент _____ *Кунець В. О.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Желєзняк А. М.*
(підпис) (прізвище та ініціали)

УДК 004.89

Кваліфікаційна робота: 49 сторінок текстової частини, 6 таблиць, 26 рисунків, 21 джерел літератури, 6 додатки.

«Мультиагентна система для відстеження та аналізу даних з розповсюдження новинних подій» Кунець В.О. – Кваліфікаційна робота. Кафедра інформаційних технологій. Дубляни, Львівський національний університет ветеринарної медицини та біотехнологій імені Степана Ґжицького.

У роботі було досліджено питання автоматизованого збору, обробки та розповсюдження новинних подій за допомогою мультиагентної системи. Реалізовано механізм збору новин з Google News API, їх переклад та класифікація, а також поширення новин через WhatsApp Cloud API. Розроблений вебінтерфейс для керування новинами. Представлено тестування, архітектура системи та аналіз ефективності.

Ключові слова: мультиагентна система, новини, аналіз тексту, класифікація

ЗМІСТ

ВСТУП

У сучасному інформаційному суспільстві, коли потік новинних подій тільки зростає в геометричній прогресії, питання ефективного збору, аналізу та поширення новин стає більш актуальним. Людство щоденно стикається з великою кількістю інформації, яка надходить з різних джерел, зокрема, новинних сайтів, соціальних мереж, месенджерів, агрегаторів та офіційних каналів. У таких умовах виникає потреба в інтелектуальних системах, які здатні автоматично обробляти інформаційні потоки, виокремлювати ключові події новин, визначати достовірність, а також ефективно поширювати результати аналізу до зацікавлених користувачів. Один із перспективних напрямів у цьому контексті є застосування мультиагентних систем — інтелектуальних обчислювальних моделей, окремі агенти яких взаємодіють між собою задля досягнення спільної мети розповсюдження.

Мультиагентні системи демонструють високу ефективність у розв’язанні задач зі складною логікою, вони потребують паралельного аналізу великої кількості змінних, а також більшої адаптації до змін зовнішнього середовища. Вони дозволяють розділити процес на окремі ролі: один агент може бути відповідальним за збір новин з відкритих джерел, інший — за лінгвістичний аналіз, третій — за класифікацію за категоріями, ще один — за визначення ознак фейковості, а останній — за передачу новини користувачу через зручний канал, наприклад, месенджер WhatsApp, Email-розсилка. Такий підхід не тільки підвищує продуктивність системи, але й дозволяє гнучко масштабувати її функціональність, додаючи нові агенти або змінюючи алгоритми поведінки існуючих.

Мета даної дипломної роботи - розробка мультиагентної системи, яка автоматизує процес відстеження новин, здійснює базовий аналіз змісту (включаючи переклад, категоризацію та перевірку на фейковість) і передає

результати у вигляді структурованих повідомлень користувачу через WhatsApp Cloud API, Email оповіщення. Для реалізації цього проєкту використовуються сучасні веб-технології на Python, зокрема Flask [2] як серверна основа, а також модулі для обробки природної мови (NLP), API Google News для збору новин і хмарна платформа Oracle Cloud як середовище розгортання системи.

Актуальність даної теми обумовлена не тільки стрімким зростанням обсягів новинної інформації, а й загрозами, які виникають внаслідок поширення дезінформації, маніпулятивних повідомлень або фейкових новин з різних джерел. Ефективна фільтрація та аналітичне опрацювання таких матеріалів – це серйозний виклик як для дослідників, так і для розробників. Саме тому створення автоматизованої системи, яка працює у режимі реального часу, є необхідним внеском у сферу інформаційної безпеки, цифрової журналістики та масових комунікацій.

У межах роботи були поставлені наступні завдання:

1. Дослідити архітектурні принципи мультиагентних систем.
2. Вивчити існуючі інструменти збору та обробки новин.
3. Реалізувати програмну систему, здатну збирати, аналізувати та надсилати новини користувачеві.
4. Протестувати функціональність проєкту на реальних прикладах.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ МУЛЬТИАГЕНТНИХ СИСТЕМ ТА ОБРОБКИ НОВИН

1.1 Теоретичні основи та поняття мультиагентних новин

Інформаційні технології сучасного етапу розвитку вимагають систем, як здатні до автономного прийняття рішень, адаптації до динамічних змін середовища та ефективної взаємодії з іншими учасниками цифрових процесів. Одним із найбільш перспективних напрямів у цій сфері є мультиагентні системи (МАС) — інтелектуальні програмні комплекси, що складаються з декількох агентів, кожен з яких виконує визначену роль, діючи автономно або у координації з іншими.

Агент — це програмна сутність, яка здатна сприймати навколишнє середовище, приймати рішення та здійснювати дії задля досягнення визначених цілей. Якщо система складається з багатьох агентів, які взаємодіють між собою, узгоджують дії, розподіляють завдання та спільно вирішують задачі, її називають мультиагентною. Основні характеристики мультиагентних систем:

Автономність — кожен агент діє незалежно та самостійно; *Реактивність* — агенти здатні реагувати на зміни середовища; *Протокол взаємодії* — агенти спілкуються за допомогою чітко визначених протоколів; *Колективність* — система досягає результатів через узгоджену роботу кількох агентів; *Адаптивність* — агенти можуть змінювати поведінку відповідно до змін в умовах роботи.

МАС ефективно використовуються в різних галузях: робототехніці, інтелектуальних транспортних системах, фінансовому моделюванні, інформаційному пошуку, обробці великих даних, системах кібербезпеки та, зокрема, у сфері збору та обробки новинної інформації.

В контексті даної роботи мультиагентна система дозволяє розділити складний процес відстеження новин на окремі задачі:

— агент-збірник отримує новини з джерел (наприклад, Google News),

— агент-аналітик обробляє текст, визначає тематику, здійснює переклад,

— агент-доставник — передає користувачу новину через канал зв’язку (наприклад, WhatsApp). Це дозволяє досягти гнучкої модульної структури з можливістю масштабування.

Рисунок 1.1. – Загальна структура мультиагентної системи новин

Ще одним важливим аспектом мультиагентних систем є спосіб взаємодії між агентами. У класичних реалізаціях агенти можуть комунікувати за допомогою системи повідомлень або спільного середовища збереження даних, яке використовується як «чорна дошка» (blackboard). У дипломному проєкті цей підхід реалізовано через файл news_storage.json, який виступає центральним сховищем для зібраних, оброблених та готових до відправки новин. Кожен агент має доступ до цього файлу: збирач новин дописує нові записи, аналітик редагує їх, додаючи переклад та категорію, а агент-доставник зчитує дані для надсилання користувачам.

Завдяки цьому забезпечується асинхронна взаємодія: агенти не залежать від синхронного виклику один одного і можуть працювати паралельно. Наприклад, під час того, як один агент обробляє новини, інший може вже надсилати попередні результати користувачу. Такий підхід значно підвищує ефективність обробки

великої кількості інформації, що особливо важливо при роботі з новинами у реальному часі. У науковій літературі мультиагентні системи класифікуються за різними критеріями: *За типом агентів:* реактивні, когнітивні, гібридні; *За способом комунікації:* централізовані (через головного агента) та децентралізовані (взаємна комунікація); *За середовищем застосування:* фізичні (роботи, сенсори) та віртуальні (системи обробки даних).

У даній роботі реалізовано віртуальну мультиагентну систему з децентралізованою логікою взаємодії через файл-зв'язок. Хоча така система є спрощеною у порівнянні з повноцінними розподіленими агентними платформами (наприклад, JADE чи SPADE), вона дозволяє ефективно досягати поставлених завдань при мінімальному обсязі ресурсів.

Особливу увагу заслуговує поняття інтелекту агента. У більшості практичних реалізацій, таких як у цьому проєкті, інтелект означає здатність агента до обробки даних, прийняття рішень на основі заданих правил або моделей, а також до взаємодії з іншими агентами. Наприклад, NLP-агент у системі може мати вбудовану модель машинного навчання для класифікації новин за тематикою (економіка, політика, культура) або ж API-інтеграцію з сервісами перекладу (наприклад, DeepL або Google Translate), що забезпечує додаткову гнучкість.

У загальному випадку ефективність мультиагентної системи оцінюється за критеріями:

- Швидкість обробки інформації;
- Якість прийнятих рішень (наприклад, точність класифікації);
- Масштабованість (можливість додавання нових агентів без перебудови всієї системи);
- Надійність у роботі (відсутність збоїв при паралельній роботі агентів).

1.2 Огляд методів збору, класифікації та фільтрації новин

В умовах цифрового світу новини стали однією з найдинамічніших форм інформації. Сотні новинних порталів щогодини публікують матеріали, які розповсюджуються через різні канали — вебсайти, мобільні додатки, соціальні мережі, месенджери. Саме тому важливим етапом у побудові системи для роботи з новинами є вибір ефективного методу збору актуальної інформації з відкритих джерел.

Одним з найпоширеніших підходів є використання агрегаторів новин. Серед них варто виділити такі сервіси, як Google News, Bing News, Yahoo News, які дозволяють отримувати стислий потік новин за тематиками, мовами, ключовими словами. У рамках цієї роботи основним джерелом новин є Google News, яке надає зручні механізми збору публікацій через стандартні RSS-канали або неофіційні API.

З технічного погляду новини можна збирати наступними способами: Scraping (парсинг HTML) — підходить для сайтів без API, проте потребує підтримки; RSS (Really Simple Syndication) — простий і надійний механізм, проте обмежений у даних; API (програмні інтерфейси доступу) — найбільш ефективний метод, що дозволяє отримувати структуровану інформацію з можливістю фільтрації та сортування.

Після збору виникає потреба у класифікації новин. У межах проєкту використовується базова модель тематичної класифікації — політика, економіка, культура тощо. Таку класифікацію можна реалізувати на основі ключових слів, шаблонів або з використанням NLP-бібліотек, таких як spaCy чи scikit-learn. У випадку подальшого розширення систему можна під'єднати до моделей машинного навчання для класифікації за змістом чи емоційним тоном.

Особливу роль відіграє також фільтрація новин, яка дозволяє виключити повторювані або нерелевантні матеріали. Один із підходів — перевірка на

унікальність за заголовками або URL, інший — оцінка змісту за схожістю ключових фраз.

Для кращої візуалізації порівняймо основні методи збору новин за кількома критеріями (табл.1.1.).

Таблиця 1.1. - Порівняння методів збору новин

Метод	Переваги	Недоліки	Приклад використання
HTML-парсинг	Гнучкість, незалежність від API	Крихкість, залежність від структури сторінки	BeautifulSoup, lxml
RSS	Простота, стандартність, легкість впровадження	Обмежена інформація, часто лише заголовки	feedparser, built in tools
API	Структуровані дані, фільтрація, масштабованість	Обмеження в запитах, потребує ключів авторизації	Google News API, GNews API

У межах цієї роботи був обраний гібридний підхід — комбінація використання RSS (для заголовків) та API (для розширеної інформації, зокрема опису, посилання на джерело та зображення). Це дозволяє досягти балансу між повнотою новин і швидкістю їх отримання.

Щодо класифікації новин, то навіть базова модель, побудована на наборі тематичних ключових слів, дозволяє значно покращити користувацький досвід, оскільки кінцевий користувач отримує лише релевантну для нього інформацію.

У подальшому планується розширити систему за рахунок додавання модуля визначення фейкових новин шляхом аналізу структури тексту, джерел, стилістичних ознак та застосування моделей на кшталт BERT.

Таким чином, ефективне поєднання методів збору, класифікації та фільтрації дозволяє будувати гнучкі й адаптивні системи доставки новин, що відповідають вимогам сучасного користувача.

1.3 Аналіз аналогічних рішень та платформ

Перш ніж розробляти власну систему для відстеження, аналізу та поширення новин, доцільно розглянути наявні рішення, які вже функціонують на ринку, а також оцінити їх архітектурні підходи, технічні можливості та обмеження. Це дозволяє не лише перейняти корисні ідеї, а й виявити слабкі сторони, які можна уникнути при створенні власного продукту.

Сьогодні існує велика кількість агрегаторів новин, платформ для моніторингу інформаційного простору та систем розсилки новин. Більшість з них є або вузькоспеціалізованими, або масовими комерційними продуктами, які орієнтовані на широке коло користувачів без можливості глибокої кастомізації. Наприклад, сервіси Google Alerts, Flipboard чи News360 дозволяють підписуватись на новини за темами, але не дають можливості впливати на спосіб збору, аналізу чи доставки новин.

З іншого боку, існують професійні системи, які використовуються у корпоративному середовищі — для моніторингу репутації, інформаційного поля або конкурентів. Такі рішення, як Meltwater, Brandwatch чи Mention, працюють за принципом складної аналітичної платформи, де новини обробляються за численними метаданими, але їхня інтеграція є платною та закритою і обмеженою з точки зору адаптації.

Для цілей цієї кваліфікаційної роботи важливо оцінити також архітектуру подібних систем. Більшість з них мають монолітну структуру або використовують централізовану серверну обробку, без мультиагентного поділу. Це означає, що всі дії — від збору до фільтрації — виконує один загальний процес або модуль. Такий підхід значно обмежує гнучкість, масштабованість та адаптацію до нових завдань.

У порівнянні з цим, мультиагентна система, яку пропонується реалізувати в межах даної роботи, дозволяє розділити функціональність на окремі компоненти (агенти), що значно полегшує її розширення та обслуговування.

Рисунок 1.2 – Порівняння монолітної та мультиагентної архітектури системи

Зліва – типовий підхід «все-в-одному»; справа – запропонована модульна мультиагентна структура, де кожен компонент виконує окрему функцію: збір, аналіз, класифікацію, доставку. Такий підхід забезпечує гнучкість масштабування і спрощує супровід.

Також для повноти аналізу доцільно порівняти наявні рішення за ключовими критеріями. У таблиці 1.2 наведено функціональні можливості платформ, що були розглянуті.

Таблиця 1.2 - Порівняння функціональності аналогічних систем

Система / Платформа	Збір новин	Класифікація	Переклад	Доставка через месенджер	Архітектура
Google Alerts	+	-	-	-	Централізована
FlipBoard	+	частково	-	-	Централізована
Brandwatch	+	+	+	-	Комерційна
Власна мультиагентна	+	+	+	+	Мультиагентна

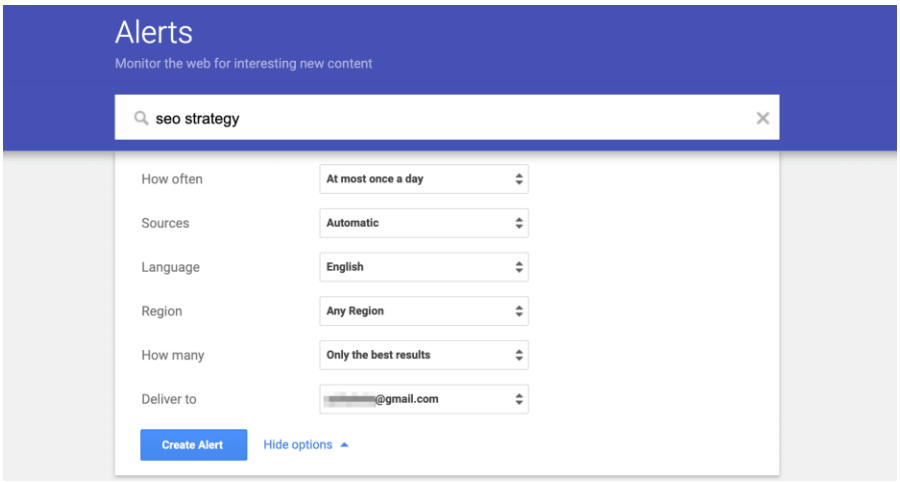

Рисунок 1.3 – Google Alerts. Скріншот сторінки з налаштуванням сповіщень.

Сервіс Google Alerts забезпечує базову фільтрацію новин за ключовими словами, але не дозволяє їх переклад або класифікацію. Надсилання здійснюється лише через email, і не передбачає інтеграції з месенджерами.

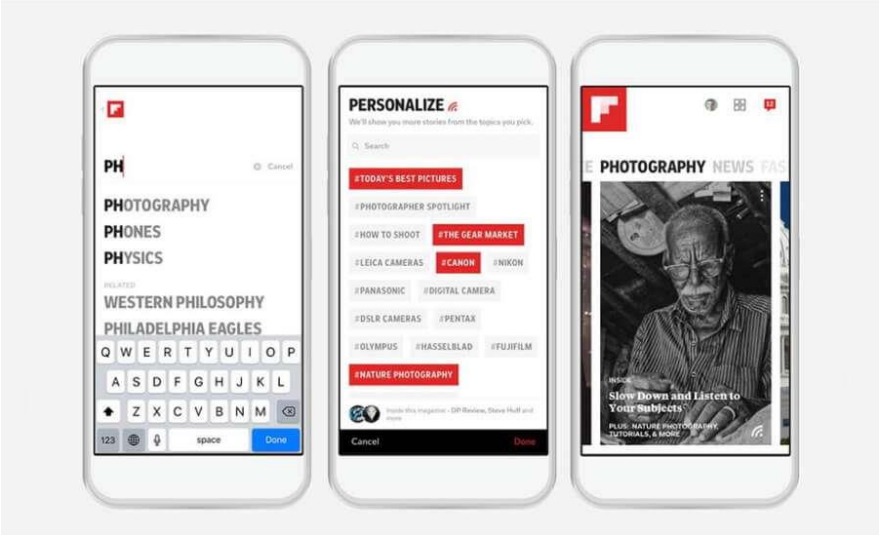

Рисунок 1.4 – Flipboard. Скріншот з головної стрічки новин мобільного додатку [1]

Інтерфейс Flipboard демонструє стрічку новин з фільтрацією за категоріями, однак система не дозволяє розширення функціоналу, не має відкритого API і не підтримує автоматизовану доставку новин через зовнішні канали зв’язку.

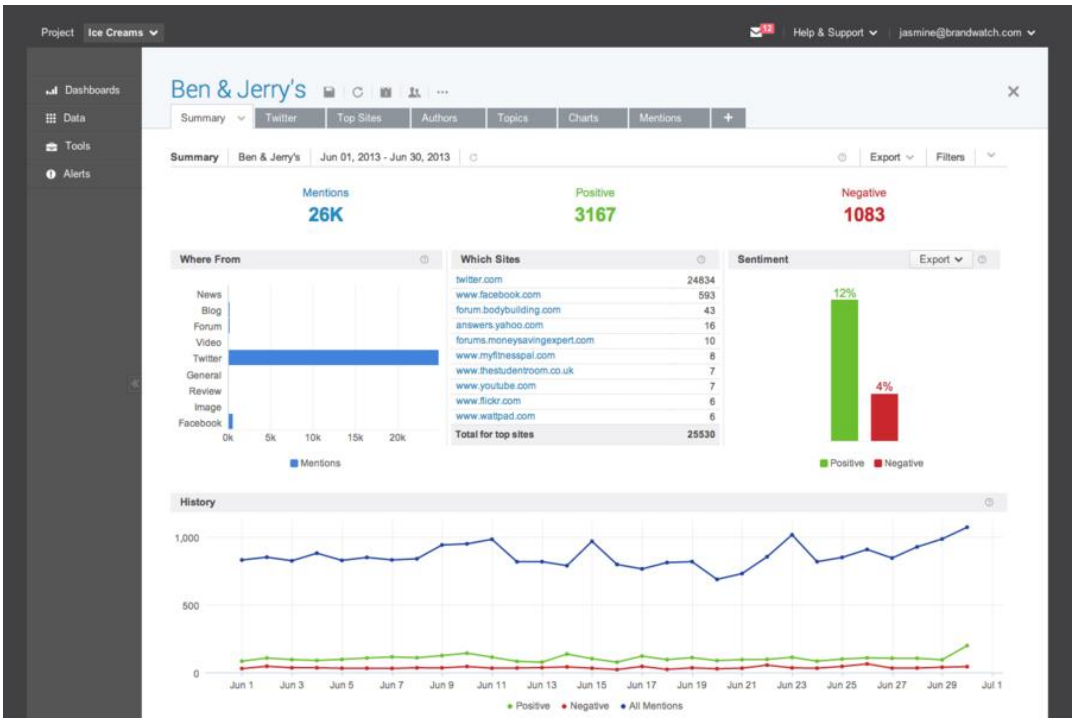

Рисунок 1.5 – Brandwatch. Панель інтерфейсу з дашбордом (аналітика).

Проведений аналіз демонструє, що хоча існують системи з подібною функціональністю, жодна з них не об’єднує в собі автоматичний збір, обробку та надсилання новин через месенджер, реалізовані у формі мультиагентної платформи з відкритою архітектурою. Саме така модель лежить в основі цієї дипломної роботи.

РОЗДІЛ 2
ПРОЄКТУВАННЯ МУЛЬТИАГЕНТНОЇ СИСТЕМИ НОВИН

2.1 Постановка задачі та моделювання системи новин

У сучасному інформаційному просторі постає необхідність в автоматизованих системах, які здатні в режимі реального часу відстежувати актуальні події, аналізувати інформацію та доставляти її користувачу через зручні цифрові канали. Така система повинна враховувати обмеженість часу користувача, динамічність новинного потоку, потребу в попередньому фільтруванні й релевантності інформації.

Метою цієї кваліфікаційної роботи є створення мультиагентної системи, що реалізує повний цикл обробки новин: від збору до поширення. Система повинна мати гнучку архітектуру, модульність та можливість масштабування. У її основі закладена концепція взаємодії незалежних агентів, кожен з яких відповідає за окрему функцію — таким чином досягається розподіл навантаження та підвищується адаптивність системи.

Основні функціональні вимоги до системи:

- ✓ Автоматичне отримання новин з відкритих джерел (Google News);
- ✓ Перевірка новин на дублікатність та релевантність;
- ✓ Автоматичний переклад новин на українську мову;
- ✓ Класифікація за категоріями (політика, економіка, культура тощо);
- ✓ Збереження новин у локальне сховище (news_storage.json);
- ✓ Відображення новин у вебінтерфейсі з можливістю редагування та видалення;
- ✓ Надсилання новин користувачам через WhatsApp Cloud API.

Визначення структури системи проводилося на основі функціональної декомпозиції. Це дозволяє розбити загальну задачу на окремі компоненти, кожен з яких має власну зону відповідальності.

Таблиця 2.1 - Основні компоненти системи та їх призначення

Компоненти	Призначення
Агент-збірник	Отримання новин з Google News, у структурованому форматі
Агент-обробник	Переклад новин, класифікація, фільтрація дублікатів
Агент-доставник	Надсилання оброблених новин користувачу через WhatsApp API Cloud, Email
Інтерфейс керування	Відображення новин, кнопки редагування/видалення, контроль категорій
Сховище новин	Локальний файл news_storage.json для збереження повної інформації про новини

Для подальшого моделювання було прийнято рішення реалізувати інтерфейс взаємодії між агентами на основі обміну через загальне сховище. Такий підхід дає змогу уникнути прямої залежності між модулями та дозволяє виконувати кожен з етапів асинхронно. Наприклад, навіть у випадку збою на етапі надсилання повідомлення, зібрані та оброблені новини залишаться збереженими та доступними для повторної спроби доставки.

З урахуванням поставлених завдань, модель системи може бути зображена у вигляді агентно-орієнтованої структури, де компоненти функціонують незалежно, але взаємодіють через визначені інтерфейси. У подальших підрозділах буде детально описано архітектуру, реалізацію компонентів і логіку обміну даними між агентами.

2.2 Архітектура системи та опис компонентів

Архітектура інформаційної системи визначає логіку побудови, взаємозв’язки між її складовими частинами, спосіб обміну даними та рівень взаємозалежності між компонентами. Для ефективного функціонування мультиагентної системи, що виконує збір, обробку та поширення новин, було обрано модульну архітектуру, в основі якої — взаємодія незалежних програмних агентів.

Загальна архітектура побудована за принципом трирівневої логіки: Рівень збору даних — агент-збірник новин, який працює з Google News через API або RSS. Рівень обробки — агент-аналітик, який здійснює переклад, класифікацію, формування опису. Рівень поширення — агент-доставник, який надсилає новини користувачу через месенджер. Усі агенти взаємодіють через спільне JSON-сховище, яке виступає єдиним джерелом даних.

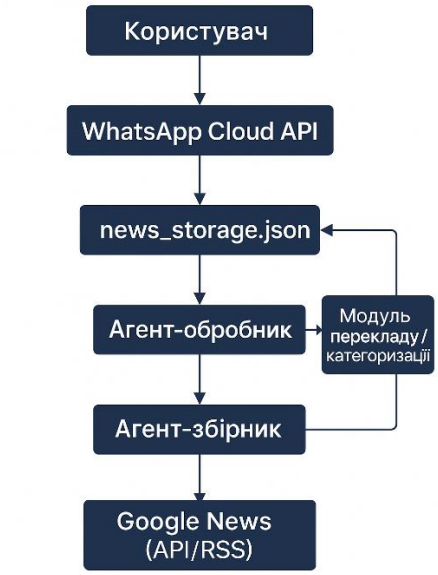

Рисунок 2.1 – Загальна архітектура мультиагентної системи

На схемі зображено взаємозв’язки між агентами та зовнішніми сервісами. Дані циркулюють через сховище news_storage.json, яке забезпечує незалежність

компонентів.

Кожен компонент системи виконує конкретне завдання, а його логіка реалізується окремим модулем у програмному коді. Це дозволяє зручно оновлювати або замінювати окремі частини без потреби втручання в усю систему.

Опис основних компонентів: Агент-збірник новин (fetch_news.py) Отримує новини з Google News, витягує заголовки, опис, пряме посилання на джерело, зображення (за наявності) і записує до news_storage.json. Агент-аналітик Аналізує зібрані новини: здійснює переклад з англійської на українську (через DeepL [3] або Google Translate API), визначає категорію (економіка, політика, культура), формує короткий опис. Агент-доставник (send_message.py) Забезпечує зв’язок з WhatsApp Cloud API [4]. Формує повідомлення на основі шаблону, додає посилання, зображення та опис, і надсилає користувачу. Вебінтерфейс (home.html, manage.html) Відображає новини для адміністратора: перегляд, редагування, видалення, надсилання вручну. Також реалізовано автооновлення новин кожні 2 хвилини.

Рисунок 2.2 – Взаємодія компонентів на прикладі одного циклу

Цей рисунок демонструє типовий сценарій роботи одного повного циклу новини – від отримання до доставки.

Обрана архітектура дозволяє уникнути надмірної складності та забезпечує достатню гнучкість для розширення. Наприклад, у майбутньому до системи

можна додати нового агента — наприклад, для визначення фейковості новини, або підключити додатковий канал доставки, наприклад, Viber або Signal.

У наступному підрозділі буде описано моделі взаємодії агентів більш детально, з акцентом на формат зберігання новин, структуру сховища та оновлення даних.

2.3 Модель взаємодії агентів і схеми зберігання даних

Ефективність роботи мультиагентної системи значною мірою залежить від правильно спроєктованої моделі взаємодії між її складовими компонентами. Відсутність надмірної залежності між модулями, чітко визначені точки обміну даними та зрозуміла структура зберігання інформації — це основа для стабільної, масштабованої і зрозумілої системи.

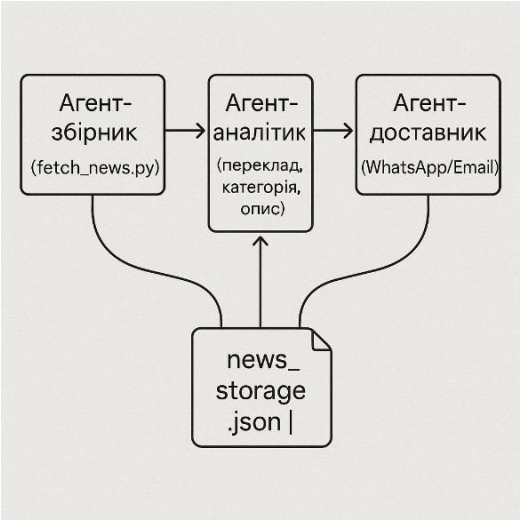

Рисунок 2.3 – Модель взаємодії агентів через сховище news_storage.json

У розробленій системі всі агенти взаємодіють не напряму, а через централізоване JSON-сховище новин, реалізоване у вигляді файлу news_storage.json. Цей файл є своєрідною "базою даних", де кожна новина представлена у вигляді окремого об’єкта з відповідними ключами: заголовок,

опис, мова, категорія, URL, зображення, статус обробки, канал доставки тощо.

Дані циркулюють виключно через загальне сховище. Це дозволяє уникнути прямої залежності між модулями.

Формат структури даних:

Кожна новина зберігається як об'єкт у форматі JSON, наприклад:


```
 {
  "title": "Єврокомісія схвалила новий пакет допомоги",
  "description": "Новий пакет включає фінансування на суму 18 млрд євро.",
  "category": "політика",
  "language": "uk",
  "source_url": "https://example.com/news1",
  "image_url": "/static/media/news1.jpg",
  "translated": true,
  "sent_whatsapp": true,
  "sent_email": false
 }
```


Цей формат дозволяє:

1. Відслідковувати стан кожної новини;
2. Уникати повторного надсилання;
3. Фільтрувати за категоріями або статусом обробки;
4. Легко додавати нові параметри (наприклад, fake_score, date_added, тощо).

Типи взаємодії агентів.

Модель передбачає асинхронну взаємодію — кожен агент працює незалежно:

1) Агент-збірник періодично опитує Google News, додає новини до news_storage.json.

2) Агент-аналітик запускається окремо, обробляє новини, які ще не

перекладені або не класифіковані.

3) Агент-доставник перевіряє статус новин і, залежно від обраного каналу (WhatsApp або Email), надсилає повідомлення.

Переваги такої моделі:

1) Модульність — кожен агент може бути оновлений незалежно;

2) Простота тестування — можливість протестувати кожен етап окремо;

3) Гнучкість — можна додавати нові поля або агенти без повного переписування структури;

4) Надійність — при збоях на одному етапі інші агенти продовжують роботу.

Додатково, реалізована підтримка декількох каналів доставки новин (WhatsApp, Email), що значно розширює аудиторію та додає гнучкості при інтеграції в реальні проєкти. У news_storage.json кожна новина має статуси sent_whatsapp, sent_email, які дають змогу точно контролювати канали поширення.

У наступному підрозділі буде описано структуру інтерфейсу користувача, який дозволяє керувати новинами: переглядати, редагувати, видаляти або надсилати вручну.

2.4 Проектування інтерфейсу користувача

Користувацький інтерфейс відіграє ключову роль у забезпеченні ефективної взаємодії людини з інформаційною системою. У межах реалізованої мультиагентної системи було створено два основні інтерфейси:

– інформаційний, що орієнтований на кінцевого користувача;

– керуючий, призначений для адміністратора або оператора системи. Обидва інтерфейси розроблені з урахуванням сучасних вимог до доступності, зручності використання та адаптивності.

Основні принципи, покладені в основу проєктування:

1. простота і мінімалізм — інтерфейс не перевантажений зайвими

елементами;

2. логічна структура — новини згруповані за категоріями, кнопки мають чітке призначення;
3. відповідність сценаріям використання — для кожного типу користувача передбачено лише необхідний функціонал;
4. адаптивність — інтерфейс коректно відображається на різних пристроях і в різних розмірах екранів.

Обидва інтерфейси спроєктовані на основі компонентного підходу, з урахуванням можливості подальшого розширення. В основі розробки — використання HTML, CSS з фреймворком Bootstrap, та JavaScript з інтеграцією AJAX-запитів для динамічного оновлення контенту. Серверна частина реалізована на базі фреймворку Flask, що дозволяє здійснювати обмін даними з клієнтською частиною через формат JSON.

Інформаційний інтерфейс дозволяє користувачеві ознайомлюватися з останніми новинами, переглядати короткий опис, відкривати повну новину у джерелі, сортувати інформацію за категоріями. Особливу увагу приділено візуальному оформленню карток новин, які включають заголовок, дату, зображення та опис. Керуючий інтерфейс, у свою чергу, передбачає наявність додаткового функціоналу, доступного лише адміністратору. Зокрема:

1. можливість редагування вмісту новини;
2. запуск перекладу з англійської мови;
3. надсилання новини у вибраний месенджер (WhatsApp або Email);
4. додавання/оновлення зображення;
5. збереження змін у локальному сховищі.

Проєктування обох інтерфейсів базувалося на розмежуванні ролей користувачів, що відповідає підходу до реалізації мультиагентної архітектури — кожен агент виконує окрему функцію, і користувач взаємодіє лише з результатом цієї роботи. З точки зору архітектури, клієнтська частина реалізована як односторінковий інтерфейс (SPA-фрагменти), який взаємодіє з сервером через

API, не потребуючи повного перезавантаження сторінки. Це дозволяє досягти високої швидкодії та зручного користування.

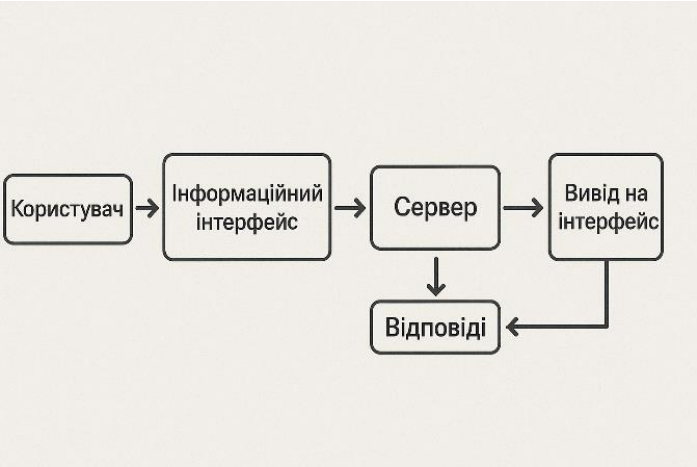

Рисунок 2.4 – Загальна логіка взаємодії користувача з системою

Рисунок 2.5 – Схема ролей і доступу до функціоналу інтерфейсів

Таким чином, інтерфейс користувача було спроєктовано з дотриманням принципів зручності, адаптивності та розширюваності. З урахуванням можливості майбутнього розширення функціоналу, закладено основу для масштабування системи без необхідності повного перепроектування інтерфейсу.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Отримання, збереження, переклад та категоризація новин

На першому етапі реалізації мультиагентної системи було створено окремий агент для автоматичного збору новин із відкритих джерел. Для цього використано модуль fetch_news.py, який здійснює запити до Google News через RSS-канали за заданими ключовими словами та категоріями. Серед основних полів, що збирає агент для кожної новини, можна виокремити: заголовок; короткий опис; пряме посилання на джерело; категорія (політика, культура, бізнес, міжнародні); зображення (якщо необхідне); дата публікації.

Зібрані новини фільтруються від дублікатів і зберігаються у структурованому вигляді у файл news_storage.json, що виступає як локальне сховище для подальшої обробки іншими агентами.

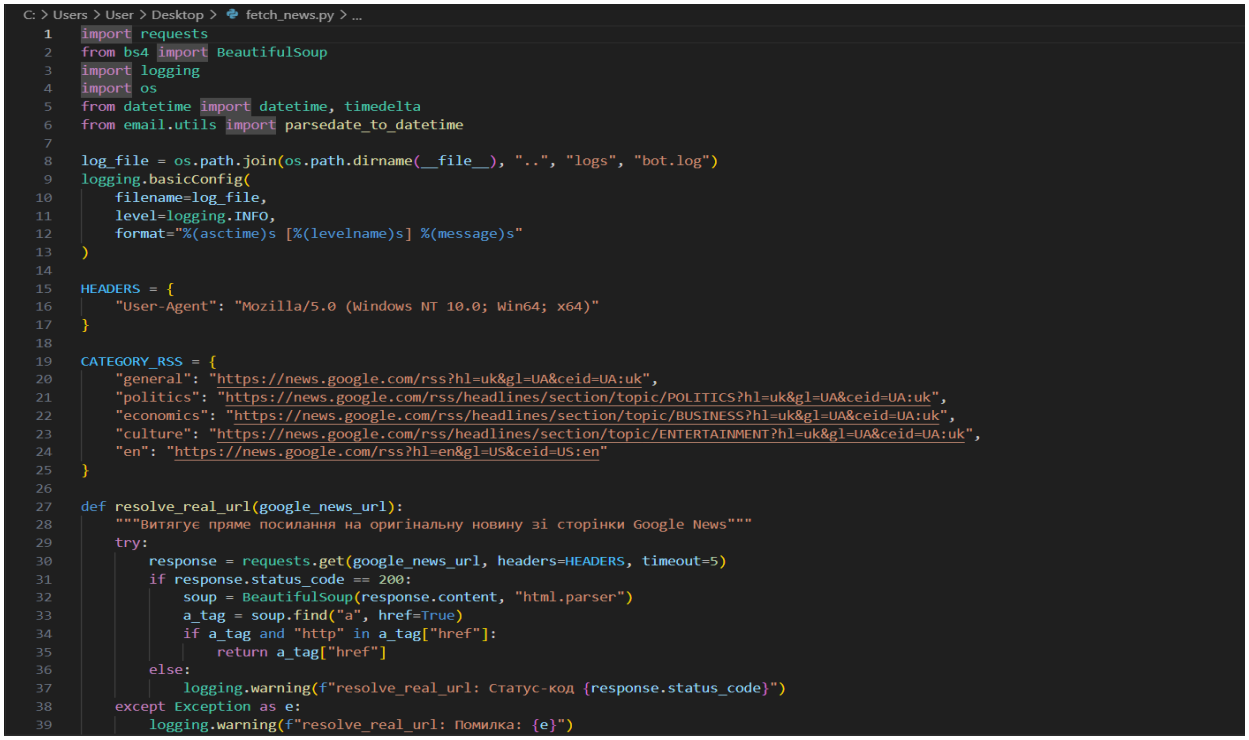

Рисунок 3.1 – Структура агента збору новин (fetch_news.py)

Після збереження новини передаються агенту перекладу та категоризації. Для перекладу з англійської мови використовується зовнішній API-сервіс (наприклад, DeepL API), який забезпечує високу якість автоматичного перекладу.

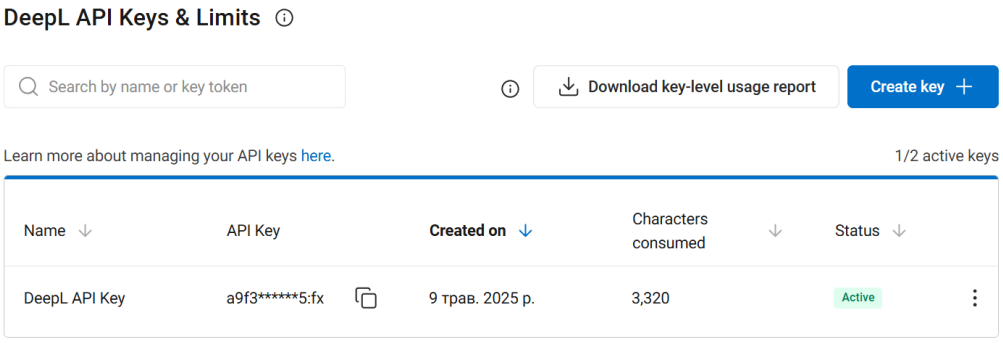

Рисунок 3.2 – Скріншот з головної сторінки Deepl API

Процес перекладу реалізовано в окремій функції, що отримує англомовний заголовок та опис, і повертає переклад українською мовою. Переклад зберігається у відповідні поля об’єкта новини у форматі JSON.

Категоризація новин виконується на основі ключових слів у заголовку та описі. Було розроблено просту евристичну модель, що визначає категорію відповідно до наявності термінів, притаманних політичній, культурній, економічній або міжнародній тематиці.

Рисунок 3.3 – Загальна схема обробки новини: від збору до категоризації

Категоризація новин здійснюється за ключовими словами, попередньо

сформованими вручну для кожної категорії. Наприклад, слова "уряд", "парламент", "вибори" зараховують новину до політичної категорії, тоді як "театр", "музика", "кіно" — до культурної.

Таблиця 3.1 – Приклад класифікації новин за ключовими словами

Категорія	Ключові слова	Приклад новини
Політика	уряд, вибори, парламент, президент, закон	«Президент анонсував нові зміни до законодавства»
Бізнес	ринок, інвестиції, економіка, біржа, криптовалюта	«Ринок криптовалют стрімко зростає»
Культура	музика, театр, кіно, фестиваль, виставка	«У Львові стартував міжнародний кінофестиваль»
Міжнародні	ООН, НАТО, війна, дипломатія, санкції, конфлікт	«ООН засудила нові дії уряду Північної Кореї»

Щоб уникнути помилок при збереженні, було реалізовано перевірку на наявність ключових полів у кожній новині: заголовка, опису, посилання, дати. Якщо хоча б один з елементів відсутній — новина не потрапляє в фінальний JSON.

Файл news_storage.json оновлюється щоразу, коли надходить нова партія новин. При цьому реалізовано механізм запобігання дублюванню: кожна новина перевіряється за унікальним хешем, сформованим на основі заголовка та посилання.

Збереження новин у форматі JSON забезпечує простоту подальшого використання — наприклад, для аналізу, надсилання або побудови статистики. Типова структура одного запису виглядає так:

{

"title": "Президент анонсував нові реформи",
"description": "Згідно з новою політикою уряду, буде змінено систему оподаткування...",
"link": "https://example.com/news/president-reforms",
"category": "Політика",
"image_url": "https://example.com/media/image1.jpg",
"lang": "uk",
"translated": true,
"timestamp": "2025-05-27T10:00:00"
}

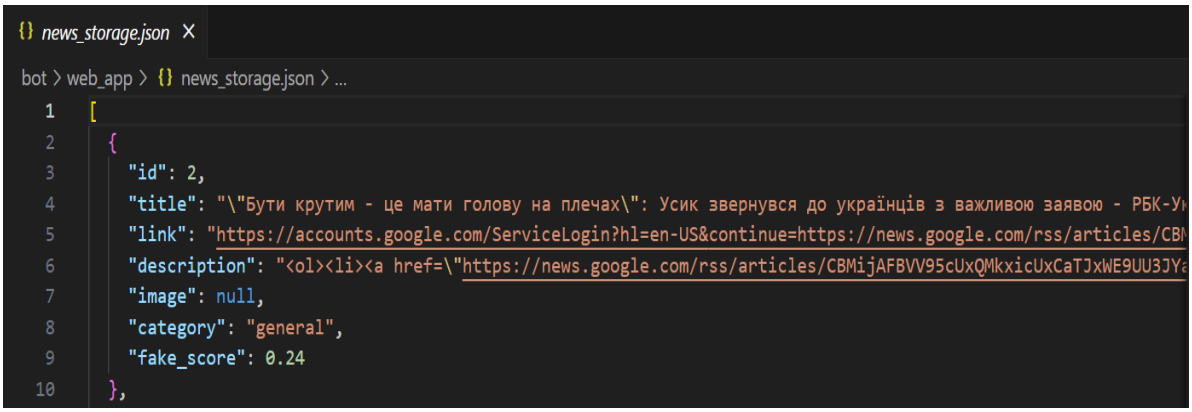

Рисунок 3.4 – Фрагмент збереженої новини у файлі news_storage.json

Крім основного сховища, було передбачено резервне збереження у файл news_general.json, що містить повну історію оброблених новин, зокрема тих, які не потрапили до стрічки через дублювання або порожні поля.

Таким чином, на першому етапі реалізації створено функціональний ланцюг: автоматичний збір → переклад → категоризація → збереження, який є базою для подальшої взаємодії із системою.

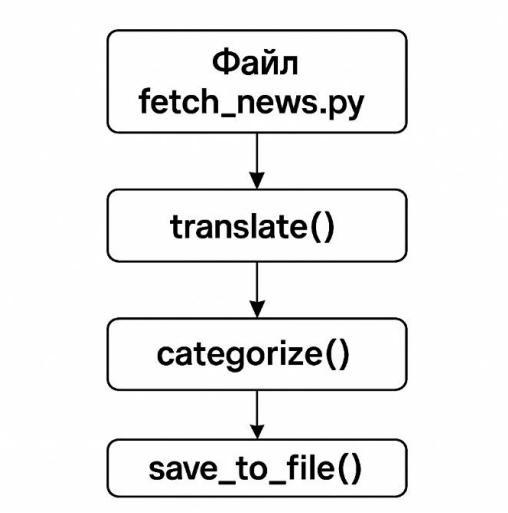

Рисунок 3.5 – Архітектура взаємодії агентів збору, перекладу та збереження новин

На рисунку 3.5 представлено архітектуру взаємодії агентів збору, перекладу та збереження новин мультиагентної системи. Агенти працюють у напівавтономному режимі, що дозволяє масштабувати процес при збільшенні кількості джерел чи мовних пар.

3.2 Інтеграція з WhatsApp Cloud API та надсилання новин

Після реалізації механізму збору, перекладу та категоризації новин було створено окремий агент для надсилання новин кінцевим користувачам через месенджери.

У межах цієї підсистеми реалізовано два канали доставки:

1. WhatsApp Cloud API (Meta);
2. Email-повідомлення через SMTP.

Ці методи було обрано як найзручніші для мобільних користувачів, оскільки не вимагають додаткових застосунків або облікових записів, окрім звичних платформ для комунікації.

Для реалізації надсилання повідомлень у WhatsApp використано офіційний сервіс Meta WhatsApp Cloud API. Цей сервіс надає розробникам можливість

надсилати шаблонні повідомлення через REST API, використовуючи авторизаційний токен та ідентифікатор телефонного номера (phone number ID).

Після реєстрації в Meta Business Suite було створено окремий додаток, згенеровано персональний API-токен і отримано всі необхідні параметри для налаштування. Ці дані було збережено у .env файлі проєкту:

```
WHATSAPP_TOKEN=EAAG...ZCZDZD
PHONE_NUMBER_ID=112233445566778
```

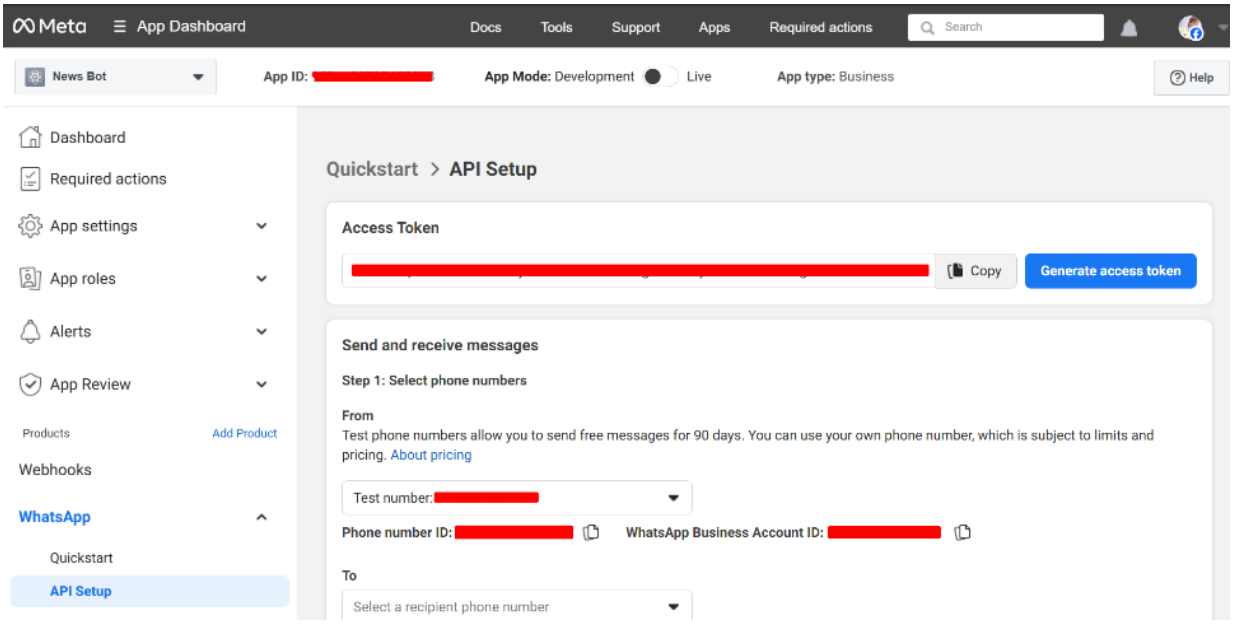

Рисунок 3.6 – Сторінка налаштування WhatsApp Cloud API в Meta Developer Portal

Функціонал надсилання реалізовано у модулі send_message.py, де визначено функцію send_via_whatsapp(text, phone_number). Ця функція формує POST-запит до API, передаючи номер одержувача, шаблон і текст новини, що включає заголовок, опис, посилання та категорію.

Шаблон повідомлення було створено у Meta Business Suite та підтверджено модерацією. Він підтримує змінні полів ({{1}}, {{2}}, тощо), які динамічно заповнюються при формуванні повідомлення.

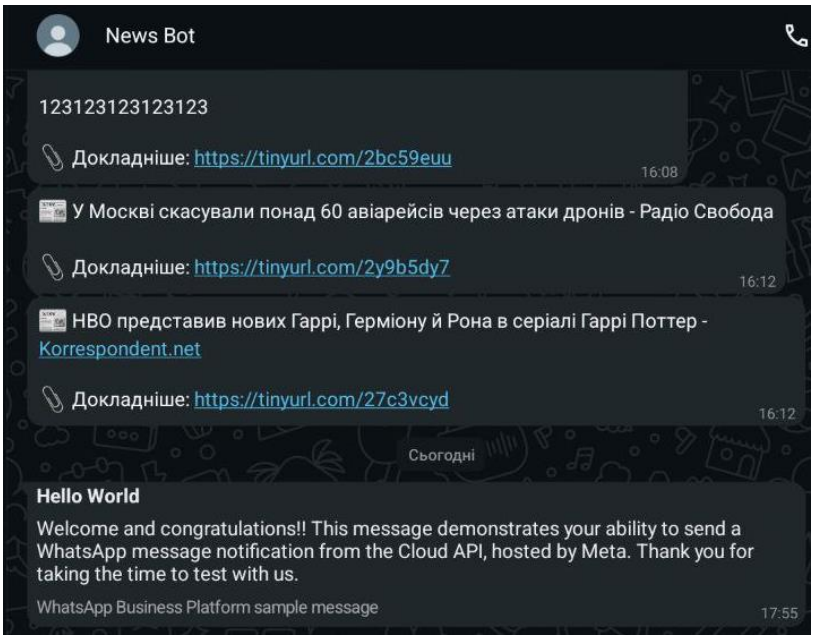

Рисунок 3.7 – Приклад створення шаблону повідомлення у Meta Business Suite

Перед надсиланням користувач у manage.html обирає новину та тисне кнопку «Надіслати через WhatsApp», після чого AJAX-запит викликає API-функцію, яка надсилає новину у форматі:

📌 Заголовок: Президент анонсував нові реформи

📄 Опис: Згідно з новою політикою уряду...

📎 Джерело: https://example.com/news.

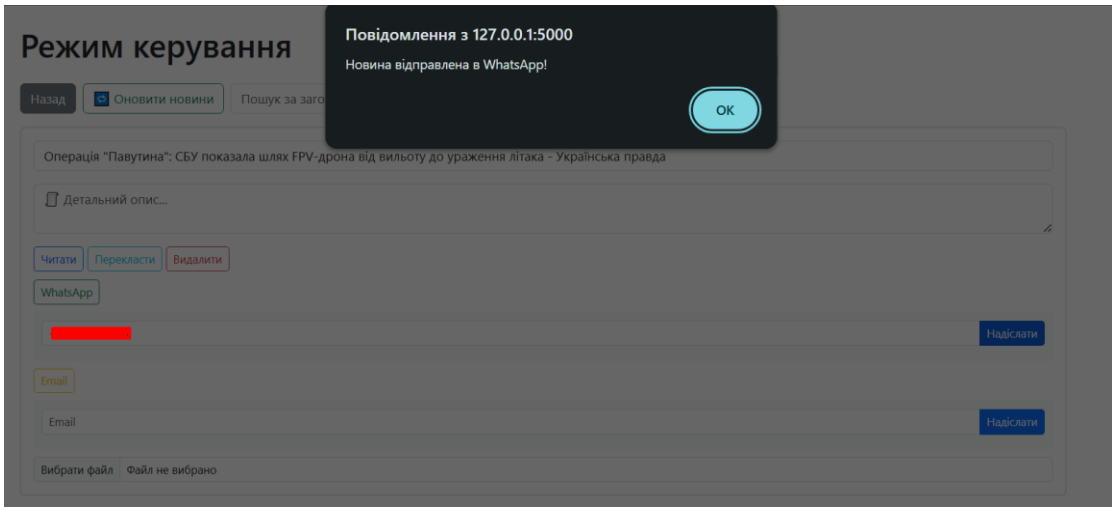

Рисунок 3.8 – Надсилання новини через WhatsApp із інтерфейсу manage.html

Було реалізовано обробку відповідей від API. Якщо повідомлення не доставлено (наприклад, через неправильний номер або токен), створюється запис у лог-файл logs/bot.log.

Другий спосіб доставки реалізовано через стандартний протокол SMTP із використанням облікового запису Gmail, який було налаштовано на приймання з'єднань із меншим рівнем безпеки (або з використанням App Passwords).

Конфігурація збережена у .env:

```
EMAIL_SENDER=myprojectnews@gmail.com
EMAIL_PASSWORD=app-password-here
```

Для надсилання реалізовано функцію send_email(subject, body, to_email), яка підключається до SMTP-сервера Google, створює текст повідомлення у форматі plain text або HTML, і відправляє його отримувачу.

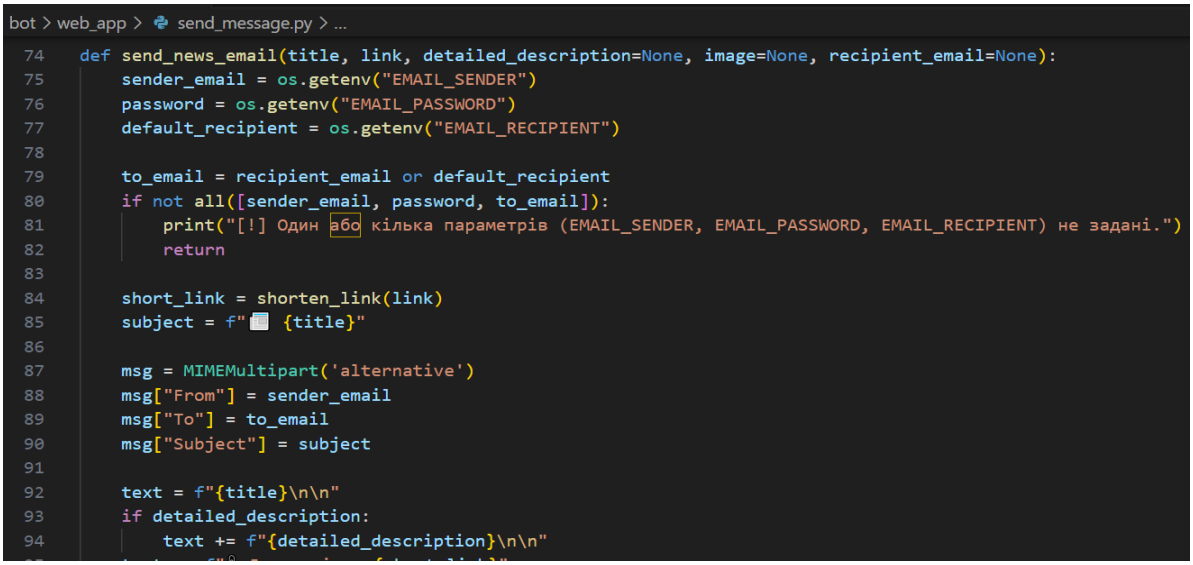

Рисунок 3.9 – Код SMTP-авторизації у send_message.py

Для надсилання реалізовано функцію send_email(subject, body, to_email), яка підключається до SMTP-сервера Google, створює текст повідомлення у форматі plain text або HTML, і відправляє його отримувачу.

На рисунку 3.10 показано код SMTP-авторизації у send_message.py в режимі керування.

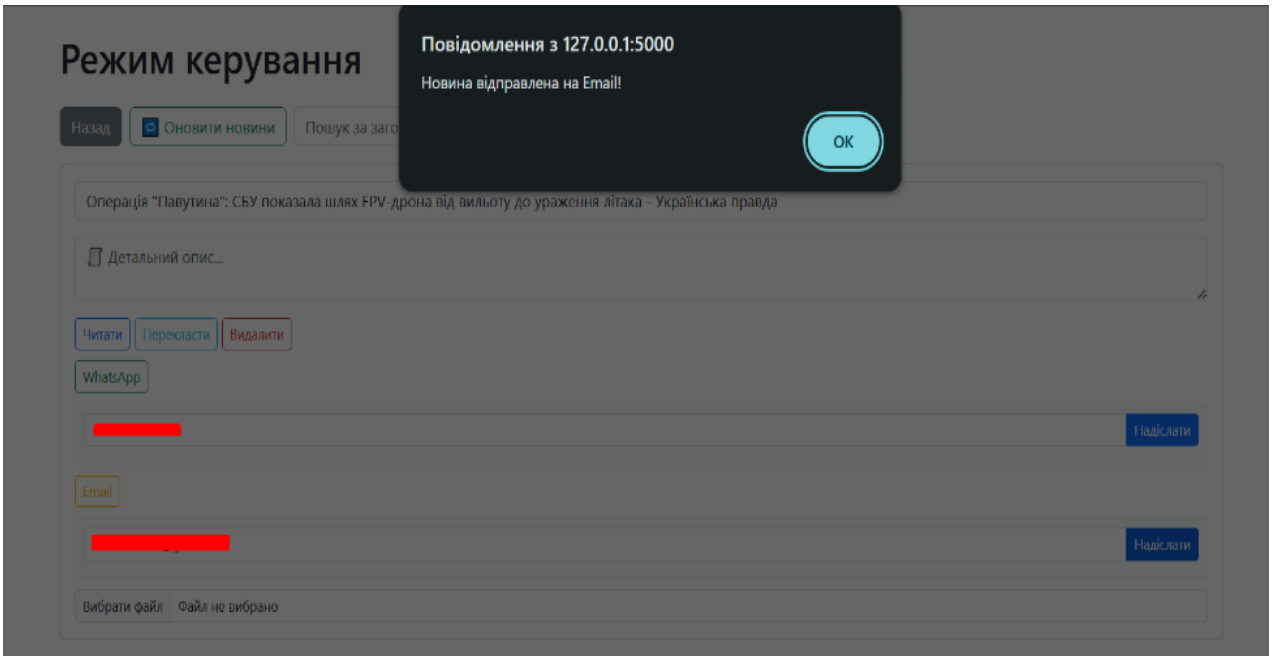

Рисунок 3.10 – Код SMTP-авторизації у send_message.py

Усі операції надсилання мають вбудовану обробку винятків. Помилки авторизації, недоступність API, неправильний формат номера чи email — усе це фіксується у log-файлі logs/bot.log із зазначенням часу, коду помилки та вмісту повідомлення.

Завдяки цьому:

1. забезпечується стабільність роботи системи;
2. спрощується діагностика проблем при роботі з API;
3. зберігається історія надсилань у лог-файлі.

Таким чином, було реалізовано два повноцінні канали надсилання новин — WhatsApp Cloud API та Email — із повною інтеграцією до інтерфейсу керування, можливістю конфігурації та логуванням усіх дій. Це дозволяє гнучко доставляти інформацію кінцевим користувачам без потреби у зовнішніх сервісах або складних налаштуваннях.

3.3 Безпека, тестування та надійність мультиагентної системи

Надійність функціонування мультиагентної системи новин значною мірою залежить від впроваджених засобів контролю, тестування та захисту даних. У межах реалізації проєкту було передбачено такі ключові аспекти:

1. Безпека зберігання конфіденційної інформації. Усі ключові змінні середовища, включаючи токени для WhatsApp Cloud API, SMTP-паролі та ідентифікатори, зберігаються не у відкритому коді, а у файлі конфігурації .env. Для роботи з ним використовується бібліотека python-dotenv, що дозволяє ізолювати конфіденційні параметри від основного коду:

```
WHATSAPP_TOKEN=EAAG...ZCZDZD
EMAIL_SENDER=myprojectnews@gmail.com
EMAIL_PASSWORD=app-password-here
```

Це мінімізує ризик витоку чутливої інформації при розгортанні системи на сервері або під час командної роботи.

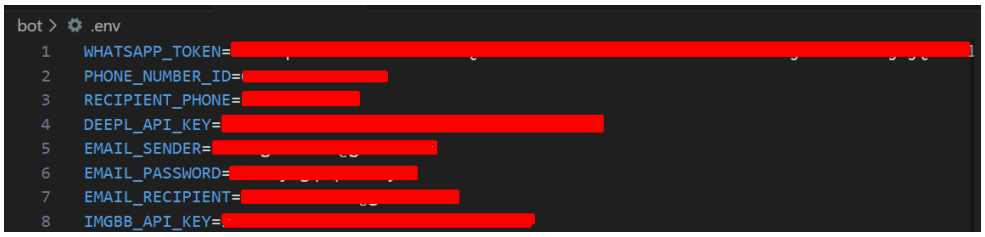

Рисунок 3.11 – Структура збереження ключів у файлі .env

Окрім цього, до сервера Flask обмежено доступ лише через локальні запити або авторизовані IP-адреси. Це дозволяє уникнути несанкціонованих викликів API ззовні.

2. Система логування подій. Для моніторингу внутрішніх процесів у системі було реалізовано детальне логування у файл logs/bot.log. Всі ключові події — від запуску агента до відповіді API — записуються з міткою часу, рівнем важливості (INFO, ERROR, DEBUG) та коротким описом. Приклади типових записів:

[2025-05-26 10:14:08] INFO Завантажено 12 новин з Google News.

[2025-05-26 10:14:11] INFO Переклад виконано успішно (DeepL).

[2025-05-26 10:14:12] ERROR Відповідь API WhatsApp: 401 Unauthorized – Invalid token.

Це дозволяє швидко знаходити та усувати проблеми без потреби в ручному відстеженні коду.

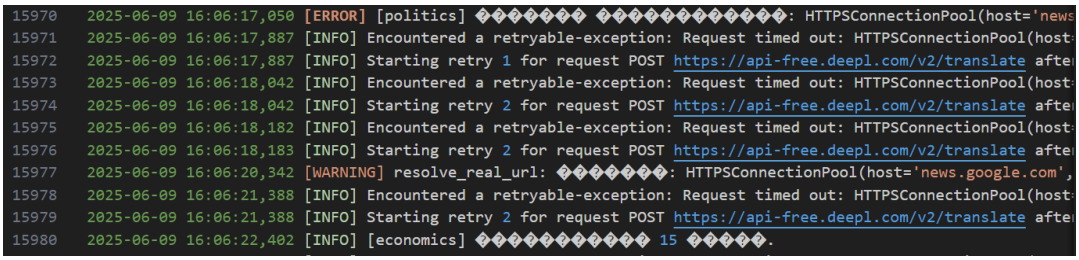

Рисунок 3.12 – Фрагмент лог-файлу logs/bot.log із повідомленням про помилку авторизації

Логування реалізовано стандартними засобами Python (logging.basicConfig(...)) з обмеженням розміру файлу, що дозволяє уникати надмірного зростання log-файлу на сервері.

3. Обробка помилок та винятків. Усі зовнішні API-запити (WhatsApp, Email, DeepL) та читання/запис JSON-файлів обгорнуті у try-except блоки. При виникненні помилки:

– створюється запис у bot.log;

– виводиться повідомлення в інтерфейсі або консолі;

– у критичних випадках процес перезапускається або скасовується.

Приклад коду обробки помилки запиту WhatsApp API:

```
try:
    response = requests.post(url, headers=headers, json=payload)
    response.raise_for_status()
except requests.exceptions.RequestException as e:
```

logging.error(f"Помилка при надсиланні у WhatsApp: {str(e)}")

4. Тестування функціоналу. Для перевірки працездатності окремих функцій (особливо API-інтеграції) створено модуль test_send.py, який дозволяє протестувати надсилання повідомлень без запуску повної системи. Також було реалізовано ручне тестування компонентів через інтерфейс manage.html, що дозволяє: вибрати конкретну новину; змінити дані перед надсиланням; перевірити форматування перед реальним повідомленням.

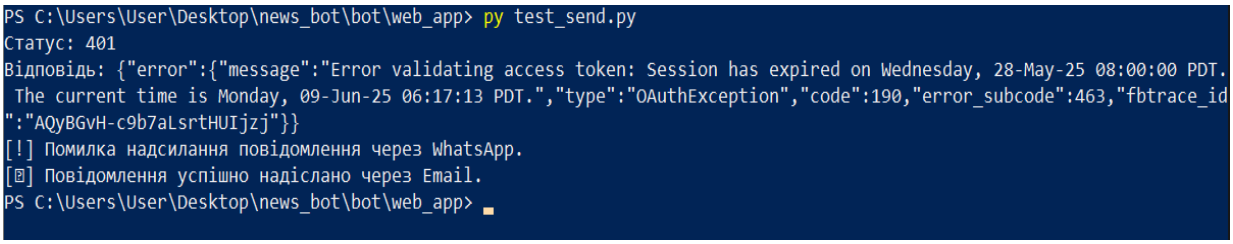

Рисунок 3.13 – Тестовий запуск надсилання через test_send.py

Окремо було протестовано сценарії з помилковим email, недоступністю API DeepL, пошкодженим JSON-файлом, відсутнім зображенням — усі ці ситуації правильно обробляються без аварійного завершення програми.

5. Надійність локального сховища. JSON-файли news_storage.json та news_general.json мають контроль цілісності: при зчитуванні перевіряється, чи є файл порожнім або пошкодженим. У такому випадку створюється резервна копія або виводиться повідомлення про відновлення.

Таким чином, реалізовано надійну систему тестування, логування та обробки помилок, яка дозволяє забезпечити безпечну роботу мультиагентної системи новин навіть у непередбачуваних умовах. Усі дії фіксуються та контролюються, що дозволяє гнучко масштабувати рішення в майбутньому або перенести його на продуктивний сервер.

3.4 Оцінка продуктивності та масштабованості системи

Одним із важливих аспектів функціонування інформаційних систем є їх здатність обробляти значні обсяги даних без втрати стабільності, точності та швидкодії. У межах цієї роботи було проведено оцінку продуктивності реалізованої мультиагентної системи новин, а також розроблено підходи до її масштабування у майбутньому.

Поточна реалізація системи охоплює повний цикл: збір новин → переклад → категоризація → збереження → відображення → надсилання. Вимірювання часу виконання основних етапів у середньому дали такі результати (на локальному сервері), що представлено в таблиці 3.4.

Таблиця 3.4 - Оцінка часу виконання базових операцій системи

Операція	Середній час виконання	Коментар
Збір 20 новин через RSS	3-5 с	Залежить від швидкості мережі
Переклад через DeepL API	1-2 с / новина	Обмеження безкоштовного тарифу
Категоризація	< 0.1 с / новина	Легка евристика
Запис у JSON	0.1 – 0.2 с	Локальне збереження
Надсилання через WhatsApp	2 – 3 с / повідомлення	API відповідає зі змінною затримкою
Надсилання через Email (SMTP)	1.5 – 2 с / лист	SMTP відповідає швидко

Загалом повний цикл обробки 10–15 новин займає не більше ніж 30–40 секунд, що є прийнятним для реального застосування. Завдяки асинхронності

запитів та попередньому кешуванню зображень/описів, завантаження інтерфейсу залишається швидким.

На рисунку 3.14 представлені результати тестування.

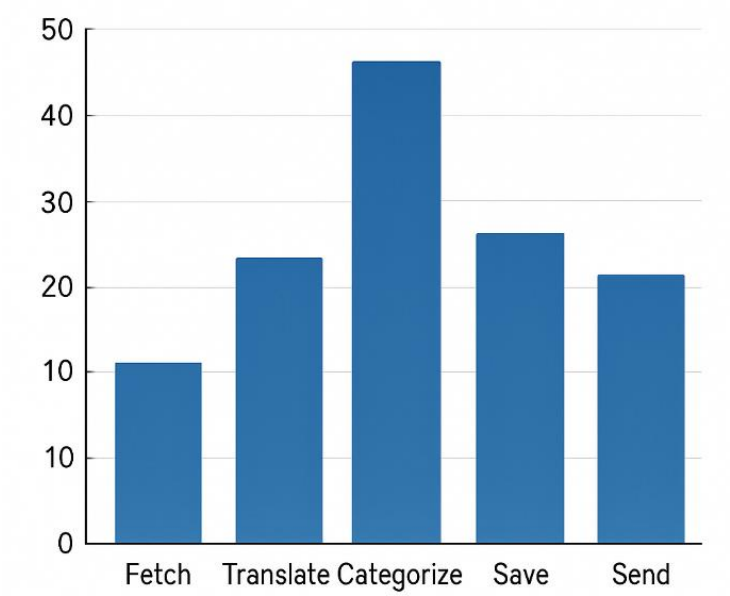

Рисунок 3.14 – Графік розподілу часу на етапах обробки однієї новини

На відміну від монолітного підходу, мультиагентна система дозволяє розділити кожен етап на окремі незалежні компоненти (агенти), які можуть працювати паралельно або на різних серверах.

Наприклад:

1. Збір новин (агент-збирач) можна розгорнути окремо для кожної категорії.
2. Переклад новин (агент-перекладач) — масштабувати під кожну мову окремо.
3. Надсилання — рознести на окремі процеси для WhatsApp, Email, Viber тощо.

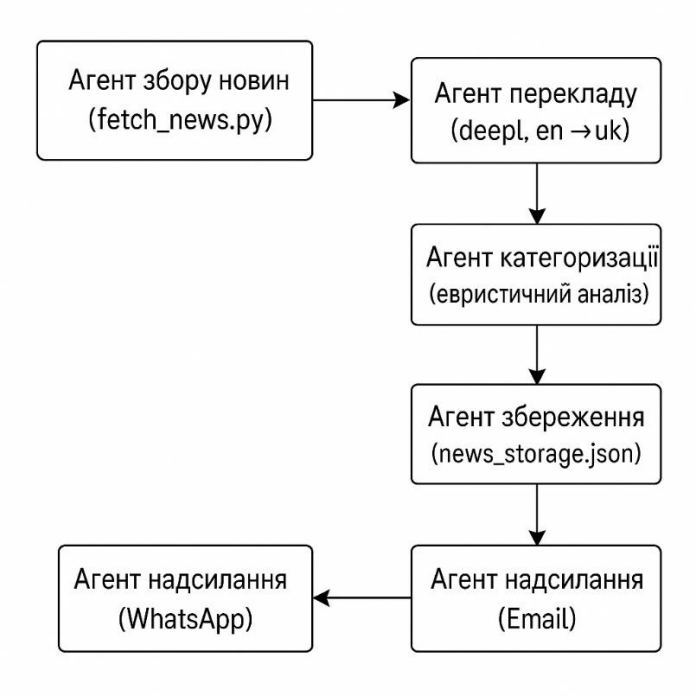

Рисунок 3.15 – Схема масштабованої мультиагентної архітектури

Для подальшого розвитку системи передбачено можливості розширення мультиагентної системи (табл.3.5).

Таблиця 3.5 - Можливості розширення системи

Напрям розширення	Потенціал реалізації
Підтримка нових джерел	Додавання інших агрегаторів (наприклад, Bing News)
Підтримка нових мов	Інтеграція перекладу з французької, німецької тощо
Перехід на базу даних	Використання SQLite [12] або MongoDB замість JSON
Додавання нових месенджерів	Viber, Signal Messenger
Статистика та аналіз	Побудова графіків, рейтингів, аналіз популярності новин

Масштабованість може також стосуватися інтерфейсу користувача — у разі зростання кількості новин можна реалізувати:

1. Пагінацію;
2. Динамічне завантаження;
3. Сортування за датою або релевантністю.

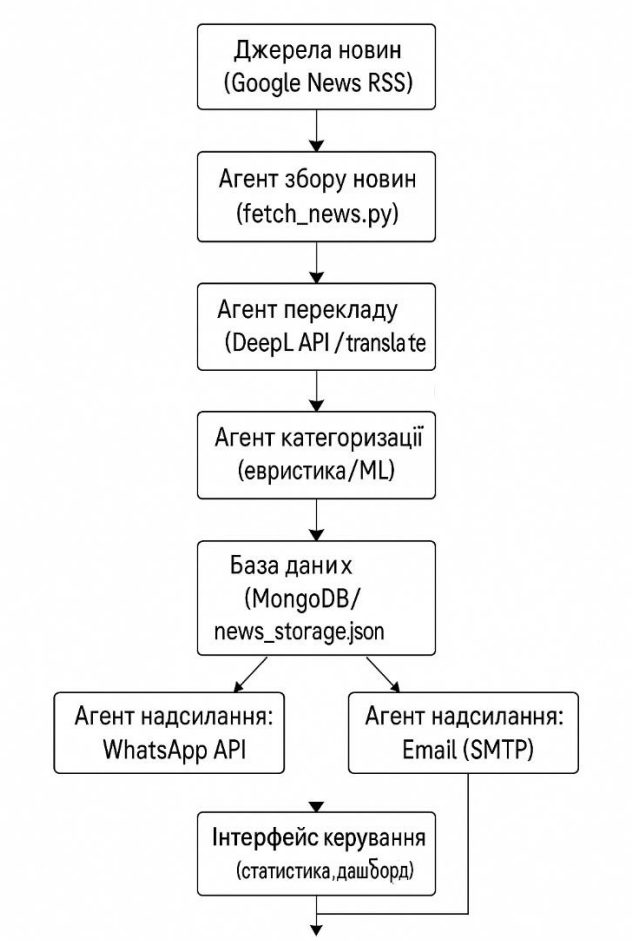

Рисунок 3.15 – Потенційна архітектура розширеної системи (V2)

Уся архітектура системи побудована з урахуванням майбутнього розширення — як у бік збільшення навантаження, так і функціональних модулів. Завдяки гнучкій структурі агентів, незалежності етапів і логуванню, система здатна масштабуватись без суттєвого втручання в ядро.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Аналіз небезпеки під час роботи за комп'ютером

Під час виконання кваліфікаційної роботи основна діяльність полягала у створенні та тестуванні програмного забезпечення, тобто в тривалій роботі за комп’ютером. На перший погляд, така робота здається безпечною — сидиш собі за столом і клацаєш клавішами. Але насправді вона має чимало прихованих ризиків, які можуть негативно впливати на здоров’я. Аналіз цих ризиків допомагає правильно організувати робоче місце й мінімізувати шкідливий вплив.

Насамперед варто згадати про електробезпеку. Електротравми можуть статися, якщо використовувати несправну техніку, працювати без заземлення або з пошкодженими кабелями. Щоб не наразити себе на небезпеку, важливо регулярно перевіряти стан обладнання, користуватися захисними вимикачами (типу УЗО) та уникати перевантаження розеток.

Ще одна поширена проблема — перевтома очей. Пильне вдивляння в екран протягом кількох годин поспіль призводить до сухості очей, погіршення зору та головного болю. Щоб уникнути цього, варто кожні 60 хвилин робити невелику перерву — хоча б на 10–15 хвилин, щоб дати очам відпочити.

Також варто звернути увагу на статику — тривале сидіння в незручному положенні без належної підтримки спини та шиї може спричинити біль у м’язах і навіть порушення постави. Тому ергономіка — не просто модне слово, а необхідність. Зручне крісло з регулюванням висоти, правильна висота стола, монітор на рівні очей — усе це допомагає уникнути неприємностей із хребтом.

Не менш важливим є і психоемоційне навантаження. Постійна концентрація, дедлайни, складні технічні завдання — усе це може виснажувати морально. У таких умовах легко втратити мотивацію або опинитися на межі вигорання, тому важливо не забувати про баланс: робота — це важливо, але відпочинок — не менш.

Щодо електромагнітного випромінювання, то завдяки сучасним екранам його рівень значно зменшився. Але все ж не варто працювати впритул до монітора — рекомендована відстань до нього становить 50–70 см.

Щоб знизити вплив усіх цих факторів, робоче місце повинно відповідати певним стандартам. Освітлення — не менше 300 лк, рівномірне, без відблисків. Температура в приміщенні — 20–24 °C, вологість — 40–60 %, а на кожного працівника має припадати не менше 4,5 м². Звісно, добре, якщо крісло підтримує поперек, а стіл відповідає росту.

Підсумовуючи: правильна організація простору, регулярні перерви, дотримання санітарних норм і використання якісної техніки — усе це не просто рекомендації, а ключ до збереження здоров'я в умовах інтенсивної комп'ютерної роботи. Бо навіть найкращі проєкти не варті вашого самопочуття.

4.2 Безпека в надзвичайних ситуаціях

У нинішніх реаліях, особливо в умовах воєнного стану в Україні, питання безпеки в надзвичайних ситуаціях стало як ніколи актуальним. Виконання дипломної роботи здебільшого проходило дистанційно — вдома або в тимчасовому житлі, за комп'ютером, із використанням засобів зв'язку та інтернету. Такий формат роботи, попри зручність, вимагав постійної готовності до різноманітних ризиків: від повітряних тривог і ракетних обстрілів до пожеж, аварій на комунальних мережах чи раптового вимкнення світла.

Найімовірніші надзвичайні ситуації під час роботи — це повітряна тривога, ракетна атака, відключення електроенергії, пожежа або хімічна небезпека. Якщо лунає сигнал тривоги — потрібно миттєво зберегти всі файли, вимкнути техніку від електромережі та прямувати до найближчого укриття. Якщо його немає, варто сховатися в безпечному місці без вікон — наприклад, у ванній або коридорі, щільно зачинити вікна й двері та дотримуватись правил поведінки під час обстрілу.

У разі відключення світла важливо мати під рукою powerbank, ліхтарик або джерело безперебійного живлення (UPS). Усі пристрої бажано підключати через подовжувачі з захистом від перепадів напруги.

Якщо виникла пожежа — слід негайно припинити роботу, вимкнути всі прилади, викликати пожежників за номером 101 і, за можливості, спробувати загасити вогонь первинними засобами: вогнегасником, водою, піском або навіть ковдрою. Та якщо є реальна загроза життю — головне не зволікати й евакуюватися в безпечне місце.

Не варто забувати й про техногенні ризики — наприклад, витік хімічних речовин або аварії на промислових об’єктах. У разі хімічної тривоги необхідно щільно зачинити всі вікна, двері та вентиляційні отвори, ввімкнути радіо чи телевізор, підготувати засоби захисту — маску, респіратор або хоча б вологу тканину — і слухати вказівки рятувальників.

Наявність чітких інструкцій, розуміння, як діяти в тій чи іншій ситуації, а також елементарна підготовка — аптечка, копії документів, ліхтарик — значно підвищують шанси зберегти здоров’я, життя і техніку навіть у найскладніших обставинах.

У підсумку — своєчасна реакція на тривогу, дотримання правил безпеки та відповідальне ставлення до організації робочого місця допомагають мінімізувати ризики, навіть коли доводиться працювати в умовах війни або потенційної катастрофи.

4.3 Аналіз травмонебезпечних ситуацій під час виконання робіт

Хоча програмування та інші види інтелектуальної діяльності зазвичай вважаються безпечними з точки зору фізичного навантаження, на практиці вони також можуть супроводжуватися ризиками для здоров’я.

Під час реалізації дипломного проєкту виникають як технічні, так і організаційні чинники, що потребують уваги.

Одним із найпоширеніших джерел небезпеки є порушення правил використання електрообладнання. Робота за комп'ютером без заземлення, з пошкодженими кабелями, несправними подовжувачами або перегрітими блоками живлення може призвести до короткого замикання, займання або ураження струмом. Це не лише створює загрозу життю, а й може пошкодити техніку чи зірвати важливий етап роботи.

Ще один фактор ризику — невдало організоване робоче місце. Якщо техніка стоїть на нестійких поверхнях, дроти хаотично прокладені або розетки перевантажені — це прямий шлях до падіння обладнання, спотикання або навіть травм.

Не менш важливим є й мікроклімат у приміщенні. Задушливе середовище, недостатнє освітлення, відсутність вентиляції чи порушення температурного режиму — усе це негативно впливає на самопочуття: виникають головний біль, втома, зниження концентрації та ефективності.

Окремо слід згадати про психоемоційні навантаження. Постійна розумова напруга, робота без достатніх перерв, відповідальність за результат, а також зовнішні чинники (наприклад, стрес через воєнний стан) можуть викликати хронічну втому, нервове виснаження або навіть професійне вигорання.

Щоб знизити ризики, варто дотримуватись кількох простих, але важливих рекомендацій:

– регулярно перевіряти справність кабелів, розеток і обладнання;

– організовувати простір так, щоб нічого не заважало роботі й пересуванню;

– дотримуватись режиму праці: перерви щогодини

– не примха, а потреба;

– контролювати освітлення, температуру та вологість повітря;

– уважно ставитися до свого самопочуття — давати собі відпочинок.

У результаті, навіть якщо робота не пов’язана з фізичними зусиллями або небезпечним обладнанням, вона все одно потребує дотримання правил охорони праці. Адже здоров’я — основа продуктивності, і турбота про нього має бути пріоритетом у будь-якій професійній діяльності.

Окрім згаданих небезпек, серйозну загрозу становлять ергономічні порушення. У багатьох випадках розробники, студенти або офісні працівники використовують невідповідне крісло, не регулюють висоту монітора чи клавіатури, що в результаті призводить до перенавантаження шийного та грудного відділів хребта. Найчастіше це проявляється у вигляді хронічного болю в шиї, спині та плечах, а також відчуття скутості й зниження продуктивності.

Додаткову небезпеку несе порушення кровообігу через довге сидіння в одній позі. Це може стати передумовою до варикозного розширення вен, оніміння кінцівок, погіршення постачання кисню до мозку. Щоб цього уникнути, рекомендується кожні 45–60 хвилин вставати, пройтися, зробити хоча б прості фізичні вправи (нахили, обертання голови, розминка кистей тощо).

Ще один фактор ризику — перевтома зору. Через тривале вдивляння в монітор на близькій відстані може виникнути так званий "комп’ютерний зоровий синдром". Симптоми включають: сухість очей, двоїння зображення, зниження різкості зору, головний біль. Уникнути цього допомагає правило 20–20–20: кожні 20 хвилин дивитися протягом 20 секунд на об’єкт, розташований щонайменше на 20 футів (6 метрів) від очей.

Також важливе значення має якість освітлення: яскравість ламп має бути достатньою, але не надмірною; світло повинно бути розсіяним, без прямих відблисків на екрані; бажано, щоб джерело освітлення було зліва або спереду — це знижує навантаження на очі та зменшує контраст між екраном і навколишнім середовищем.

Температурно-вологісний режим — ще один важливий аспект. Якщо в кімнаті душно, жарко або дуже сухо, концентрація значно знижується,

підвищується втома. Рекомендовані параметри для приміщення: температура повітря — 20–24 °C; вологість — 40–60%; бажано — наявність природної вентиляції або кондиціонування.

Особливу увагу слід приділити профілактиці професійного вигорання. Робота над дипломом передбачає високий рівень концентрації, дедлайни, відповідальність за результат, потребу швидко вирішувати складні технічні задачі. Якщо не дотримуватися режиму відпочинку, не розділяти роботу й особистий простір, не перемикатися — це може призвести до нервового виснаження, роздратованості, відсутності мотивації.

Щоб цього уникнути, потрібно: планувати день із чітким розподілом праці та відпочинку; дотримуватись режиму сну (не менше 7–8 годин на добу); чергувати інтелектуальне навантаження з фізичною активністю; за можливості змінювати обстановку (кава-брейк, прогулянка, спорт); не нехтувати емоційною підтримкою — спілкування з друзями, рідними тощо.

На завершення варто підкреслити, що здоров’я — не другорядна деталь, а невід’ємна частина будь-якої розробки чи навчального процесу. Навіть найбільш інтелектуальна праця, якщо вона виконується в несприятливих умовах, може спричинити серйозні наслідки. Саме тому важливо не лише реалізувати поставлену задачу, але й зробити це безпечно для себе.

ВИСНОВКИ

Сьогодні інформація стала основним ресурсом — її отримання, аналіз та доставка в найкоротші терміни є критично важливими як для звичайного користувача, так і для медіа, бізнесу та держави. Саме тому розробка систем, що можуть автоматично відстежувати події, аналізувати новини та надсилати їх у зручному форматі, є актуальною й затребуваною. У цій кваліфікаційній роботі було реалізовано проєкт мультиагентної системи, здатної самостійно збирати новини з відкритих джерел, перекладати їх, класифікувати за категоріями та надсилати користувачам через зручні канали — WhatsApp або Email.

Під час реалізації проєкту вдалося побудувати повноцінну архітектуру, яка ґрунтується на ідеї поділу логіки на окремі агенти. Це дало можливість створити гнучку й масштабовану систему, яку в подальшому можна розширювати без суттєвих змін у коді. Завдяки чіткому поділу функціоналу та використанню сучасних технологій (Flask, DeepL API, WhatsApp Cloud API), система працює стабільно, дозволяє легко управляти новинами, а також надає простий, інтуїтивно зрозумілий інтерфейс.

У першому розділі роботи були розглянуті теоретичні основи мультиагентних систем та аналіз аналогічних рішень. Це дало змогу зрозуміти сильні та слабкі сторони вже наявних платформ, а також обґрунтувати вибір саме мультиагентної моделі.

Другий розділ був присвячений проєктуванню системи: описано її архітектуру, структуру взаємодії агентів, схему зберігання даних, а також принципи створення користувацького інтерфейсу. Це дозволило грамотно розбити проєкт на логічні складові, що значно спростило реалізацію.

У третьому розділі було реалізовано практичну частину: створено агенти для збору новин, перекладу та категоризації, збереження інформації у форматі JSON, а також модулі для надсилання новин через месенджери.

Особливу увагу приділено надійності системи, логуванню, обробці помилок та можливості масштабування. Проведено оцінку продуктивності, і результати показали, що система може стабільно працювати навіть при зростанні навантаження.

Четвертий розділ був присвячений питанням охорони праці та безпеки. Було розглянуто основні ризики, що можуть виникати під час роботи за комп’ютером, а також дії у надзвичайних ситуаціях. Це дозволило врахувати як технічні, так і організаційні аспекти безпечної праці.

Під час розробки цієї системи я не лише поглибив свої знання у сфері Python-програмування, API-інтеграції та розробки веб-інтерфейсів, а й отримав практичний досвід побудови програмного продукту з нуля. Окрему цінність становить те, що ця система може бути використана як основа для більш складних проєктів, включаючи автоматизовані стрічки новин, моніторинг інформаційного простору, або навіть прототипи для аналітичних центрів.

У результаті виконання кваліфікаційної роботи створено реальний програмний продукт, який відповідає сучасним вимогам і здатен виконувати поставлені задачі ефективно, надійно та з урахуванням потреб користувача.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Огляд мобільного застосунку Flipboard. URL: https://icoola.ua/blog/ohliad-flipboard (дата звернення 10.04.2025 р)

2. Офіційна документація Flask. URL: https://flask.palletsprojects.com/en/2.3.x/ (дата звернення 15.04.2025 р)

3. Офіційна документація DeepL API. URL: https://www.deepl.com/docs-api (дата звернення 22.04.2024 р)

4. Офіційна документація WhatsApp Cloud API. URL: https://developers.facebook.com/docs/whatsapp (дата звернення 22.04.2025 р)

5. Vincent W. Django for Beginners. URL: https://djangoforbeginners.com/introduction/ (дата звернення 03.05.2025 р)

6. Krug S. Don't Make Me Think: A Common Sense Approach to Web Usability. 2013. 216 p.

7. Martin R.C. Clean Code: A Handbook of Agile Software Craftsmanship. 2009. 464 p.

8. Лутц М. Програмування на Python. Київ: Діалектика, 2011. 992 с.

9. Джон Дакетт. HTML and CSS: Design and Build Websites. URL: http://surl.li/rvgjq (дата звернення 28.04.2024 р)

10. Keith J. HTML5 for Web Designers. Second Edition. A Book Apart, 2015. 120 p.

11. Трофименко О.Г., Козін О.Б., Задерейко О.В., Плачінда О.Є. Веб-технології та веб-дизайн: навч. посібник. Одеса: Фенікс, 2019. 284 с.

12. Офіційна документація SQLite. URL: https://www.sqlite.org/docs.html (дата звернення 20.04.2024 р)

13. Маловичко С.В. Аналіз сучасних тенденцій та динаміки розвитку електронної торгівлі на підприємствах України. Економіка та управління народним господарством. №2. 2015. С. 65–69.

14. Андрій Пінчук. Будівництво успішного бренду. Академічна та

університетська наука: результати та перспективи. 2020. С. 272–285.

15. Панченко М. Впровадження новинного сервісу на Flask: приклад проєкту.

16. Корнієнко В. Дослідження та розробка мультиагентної системи доставки новин на базі фреймворку Flask.

17. Інструкція з охорони праці при роботі з персональним комп'ютером. URL: https://pro-op.com.ua/article/485-nstruktsya-z-ohoroni-prats-pri-robot-na-personalnomu-kompyuter (дата звернення 18.05.2025 р)

18. ДСанПіН 3.3.2.007-98 «Гігієнічні вимоги до відеодисплейних терміналів, персональних електронно-обчислювальних машин і організації праці».

19. ДБН В.2.2-3:2018 Будинки і споруди. Будинки та споруди закладів охорони здоров'я.

20. ГОСТ 12.1.005-88 Загальні санітарно-гігієнічні вимоги до повітря робочої зони.

21. API Documentation: Python Requests. URL: https://requests.readthedocs.io/en/latest/ (дата звернення 17.04.2025 р)

ДОДАТКИ

Додаток А

Фрагмент коду для надсилання новин через WhatsApp Cloud API

send_message.py

```
def send_news_whatsapp(title, link, detailed_description=None, image=None,
recipient=None):
    phone_number_id = os.getenv("PHONE_NUMBER_ID")
    token = os.getenv("WHATSAPP_TOKEN")
    default_recipient = os.getenv("RECIPIENT_PHONE")

    recipient_phone = recipient or default_recipient
    if not all([phone_number_id, token, recipient_phone]):
        print("[!] Один або кілька параметрів (PHONE_NUMBER_ID,
WHATSAPP_TOKEN, RECIPIENT_PHONE) не задані.")
        return

    short_link = shorten_link(link)
    message = f"🗞 {title}\n\n"
    if detailed_description:
        message += f"{detailed_description}\n\n"
    message += f"📎 Докладніше: {short_link}"

    url = f"https://graph.facebook.com/v18.0/{phone_number_id}/messages"
    headers = {
        "Authorization": f"Bearer {token}",
        "Content-Type": "application/json"
    }

    if image:
        data = {
            "messaging_product": "whatsapp",
            "to": recipient_phone,
            "type": "image",
            "image": {
                "link": image
            },
            "caption": message
        }
    else:
        data = {
            "messaging_product": "whatsapp",
            "to": recipient_phone,
```

```
        "type": "text",
        "text": {
            "body": message
        }
    }

try:
    response = requests.post(url, headers=headers, json=data)
    print("Статус:", response.status_code)
    print("Відповідь:", response.text)

    if response.status_code != 200:
        print("[!] Помилка надсилання повідомлення через WhatsApp.")
    else:
        print("[✓] Повідомлення успішно надіслано через WhatsApp.")
except Exception as e:
    print(f"[!] Виняток при надсиланні через WhatsApp: {e}")[
```

Додаток Б

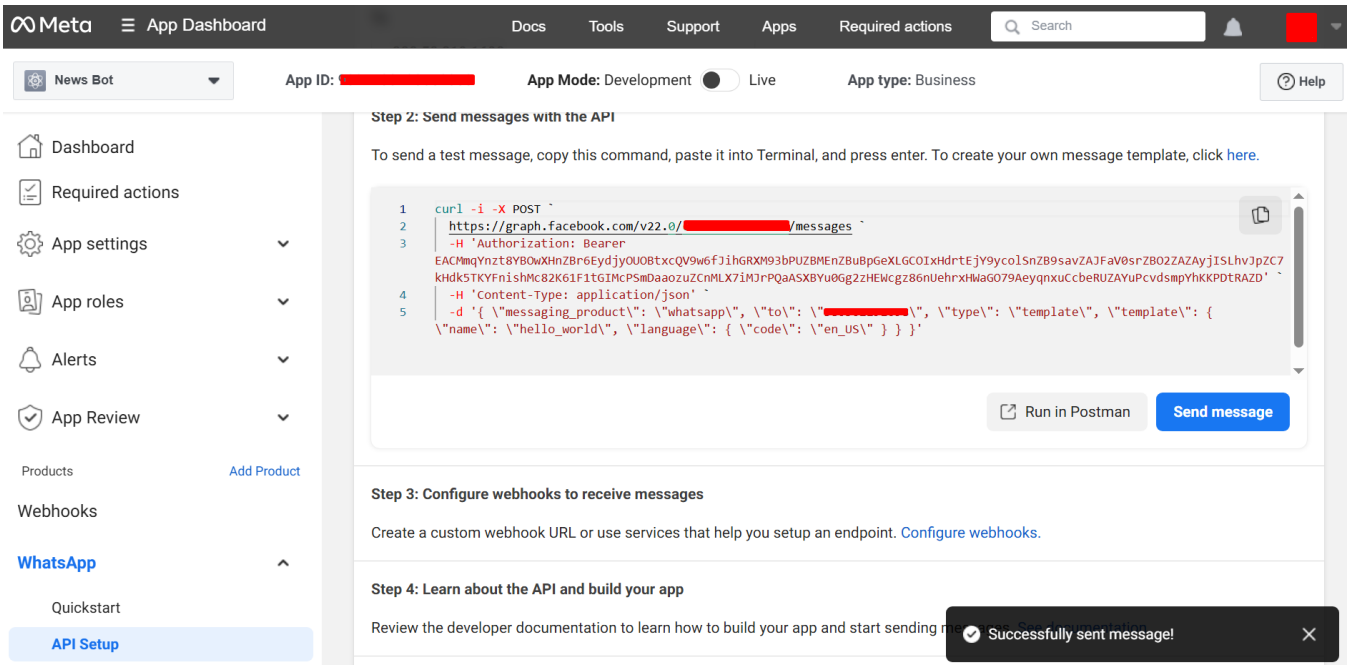

Рисунок 1 - Шаблон повідомлення, створений у Meta Business Suite

Додаток В

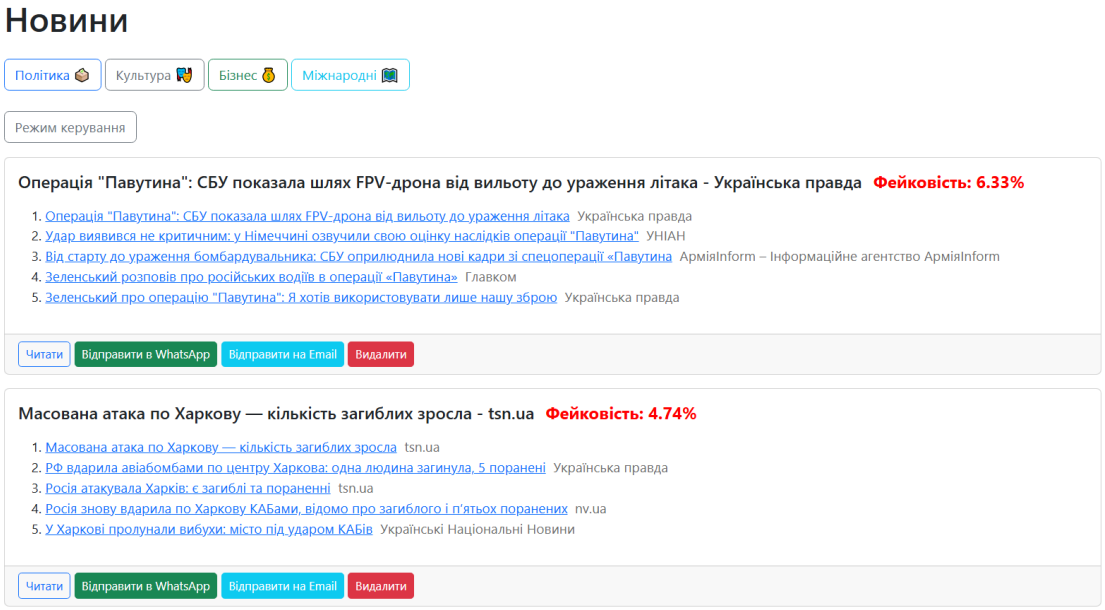

Рисунок 1 - Скріншот інтерфейсу системи home.html

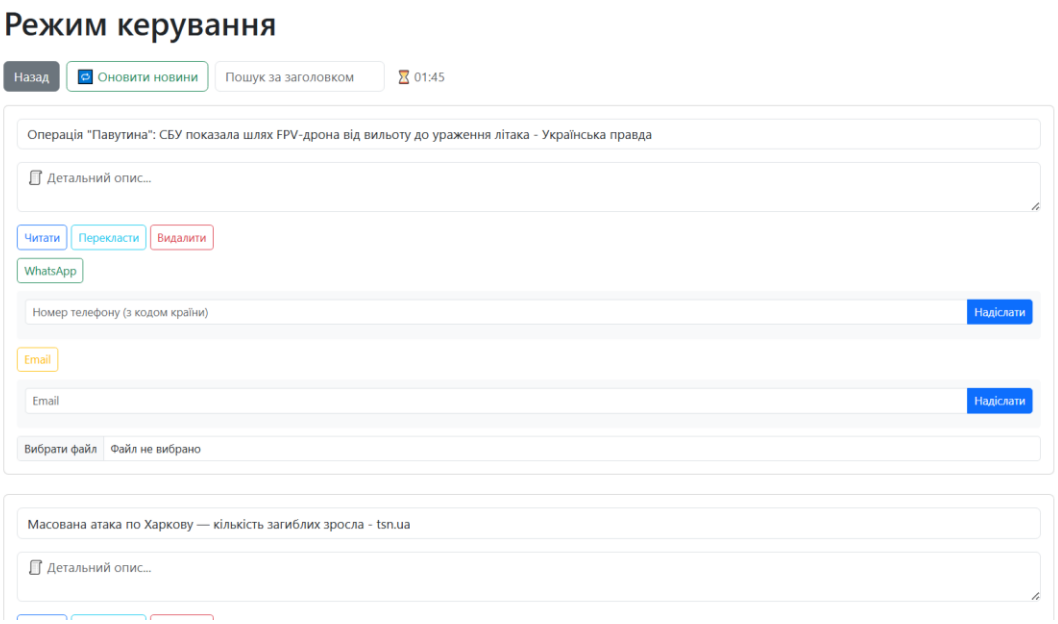

Рисунок 2 - Скріншот інтерфейсу системи manage.html

Додаток Г

Лог-файл прикладу доставки (logs/bot.log)

2025-06-07 18:33:20,339 [INFO] Автооновлення новин виконано, загальна кількість: 48

2025-06-07 18:43:20,543 [INFO] [general] Завантажено 30 новин.

2025-06-07 18:43:20,566 [INFO] [general] Завантажено 30 новин.

2025-06-07 18:43:24,997 [INFO] [politics] Завантажено 49 новин.

2025-06-07 18:43:25,038 [INFO] [politics] Завантажено 49 новин.

2025-06-07 18:43:28,976 [INFO] [economics] Завантажено 21 новин.

2025-06-07 18:43:29,054 [INFO] [economics] Завантажено 21 новин.

2025-06-07 18:43:33,093 [INFO] [culture] Завантажено 70 новин.

2025-06-07 18:43:33,133 [INFO] [culture] Завантажено 70 новин.

2025-06-07 18:43:36,867 [INFO] [en] Завантажено 38 новин.

2025-06-07 18:43:36,927 [INFO] [en] Завантажено 38 новин.

2025-06-07 18:43:40,595 [INFO] Загалом завантажено 49 новин з 5 категорій.

2025-06-07 18:43:40,601 [INFO] Новини збережено успішно.

2025-06-07 18:43:40,601 [INFO] Автооновлення новин виконано, загальна кількість: 49

2025-06-07 18:43:40,686 [INFO] Загалом завантажено 49 новин з 5 категорій.

2025-06-07 18:43:40,691 [INFO] Новини збережено успішно.

2025-06-07 18:43:40,691 [INFO] Автооновлення новин виконано, загальна кількість: 49

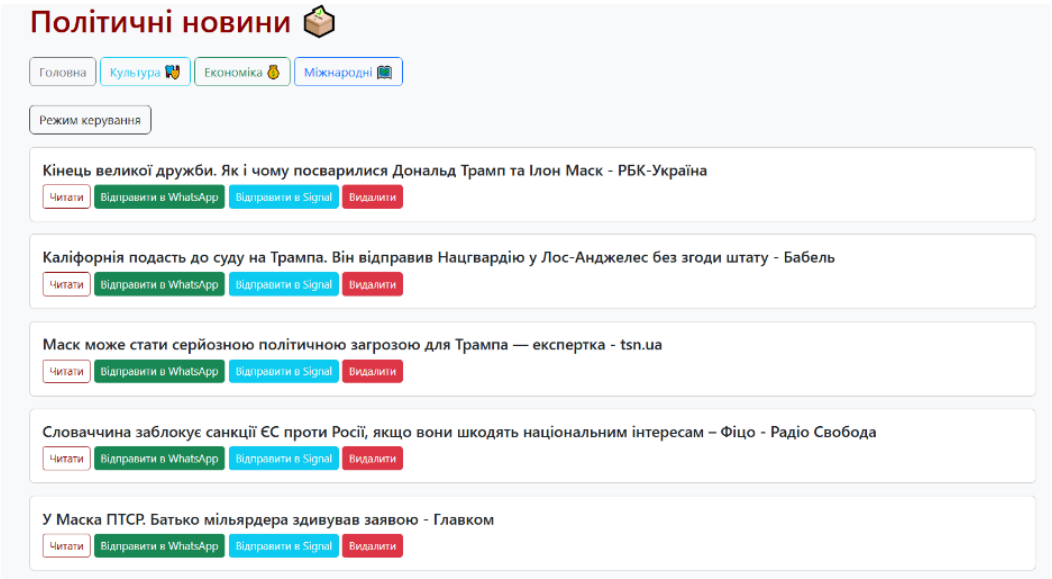

Рисунок 1 – Відображення новин у категорії «Політика»

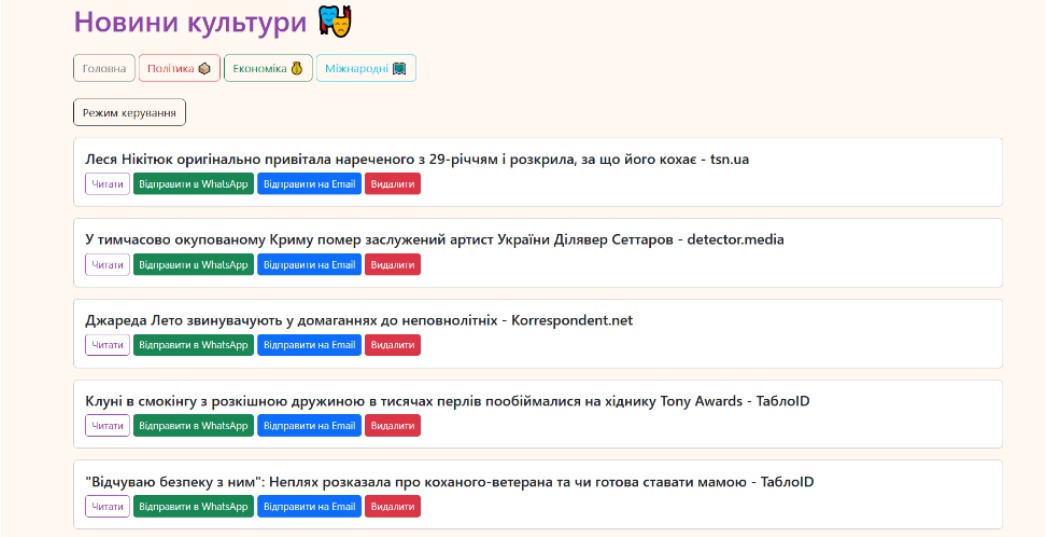

Рисунок 2 – Відображення новин у категорії «Культура»

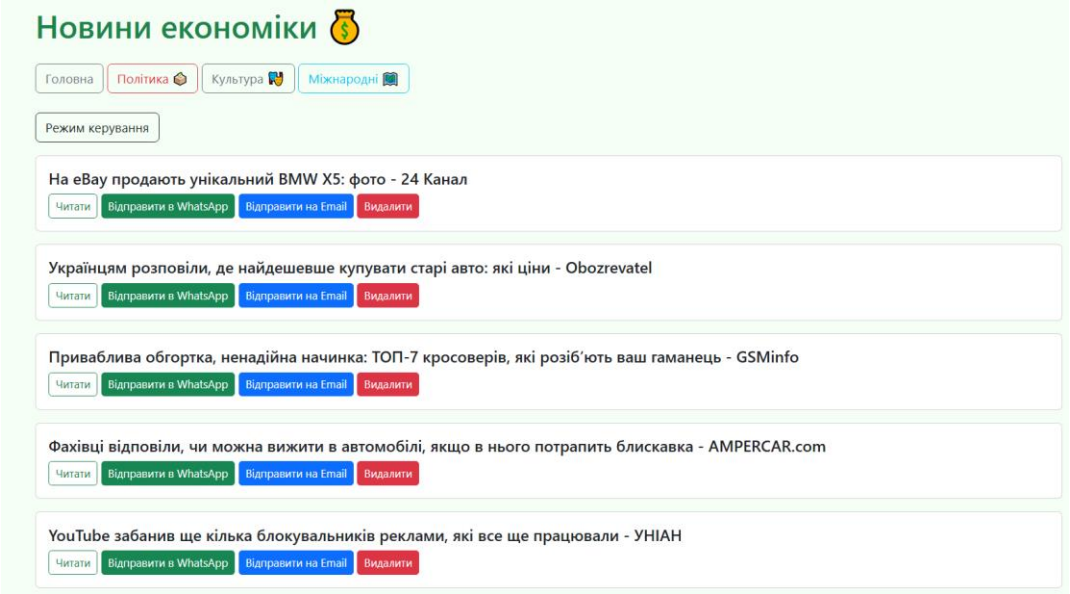

Рисунок 3 – Відображення новин у категорії «Бізнес»

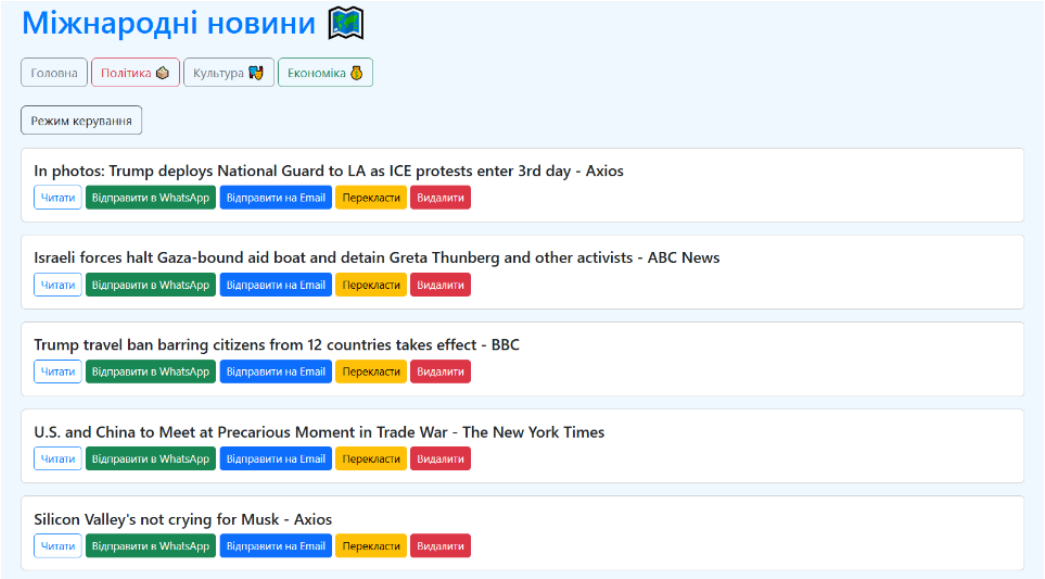

Рисунок 4 – Відображення новин у категорії «Міжнародні»

Додаток Д

Фрагмент збереженої новини у форматі JSON


```
{
  "id": 1,
  "title": "Операція \"Павутина\": СБУ показала шлях FPV-дрона від вильоту до ураження літака - Українська правда",
  "link": "https://news.google.com/rss/articles/CBMiXkFVX3lxTE1WM1M2MDU1YmY4RnpmZFJCNmJuYUpkUzNYOS1uUWxaZzlXQWMzN1BjVkhfWTZ5bllRUHYtRE16dkRwaFl2cllhYXVnRVpEYzJIekEyRW5RYkdnZkNEblE?oc=5",
  "description": "<ol><li><a href=\"https://news.google.com/rss/articles/CBMiXkFVX3lxTE1WM1M2MDU1YmY4RnpmZFJCNmJuYUpkUzNYOS1uUWxaZzlXQWMzN1BjVkhfWTZ5bllRUHYtRE16dkRwaFl2cllhYXVnRVpEYzJIekEyRW5RYkdnZkNEblE?oc=5\" target=\"_blank\">Операція \"Павутина\": СБУ показала шлях FPV-дрона від вильоту до ураження літака</a>  <font color=\"#6f6f6f\">Українська правда</font></li><li><a href=\"https://news.google.com/rss/articles/CBMisAFBVV95cUxNMlRkRjNvMkJMSXVDN2pMb0d1d05YTDU2TURNeGo3LVBqZ1Njd1dXRkxXaUlHOG5pMTlwYlQyVEZnSTRsUmNSN3FWRFh4dHVKblF0eHZxQWlWOXM5SFJNVkdCX3RPalVlYzlWdkY2V0dhWW9sVmpVc1NWa09ZNjRlNzNvZTZ3SnNYc21CNzNkOUgzVEhteUxObExVVzVLSUViVGNxSmxZMk05NG1FcHktMNIBxwFBVV95cUxPRkZlNzVzSUItVjFLdV8zajltN2E0MmhvVG9FVi1KS1pnUUE4bEU4akVEQVEzWE00Z3h4OWloMFZ0TE12TTFveUlrcmFPWFduVWlnb05YWFAwYW9ReC1kU1RxeG9jaU5EMDFUampKQmdCYktja09SNEw2cXV6STJMZ2RnZ29JbVBBekRjd1NXVWNuTWI1dEFuTXVFS2hDM2Q2TGtuT3lSVVpfSVVnMGxfZ0JPN25iVW4yRTJ3NWxreWhpdDMwT3hj?oc=5\" target=\"_blank\">Удар виявився не критичним: у Німеччині озвучили свою оцінку наслідків операції \"Павутина\"</a>  <font color=\"#6f6f6f\">УНІАН</font></li><li><a href=\"https://news.google.com/rss/articles/CBMizwFBVV95cUxQaXFTcmlOeTItT3hfZmQ1dHhlQ01uOWJkVUN3QzJ0UGFqQXY5MnlTNkhiTHlyaExNQlhOcmlOZmdsZ01IS2VZb2xudXFVeUpkajBkNG45c1VHMEJWbVd3c0NLOEl2QVBTS29BSDMwaHI1M1pENFJQUWl5dFZnejcxaHItRThleXJWaGZNNllVMndvM3QtSkpZT3lPOGVQSk91c21NVFphMVBZcm1LaGdaZ0c4c250bWZ0QUs1c1BJaHlKdTZJcWgxNmlLb1kzMTQ?oc=5\" target=\"_blank\">Від старту до ураження бомбардувальника: СБУ оприлюднила нові кадри зі спецоперації «Павутина</a>  <font color=\"#6f6f6f\">АрміяInform – Інформаційне агентство АрміяInform</font></li><li><a href=\"https://news.google.com/rss/articles/CBMiswFBVV95cUxOd3RlNUF1eGp5Tms2RXgzb2hBR3g0MjduSnRfRmN6MWxBR1hVM1hDZ3VqNTg3VFFyeHN4WW1ua21iQndwY3VLNEtNdTFDNk1TbjJCOW1pOFF3dU9rczZKcmEtQmg2NERtUGxfZFBvRkxoU2FZZnl6Tm1yd3BILWtEdW9nMU9qS2lPMHBvRGFTaTJQOHFUQV9YS1BsaVhOUDRYTVcwWnFVd09KUlFQQkc3cmRxNNlBuAFBVV95cUxPUDBmSnRQSVlacURMWWYyc2ZBcmFIcXlvTmxLaEhQTUU4eE5xSTZLWFVDcm8tU
```

i1XVTVZM0lEaU9kejczVlBoOVJqYUhvN3dDZGo1UXNmcXZHcGV1Mmg2eHBj
SkRGZnY2X2hHMExNNVV2OVQ3aWZxLXhkUGxsWUF2R2lnTVBaOXB1Q1Bf
MVFxcEJOM3N4S3BXR3Nhd3BDZWI0dmptSjBtQnB4VndxRk40UW9Ld3JzVF9q
?oc=5\" target=\"_blank\">Зеленський розповів про російських воділів в операції «Павутина» Главком
<a
href=\"https://news.google.com/rss/articles/CBMiXkFVX3lxTFBQakF5Q1kyNFZqb0
dDNVd3bGFUV1dINVc1elZEYnRYdllkcm0xU2NaVFcwSkxadEh3M3pDVmtORn
MzQVJTNkViY1dWSWt2TDlrNlQ4YUdicWI1R3ItQnfSAWpBVV95cUxQLWtCS1
VIblFzZjBBRVpIczVHc0kyejEwNEVPTXVpazhoS012QlZEZWNEU0tXUTN4XzR
UV2NNQjJDQTUxM2Q4UzROREluakJBTVFGUDRzMmJIaXN4U0hJdUxoTWtoc3
Z3azlR?oc=5\" target=\"_blank\">Зеленський про операцію \"Павутина\": Я хотів використовувати лише нашу зброю Українська правда
",
    "image": null,
    "category": "general",
    "fake_score": 6.33
  },
  {
    "id": 2,
    "title": "Масована атака по Харкову — кількість загиблих зросла - tsn.ua",
    "link":
"https://news.google.com/rss/articles/CBMikgFBVV95cUxNRERPaHlLdDVEdDdYR
ThNT05JRElHUEVVU0RmUjZvdHhodXJLaWV1djVsb0pleEFjZ2pyMFN1TDZwV
3JQZDFyTVlGbVVtLTNZZVEzSlZ1TWhkU01VRVdPdzdMOFlBdTF4czAwcnlEW
ko2OFQxX1prX3RGSy1FRmI4X2V3NnpMYXpodVBXaDNPc0RXZ9IBlwFBVV95
cUxQQWhFX0djZnR6N3pDRVh5NVFHZXJaLVV1YjB4T3NNSDdWWjNsdWhUO
Gt0T2hkSXRSNGRMSjM2TjE0TDAtR2J3Zm5PVEtiZEt6bnR3Z0FwZXVnQVlhSn
c0dlIzenF3TUJ5d1RPQ1JRaFQ3NkF3dHlOYW8ta1RVTVk5ZHltSERvcUhEMWJjO
XB5SUl4dVBma213?oc=5",
    "description": "
<a
href=\"https://news.google.com/rss/articles/CBMikgFBVV95cUxNRERPaHlLdDVEd
DdYRThNT05JRElHUEVVU0RmUjZvdHhodXJLaWV1djVsb0pleEFjZ2pyMFN1T
DZwV3JQZDFyTVlGbVVtLTNZZVEzSlZ1TWhkU01VRVdPdzdMOFlBdTF4czAw
cnlEWko2OFQxX1prX3RGSy1FRmI4X2V3NnpMYXpodVBXaDNPc0RXZ9IBlwF
BVV95cUxQQWhFX0djZnR6N3pDRVh5NVFHZXJaLVV1YjB4T3NNSDdWWjNs
dWhUOGt0T2hkSXRSNGRMSjM2TjE0TDAtR2J3Zm5PVEtiZEt6bnR3Z0FwZXVn
QVlhSnc0dlIzenF3TUJ5d1RPQ1JRaFQ3NkF3dHlOYW8ta1RVTVk5ZHltSERvcUhE
MWJjOXB5SUl4dVBma213?oc=5\" target=\"_blank\">Масована атака по Харкову — кількість загиблих зросла tsn.ua
<a
href=\"https://news.google.com/rss/articles/CBMiXkFVX3lxTE9rZEVreEJ6OS1kbV
Nyb3BmYXZmWi1hdnBGYTgzcEVocXlPMmFjRzF1ZnFXQ2hHOFNlbUtTUk9YU
1p0WFdQc1MtSlQ1cEZZSjZDdHIyWGdaeW9xUDVmaWfSAWpBVV95cUxPM1d
CYmMzTkRwT3lKY1FoZEhSeC1LRFRwQU1zYXpEaWZZTUlOck10LXMxMGR
OdFduaTRBc1daUG0yOGprNjVUb3ZkWnRFakE0V3hTaVlkT25IdWF0OExJWEZh
d3FsT0ZydEhR?oc=5\" target=\"_blank\">РФ вдарила авіабомбами по центру Харкова: одна людина загинула, 5 поранені <font

color=\"#6f6f6f\">Українська правда
<a
href=\"https://news.google.com/rss/articles/CBMihwFBVV95cUxNZmNNT3FLQVph
azR0UFJLUVF3SWdrTWJSVGdHMlRqeWVHUEJBaDRRY2hYNTNyWm50LThPa
DAxcVB5VkJMYzhtb2JJY1pydzk0bDlKNnZJcnZNS2lpckJVbkhibU5ULUtyOUFwa
kc3YVpzLTA2bFNySVc3OWctZmszR1h1a0d2Q0XSAYwBQVVfeXFMTlBDWG9
3MEIzWTkzcU9zMHVXenpjVGROdjVZc1U3M2swLXdaRWV0QU41cGZjTWZsd
E91SS10UjhOMk1vTjdPWU11dFktazhodWYzVmZtbVZxNEN6T01qQ3dKS0ZIQV
RWdUJydDFlOFBULURZMms3V2tkc2k0VVNxc2JqRWt2WDJfVElLVGk?oc=5\"
target=\"_blank\">Росія атакувала Харків: є загиблі та
пораненні tsn.ua
<a
href=\"https://news.google.com/rss/articles/CBMiwwFBVV95cUxNT253UGlrVllqc3
ZRMmpnY2EzdDhydTVkVV84QU1hdFltTURocEVneDRLQkN5eHRZUUpHRTN1
RmdKUXd6ZVk5d3NXOXpZSEd2VWpxM08xZWlxZFUySVJ0eEtLbm9Rel9EeUM
zTlVQeVpXNkdyaUEwMXM2OXlTRk91NXAwT2NvMjRrNWtvaGZOcTI2bm1vZi
00Wl8tMlFoTENFSGdVY1V6V3VyanVxcVN5VUpyTmVpbVE0a1dpQTFOX3JW
WTDSAbQBQVVfeXFMTVprMm54aDhqbllNOTlSYUE1NW9DQ3MzdGlNY3RO
Vms0VnR1MWZUcE5uZ2hVQ292b0dEX2VEMDNYMWN4N2lTYWdOWU9NVX
Z4N0cxb0dWUUxhaF8yV0FxSHVUTUg5UE5MOGxESnpZOExaWUxvNDdaSXJU
WmtMd3hjWDd6dW1mQVNrVEVzdHg2aElrcEI4MUV5aU5mSl9YNkVyU1h1RFV
ZbF9SOGRhVXpFTnVDUDJi?oc=5\" target=\"_blank\">Росія знову вдарила по
Харкову КАБами, відомо про загиблого і п’ятьох
поранених nv.ua
<a
href=\"https://news.google.com/rss/articles/CBMie0FVX3lxTFBaSzBORjllYVBfMy1
PUGNBSFVweXA4TS1DY1lBbXZTX3l3eWlOaWd1aEdUMGtjODlQMFJ0eU5TSl
hlWTdLZlpGNWJsZ3RRcm5ReU5UZHpVTUI0Y1dnRHNQODFWOXBZenRhYml
IcmRjeFBCeVJ5Q2NGbTdBMNIBekFVX3lxTFBlSldRekFPVlFlbjEzMk1TOEdQTE
JfdzNPWE1fX2dmNE1EWmFDTXlXeHQ0ejJZREhETjRPU1NteW5vZVRSTjg0QV
FuN1ZDZENkLVVnRWhyb0dsY2pLN3dMMmpDVldWbUpEUmxfb08xQlFkQm9N
M0FWVzZR?oc=5\" target=\"_blank\">У Харкові пролунали вибухи: місто під
ударом КАБів Українські Національні
Новини
",
    "image": null,
    "category": "general",
    "fake_score": 4.74
  },

Додаток Е

Файл .env

```
WHATSAPP_TOKEN=your_token
WHATSAPP_PHONE_NUMBER_ID=your_id
EMAIL_SENDER=myproject@gmail.com
EMAIL_PASSWORD=app_password
DEEPL_API_KEY=your_deepl_key
```