

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему СТВОРЕННЯ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ АГРЕГУВАННЯ
ТА ОПРАЦЮВАННЯ НОВИННИХ ПОТОКІВ
DESIGN OF A SOFTWARE SYSTEM FOR AGGREGATION
AND PROCESSING OF NEWS STREAMS

Виконав(ла): студент(ка) 4 курсу, групи КНТ-212

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

МАКСИМЕНКО О.О.

(ПРІЗВИЩЕ та ініціали)

Керівник ЗАЙКО Т.А.

(ПРІЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2026 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

МАКСИМЕНКА Олександра Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Створення програмної системи для агрегування та опрацювання новинних потоків. Design of a Software System for Aggregation and Processing of News Streams

керівник проєкту (роботи) к.т.н., доцент, ЗАЙКО Тетяна Анатоліївна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 7 ” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 03 червня 2026 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програмної системи. 4. Експлуатація, тестування та експериментальне дослідження програмної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ЗАЙКО Т.А., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “ 7 ” квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програмної системи.	2 тиждень	Розділ 3
5	Розробка програмної системи.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмної системи.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Олександр МАКСИМЕНКО
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Тетяна ЗАЙКО
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
104 с., 4 табл., 30 рис., 3 дод., 20 джерел.

PHP, SQLITE, VISUAL STUDIO CODE, БАЗА ДАНИХ,
ВЕБЗАСТОСУНОК, МОВА ПРОГРАМУВАННЯ, ПРОГРАМНА СИСТЕМА.

Об'єкт дослідження – методи та алгоритми комп'ютерних обчислень
для створення програмних систем агрегації та опрацювання новинних потоків.

Предмет дослідження – програмні системи для агрегації та оброблення
новинних потоків.

Мета роботи – розробка програмної системи для агрегації та
оброблення новинних потоків для зменшення часу дослідження новин.

Матеріали, методи та технічні засоби: мова програмування PHP,
середовище розробки Visual Studio Code, система управління базами даних
SQLite.

Результати. Розроблено проєктні рішення для створення програмної
системи для агрегації та оброблення новинних потоків. Створено програмну
систему для агрегації та оброблення новинних потоків. Проведено
дослідження розробленої програмної системи для агрегації та оброблення
новинних потоків.

Висновки. Мету роботи досягнуто. Розроблено програмну систему для
агрегації та оброблення новинних потоків за допомогою мови програмування
PHP, середовища розробки Visual Studio Code та системи управління базами
даних SQLite.

Галузь використання – оброблення новинної інформації.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 104 pages, 4 tables, 30 figures, 3 appendixes, 20 sources.

PHP, SQLITE, VISUAL STUDIO CODE, DATABASE, WEB APPLICATION, PROGRAMMING LANGUAGE, SOFTWARE SYSTEM.

The object of research is methods and algorithms of computer computations for creating software systems for aggregation and processing of news streams.

The subject of the research is software systems for the aggregation and processing of news streams.

The purpose of this work is to develop the software system for the aggregation and processing of news streams to reduce the time of news research.

Materials, methods and technical tools: PHP programming language, Visual Studio Code development environment, SQLite database management system.

Results. Project solutions for the creation of software system for the aggregation and processing of news streams has been designed. The software system for the aggregation and processing of news streams has been developed. The study of the developed software system for the aggregation and processing of news streams has been conducted.

Conclusions. The goal of the work has been achieved. The software system for the aggregation and processing of news streams using the PHP programming language, the Visual Studio Code development environment, and the SQLite database management system has been developed.

Scope of use – processing of news information.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз предметної області.....	12
1.1 Програмні системи для агрегації та оброблення новинних потоків	12
1.2 Аналіз програмних систем для агрегації та оброблення новинних потоків.....	17
1.3 Висновки за розділом 1	31
2 Матеріали і методи	33
2.1 Вибір мови програмування	33
2.2 Вибір середовища розробки для створення програмної системи для агрегації та оброблення новинних потоків	38
2.3 Вибір засобів зберігання та опрацювання даних.....	43
2.4 Висновки за розділом 2	46
3 Опис програмної системи	47
3.1 Структура програмної системи для агрегації та оброблення новинних потоків.....	47
3.2 Функціонування програмної системи для агрегації та оброблення новинних потоків	50
3.3 Розробка бази даних програмної системи для агрегації та оброблення новинних потоків	54
3.4 Проєктування інтерфейсу програмної системи для агрегації та оброблення новинних потоків	58
3.5 Висновки за розділом 3	60
4 Експлуатація, тестування та експериментальне дослідження програмної системи	62
4.1 Призначення й умови застосування програмної системи	62
4.2 Характеристики програмної системи для агрегації та оброблення новинних потоків	64

4.3 Інструкція по експлуатації програми.....	65
4.3.1 Звернення до програми.....	65
4.3.2 Вхідні й вихідні дані	66
4.3.3 Повідомлення.....	66
4.4 Виконання програмної системи для агрегації та оброблення новинних потоків.....	67
4.5 Тестування програмної системи для агрегації та оброблення новинних потоків.....	74
4.6 Висновки за розділом 4	74
Висновки	75
Перелік джерел посилання	79
Додаток А Технічне завдання	81
Додаток Б Фрагмент тексту програми	86
Додаток В Слайди презентації	97

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

JSON – Java Script Object Notation;

MVC – Model View Controller;

RSS – Rich Site Summary;

БД – база даних;

ІС – інформаційна система;

ПС – програмна система.

ВСТУП

У сучасному інформаційному середовищі спостерігається постійне зростання обсягів цифрового контенту, що поширюється через численні Інтернет ресурси, новинні портали, соціальні мережі та тематичні платформи. Інформаційні повідомлення надходять безперервно та формують складний і динамічний потік даних, який важко ефективно опрацювати без застосування спеціалізованих програмних інструментів. За таких умов особливої ваги набувають програмні системи, призначені для автоматизованого збирання, впорядкування та аналізу новинних повідомлень. Актуальність розроблення подібних рішень пояснюється потребою у швидкому доступі до структурованої, перевіреної та систематизованої інформації, що дозволяє користувачам оперативно реагувати на події та приймати обґрунтовані рішення [1], [2].

Важливість подібних програмних систем зумовлюється також потребою у швидкому опрацюванні великих масивів текстових даних. Сучасні технології дають змогу автоматизувати процедури фільтрації, класифікації та тематичного групування інформаційних повідомлень, що значно підвищує ефективність роботи з новинними потоками. Унаслідок цього зменшується інформаційне перевантаження користувачів, а процес отримання актуальних відомостей стає більш впорядкованим і зручним. Автоматизоване оброблення новин дозволяє виявляти тенденції розвитку подій, аналізувати інформаційний фон та формувати узагальнені уявлення про певні явища або процеси [1], [2].

Особливої значущості системи агрегації новин набувають у контексті стрімкого розвитку цифрових медіа та глобальної мережі Інтернет. Новини публікуються практично миттєво після виникнення подій, тому швидкість отримання інформації стає одним із ключових факторів ефективної діяльності у різних сферах, зокрема у бізнесі, політиці, науці та журналістиці. Програмні засоби, які автоматично збирають і впорядковують новинні повідомлення,

забезпечують можливість своєчасного ознайомлення з важливими подіями та значно скорочують часові витрати на пошук необхідних даних [2]-[4].

Отже, з огляду на стрімке зростання обсягів інформації, різноманітність джерел новин та потребу у швидкому й систематизованому доступі до актуальних даних, розроблення програмної системи для агрегації та оброблення новинних потоків набуває особливої актуальності. Використання таких технологічних рішень сприяє підвищенню ефективності роботи з інформаційними ресурсами, оптимізації процесів пошуку та аналізу новин, а також створює умови для більш результативного використання інформації у різних сферах діяльності [3]-[5].

Проте використання деяких програмних систем для агрегації та оброблення новинних потоків пов'язано ризиком формування інформаційної однобічності. Алгоритми відбору новин можуть орієнтуватися на певні критерії популярності або частоти згадування тем, що призводить до домінування окремих джерел або тематик. У результаті користувач може отримувати обмежений спектр інформації, тоді як менш популярні, але потенційно важливі повідомлення залишаються поза увагою. Така ситуація може впливати на об'єктивність інформаційної картини, що формується у межах системи. Не менш важливим аспектом є технічні та ресурсні витрати, пов'язані з підтримкою функціонування подібних програмних систем. Збирання та оброблення великої кількості новинних повідомлень потребує значних обчислювальних ресурсів, ефективних механізмів зберігання даних і постійного оновлення програмних компонентів. У разі масштабування системи, коли кількість джерел та обсяг інформації постійно зростають, підвищуються вимоги до продуктивності серверної інфраструктури та оптимізації алгоритмів оброблення інформації [4]-[6].

Тому актуальною є розробка програмної системи для агрегації та оброблення новинних потоків. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмної системи для агрегації та оброблення новинних потоків для зменшення часу дослідження новин.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних систем для агрегації та оброблення новинних потоків;
- здійснити проєктування програмної системи для агрегації та оброблення новинних потоків;
- створити програмну систему для агрегації та оброблення новинних потоків;
- дослідити розроблену програмну систему для агрегації та оброблення новинних потоків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмні системи для агрегації та оброблення новинних потоків

Програмна система для агрегації та оброблення новинних потоків є спеціалізованим засобом, призначеним для автоматизованого отримання, накопичення, впорядкування та аналізу інформаційних повідомлень, що надходять із різних мережових джерел. У межах такого програмного рішення забезпечується централізоване збирання новинних матеріалів, публікацій, повідомлень інформаційних порталів, блогів та інших вебресурсів. У результаті формується єдиний інформаційний простір, у якому всі отримані матеріали відображаються у впорядкованому вигляді та можуть бути переглянуті, відсортовані або проаналізовані за певними критеріями. Використання подібних програмних систем дозволяє суттєво спростити процес відстеження великої кількості інформаційних джерел і підвищити ефективність роботи з інформаційними потоками [1]-[3].

Функціонування таких програмних систем базується на автоматизованому отриманні даних із різноманітних Інтернет ресурсів за допомогою спеціальних механізмів підписки або інших технологій обміну інформацією. Після надходження даних відбувається їх оброблення, яке може включати структурування матеріалів, їх тематичне групування, фільтрацію, ранжування за актуальністю або релевантністю, а також формування узагальнених інформаційних стрічок. У межах програмного середовища забезпечується можливість швидкого доступу до нових повідомлень, що дозволяє користувачеві оперативно отримувати відомості про події, які відбуваються у різних інформаційних сферах [3]-[5].

Особливістю таких програмних систем є здатність працювати з великими обсягами інформації, що надходить з різних джерел у режимі постійного оновлення. У процесі функціонування виконується автоматичне відстеження появи нових матеріалів та їх інтеграція у загальний

інформаційний потік. Завдяки цьому забезпечується постійна актуалізація даних і підтримується можливість швидкого реагування на зміни в інформаційному середовищі. Значна увага приділяється механізмам організації даних, що дозволяють структурувати отримані повідомлення відповідно до тематики, джерел або інших характеристик [2], [4].

Сучасні програмні системи для агрегації та оброблення новинних потоків часто доповнюються інструментами інтелектуального аналізу інформації. Такі інструменти можуть застосовувати алгоритми автоматичного визначення тематики публікацій, аналізу ключових слів, виявлення інформаційних тенденцій або узагальнення текстового вмісту. Використання подібних механізмів сприяє зменшенню інформаційного перевантаження та дозволяє зосередитися на найбільш важливих і релевантних матеріалах [4]-[6].

Переваги використання програмних систем цього типу пов'язані насамперед із підвищенням ефективності роботи з інформацією. Завдяки централізованому отриманню новин значно скорочується час, необхідний для пошуку та перегляду матеріалів у мережі. Крім того, автоматизація процесів збору та оброблення даних дозволяє уникнути дублювання інформації, забезпечує її впорядкування та створює зручні умови для подальшого аналізу. Подібні системи також сприяють більш раціональному використанню інформаційних ресурсів, оскільки надають можливість швидко визначати актуальні теми, відстежувати зміни у певних галузях та отримувати узагальнену картину інформаційного простору [3]-[5].

Таким чином, програмні системи для агрегації та оброблення новинних потоків виступають важливим інструментом організації інформаційної діяльності. Їх застосування дозволяє автоматизувати процеси збору, систематизації та аналізу новинних матеріалів, що сприяє більш ефективному використанню інформації у науковій, професійній та повсякденній діяльності [4]-[6].

Процеси агрегації новин розвинулися у складну систему з численними підходами до збору, обробки та представлення інформаційного контенту.

Розуміння різновидів агрегаторів дозволяє користувачам і фахівцям більш ефективно орієнтуватися у насиченому інформаційному середовищі сучасного Інтернету. Програмні системи агрегації відрізняються за принципами роботи, обсягом охоплення та ступенем персоналізації матеріалів, що формує різні рівні користувацького досвіду та задоволення потреб у новинах [1], [3].

Традиційні вебагрегатори новин являють собою базову модель цифрового копіївання новинного контенту. Вони здійснюють збір матеріалів із кількох загальновідомих медіаресурсів і надають їх користувачам через хронологічні стрічки або розділи за категоріями, забезпечуючи широкий спектр охоплення різних тем. Такі агрегатори зручні для швидкого отримання загальної картини інформаційного поля і дозволяють залишатися в курсі подій у багатьох сферах, проте їхній контент часто менш персоналізований і не завжди враховує індивідуальні інтереси окремого користувача [1], [5].

Сучасні спеціалізовані та нішеві платформи агрегації формують інший підхід до обробки новин. Вони зосереджуються на конкретних темах або аудиторіях, забезпечуючи глибоке висвітлення певних напрямків, таких як технології, фінанси, охорона здоров'я чи регіональні події. Для користувачів це означає доступ до ретельно відібраного контенту, адаптованого під їхні потреби, із можливістю застосування розширених алгоритмів персоналізації, які визначають релевантність новин, підбирають матеріали за інтересами та дозволяють формувати більш цілеспрямований інформаційний потік. Завдяки такому підходу користувачі отримують можливість більш ефективно відстежувати важливі теми та аналізувати інформацію, що суттєво підвищує цінність споживання новин у цифрову епоху [1], [6].

Таким чином, різні типи програмних систем агрегації новин забезпечують баланс між широким охопленням інформації та глибокою персоналізацією контенту. Традиційні агрегатори пропонують загальний огляд подій із великого числа джерел, тоді як нішеві платформи дають змогу зосередитися на конкретних темах, забезпечуючи користувачеві більш структурований, цілеспрямований і релевантний досвід споживання

інформації. Взаємодія обох підходів дозволяє створювати комплексні інформаційні стратегії для різних груп користувачів, зберігаючи баланс між швидкістю отримання даних та глибиною їх аналізу [4]-[6].

Поява програмних систем агрегації новин суттєво трансформувала способи отримання та сприйняття інформації в умовах цифрового середовища. У межах сучасного інформаційного простору такі технології змінили традиційні підходи до пошуку новин, їх перегляду та аналізу. Завдяки інтеграції матеріалів з великої кількості джерел у єдиному програмному середовищі забезпечується нова модель взаємодії з інформаційними ресурсами. У результаті новини перестають сприйматися як окремі повідомлення з конкретних медіаорганізацій і починають формуватися у вигляді комплексних інформаційних потоків, що постійно оновлюються та доступні для перегляду в одному інтерфейсі [4]-[6].

Застосування технологій агрегації значною мірою вплинуло на доступність інформації. Цифрові платформи, що реалізують подібні механізми, сприяють усуненню традиційних обмежень, які раніше ускладнювали доступ до новинних матеріалів. Інформація, що походить з різних регіонів та медіаорганізацій, може бути отримана майже миттєво, незалежно від місця перебування користувача. Такий підхід сприяє формуванню більш відкритого інформаційного середовища, у якому значно легше ознайомитися з матеріалами з різних джерел і отримати ширший огляд подій. Одночасний доступ до великої кількості медіаресурсів дозволяє порівнювати різні точки зору, що створює умови для більш критичного сприйняття новинної інформації [4]-[6].

Використання систем агрегації також призвело до змін у способах взаємодії людей з інформаційними матеріалами. У процесі роботи з такими платформами відбувається поступове формування нових моделей поведінки під час споживання новин. Інформація більше не отримується виключно з одного джерела або за допомогою послідовного перегляду окремих медіаресурсів. Натомість формується динамічний потік повідомлень, що

постійно оновлюється та відображає актуальний стан інформаційного простору. Це сприяє швидшому ознайомленню з подіями та дозволяє оперативно реагувати на появу нових повідомлень [4]-[6].

Важливу роль у сучасних програмних системах агрегації відіграють алгоритмічні механізми оброблення інформації. Багато платформ використовують складні методи аналізу даних для формування індивідуалізованих інформаційних середовищ. Під час функціонування таких систем здійснюється аналіз попередніх дій користувача, його інтересів та характеру взаємодії з новинним контентом. На основі отриманих результатів відбувається автоматичний відбір матеріалів, що найбільше відповідають визначеним уподобанням. Унаслідок цього інформаційні стрічки стають більш релевантними та краще адаптованими до потреб конкретної особи [4]-[6].

Завдяки використанню інтелектуальних алгоритмів традиційна модель пасивного читання новин поступово трансформується у більш інтерактивний формат взаємодії з інформацією. Користувач отримує не лише можливість переглядати новини, але й активно впливати на структуру власного інформаційного середовища, змінюючи джерела, тематику або способи представлення матеріалів. У результаті формується гнучка система споживання інформації, що поєднує швидкість отримання новин із можливістю їх глибшого аналізу [4]-[6].

Таким чином, визначено, що програмна система для агрегації та оброблення новинних потоків є спеціалізованим засобом, призначеним для автоматизованого отримання, накопичення, впорядкування та аналізу інформаційних повідомлень, що надходять із різних мережевих джерел. У межах такого програмного рішення забезпечується централізоване збирання новинних матеріалів, публікацій, повідомлень інформаційних порталів, блогів та інших вебресурсів. У результаті формується єдиний інформаційний простір, у якому всі отримані матеріали відображаються у впорядкованому вигляді та можуть бути переглянуті, відсортовані або проаналізовані за певними критеріями.

1.2 Аналіз програмних систем для агрегації та оброблення новинних потоків

Проаналізуємо програмні системи для агрегації та оброблення новинних потоків [7]-[14].

Програмна система Feedspot (рис. 1.1) призначена для централізованого отримання та впорядкування інформаційних матеріалів, що надходять із різноманітних Інтернет джерел. Основна ідея функціонування цього сервісу полягає у створенні єдиного інформаційного середовища, у якому користувачеві надається можливість ознайомлюватися з публікаціями з різних платформ без необхідності відвідувати кожен ресурс окремо. У межах системи здійснюється інтеграція великої кількості каналів розповсюдження інформації, що дозволяє організувати зручний доступ до новин, блогових матеріалів, відеоконтенту, аудіоподкастів та інших типів цифрових публікацій [7], [12].

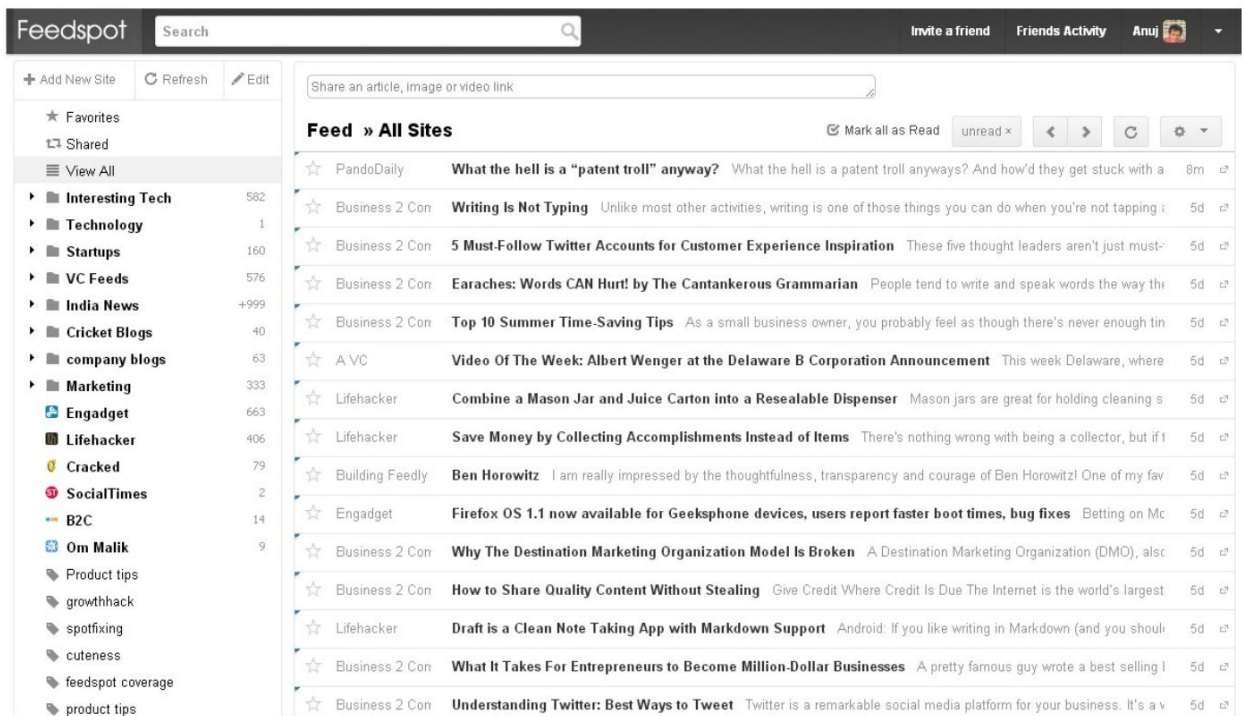


Рисунок 1.1 – Програмна система Feedspot [12]

Функціональні можливості програмної системи Feedspot орієнтовані на агрегування контенту та його подальше відображення у вигляді впорядкованого інформаційного потоку. У межах такого підходу різноманітні джерела даних поєднуються у спільному інтерфейсі, що спрощує процес ознайомлення з новими матеріалами. Користувачеві не потрібно здійснювати пошук оновлень на численних вебсайтах, оскільки відповідна інформація автоматично збирається та подається у вигляді єдиної стрічки [7], [12].

Важливою складовою функціональності цієї системи є наявність вбудованого механізму пошуку джерел новинної інформації. За допомогою такого інструменту виконується знаходження RSS-каналів, що відповідають визначеним тематичним запитам. Після виявлення відповідних інформаційних потоків вони можуть бути додані до особистого списку підписок, що дозволяє отримувати регулярні оновлення без необхідності повторного пошуку. Окрім цього, у програмному середовищі реалізовано механізми рекомендацій, які аналізують уже обрані джерела та пропонують інші канали, тематично близькі до тих, що вже використовуються. Завдяки такому підходу забезпечується поступове розширення інформаційного простору користувача та підвищується ймовірність отримання релевантних матеріалів [7], [12].

Для розширення охоплення інформаційних джерел система інтегрується із популярними сервісами соціальних мереж, зокрема з платформами Twitter, Facebook, Tumblr, Pinterest та Instagram. Завдяки такій взаємодії забезпечується автоматичне отримання повідомлень про нові публікації з облікових записів і каналів, на які здійснено підписку. Це дозволяє відстежувати активність різних інформаційних джерел у режимі єдиного інформаційного простору та оперативно отримувати повідомлення про появу нових матеріалів [7], [12].

У результаті поєднання механізмів агрегації RSS-стрічок, інтелектуального пошуку джерел і інтеграції з соціальними платформами формується комплексне середовище для моніторингу інформаційних потоків. Такий підхід дозволяє значно спростити процес отримання новин,

систематизувати інформацію та забезпечити зручний доступ до актуального контенту з великої кількості різноманітних джерел [7], [12].

Основними характеристиками програмного забезпечення Feedspot є такі [7], [12]:

- централізоване агрегування контенту: програмна система Feedspot забезпечує об'єднання матеріалів із різноманітних інформаційних джерел, що дозволяє отримувати новини, публікації блогів, відео та інші типи цифрового контенту в межах єдиного інтерфейсу без необхідності переходу між окремими вебресурсами [7], [12];

- підтримка RSS-каналів: програмна система Feedspot надає можливість підписуватися на інформаційні стрічки та автоматично отримувати оновлення з вебсайтів, що використовують технологію RSS для розповсюдження новинного контенту [7], [12];

- інтеграція із соціальними платформами: програмна система Feedspot забезпечує взаємодію з популярними соціальними мережами, що дозволяє відстежувати нові публікації з підписаних облікових записів і отримувати їх у загальному інформаційному потоці [7], [12];

- вбудований пошук джерел: у програмній системі Feedspot реалізовано механізм пошуку RSS-каналів, завдяки якому користувачеві надається можливість знаходити нові інформаційні джерела за його тематичними запитами [7], [12];

- рекомендації інформаційних ресурсів: програмна система Feedspot здатна аналізувати вже додані канали та пропонувати інші джерела, зміст яких є тематично близьким до інтересів користувача [7], [12];

- організація інформаційних потоків: функціональність програмної системи Feedspot передбачає структурування отриманих новин у вигляді стрічок або тематичних категорій, що спрощує їх перегляд і навігацію між матеріалами [7], [12];

- персоналізація інформаційного середовища: користувачеві програмної системи Feedspot надаються інструменти для формування власного

набору підписок, що дозволяє налаштовувати перелік джерел відповідно до індивідуальних інформаційних потреб [7], [12];

- доступ через вебінтерфейс: взаємодія з функціональністю програмної системи Feedspot здійснюється за допомогою браузера, що забезпечує можливість використання сервісу без встановлення спеціалізованого програмного забезпечення [7], [12];

- підтримка різних типів медіаконтенту: у межах програмної системи Feedspot можуть відображатися не лише текстові новини, а й мультимедійні матеріали, включаючи відео та подкасти [7], [12];

- автоматичне оновлення інформації: система періодично перевіряє підключені джерела та додає нові публікації до інформаційного потоку без необхідності виконання ручних дій користувачем [7], [12].

Серед недоліків програмної системи Feedspot можна відзначити обмеження функціональних можливостей у безкоштовному режимі використання, залежність ефективності роботи від доступності зовнішніх джерел інформації, а також певні обмеження щодо гнучкості налаштування відображення контенту порівняно з деякими іншими програмними засобами для агрегації новинних потоків [7], [12].

Програмна система Feedly (рис. 1.2) належить до сучасних інструментів, призначених для централізованого отримання, впорядкування та аналізу інформаційних матеріалів, що надходять із великої кількості вебресурсів. Основне призначення цього програмного засобу полягає у забезпеченні зручного доступу до новин, блогових публікацій та інших інформаційних потоків у межах єдиного середовища. Завдяки такому підходу відпадає необхідність окремого відвідування численних сайтів, оскільки всі оновлення збираються автоматично та подаються у вигляді впорядкованої інформаційної стрічки [9], [10], [13].

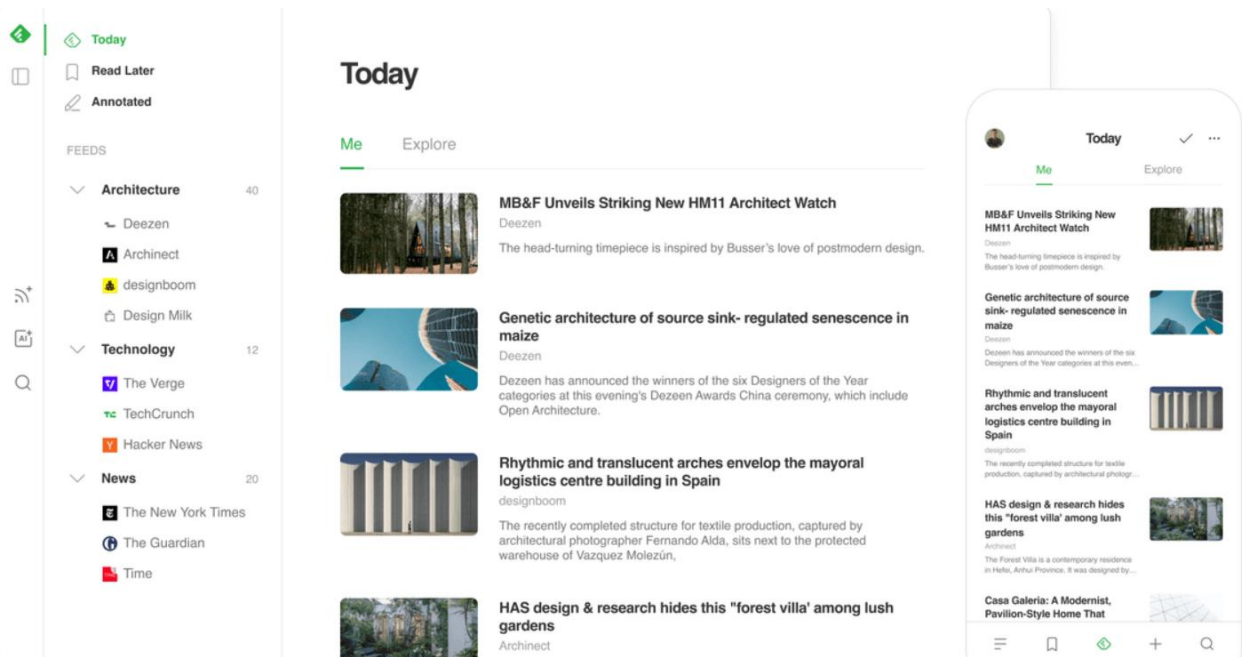


Рисунок 1.2 – Програмна система Feedly [13]

Інтерфейс платформи Feedly розроблено таким чином, щоб забезпечити комфортне сприйняття інформації та підвищити ефективність взаємодії з контентом. У процесі перегляду новин значна увага приділяється візуальному представленню матеріалів, що супроводжуються зображеннями та іншими графічними елементами. Такий підхід сприяє більш зручному ознайомленню з інформацією та покращує загальний досвід роботи із системою. Функціональні можливості сервісу не обмежуються лише обробленням RSS-стрічок, оскільки платформа здатна відстежувати оновлення з інших типів інформаційних ресурсів, зокрема відеоканалів, тематичних вебпублікацій і блогів [9], [10], [13].

З часом цей програмний продукт трансформувався з простого засобу перегляду новин у складну платформу для роботи з інформацією, що активно використовується не лише окремими користувачами, але й робочими групами та організаціями. У межах такого підходу система може виконувати роль інструмента для аналізу інформаційних потоків, дослідження тематичних матеріалів та колективної роботи з даними. Значною перевагою платформи є можливість упорядковувати великі обсяги інформації у спеціальних робочих

просторах, які використовуються для збереження, структурування та подальшого спільного опрацювання знайдених матеріалів [9], [10], [13].

У системі реалізовано інтелектуальні механізми оброблення інформації, що ґрунтуються на застосуванні технологій штучного інтелекту. У межах платформи функціонує спеціальний цифровий помічник, здатний аналізувати вміст публікацій, виконувати їх узагальнення, визначати тематичну належність матеріалів та відсіювати інформацію, яка не відповідає заданим критеріям. Завдяки цьому значно зменшується інформаційне перевантаження та підвищується ефективність роботи з великими масивами даних. Додатково реалізовано механізми пошуку, що дозволяють знаходити матеріали не лише серед безпосередньо підключених джерел, але й у ширшому інформаційному просторі, охоплюючи велику кількість зовнішніх ресурсів [9], [10], [13].

Програмна система Feedly також передбачає можливість інтеграції з різноманітними інструментами для командної роботи та обміну інформацією. Завдяки цьому отримані матеріали можуть передаватися до інших програмних середовищ, які використовуються в організаціях для комунікації, планування завдань або керування проєктами. Такий підхід сприяє ефективнішій взаємодії між членами команди та дозволяє швидко поширювати важливі інформаційні матеріали у межах робочого процесу. Крім того, система підтримує підключення зовнішніх сервісів автоматизації, що дозволяє організувати складні сценарії оброблення інформації та інтеграцію з різними програмними платформами [9], [10], [13].

Особлива увага приділяється питанням інформаційної безпеки та захисту даних. У межах програмної системи реалізовано механізми автентифікації користувачів, що дозволяють організувати централізований доступ до сервісу в корпоративному середовищі. Передбачено використання сучасних методів шифрування для захисту даних під час їх передавання та зберігання, що сприяє підвищенню рівня безпеки під час роботи з інформаційними ресурсами [9], [10], [13].

З точки зору архітектури програмний засіб функціонує як хмарна платформа, доступ до якої здійснюється через вебінтерфейс або спеціалізовані мобільні застосунки. Такий підхід забезпечує можливість роботи з інформаційними потоками незалежно від типу використовуваного пристрою та місця перебування користувача. Усі операції з оброблення новин виконуються у віддаленому серверному середовищі, що дозволяє підтримувати стабільну роботу сервісу та спрощує процес оновлення програмного забезпечення [9], [10], [13].

Програмна система Feedly також характеризується розвиненою екосистемою та активною спільнотою користувачів. Для роботи із сервісом доступна детальна документація, що описує основні можливості та способи інтеграції з іншими програмними системами. Окрім цього, для корпоративних користувачів передбачено спеціалізовані канали технічної підтримки, які дозволяють оперативно вирішувати питання, пов'язані з налаштуванням та використанням програмного середовища [9], [10], [13].

Разом із тим варто зазначити, що деякі розширені функціональні можливості платформи доступні лише у межах платних тарифних планів. Це може створювати певні обмеження для користувачів, які потребують лише базових функцій перегляду новин. Також значна кількість інструментів та параметрів налаштування інколи може ускладнювати освоєння системи для тих, хто використовує подібні сервіси виключно для простого читання новинних матеріалів. Проте загалом програмна система Feedly розглядається як потужне середовище для роботи з інформаційними потоками, яке поєднує функції агрегації, аналізу та організації великої кількості інформаційних ресурсів у єдиній платформі [9], [10], [13].

Програмна система Feedly має такі особливості [9], [10], [13]:

- централізоване агрегування контенту: програмна система Feedly забезпечує збирання новин, блогових публікацій та інших інформаційних матеріалів із великої кількості вебресурсів і відображає їх у межах єдиного інтерфейсу для зручного перегляду користувачем [9], [10], [13];

- підтримка RSS-каналів: програмна система Feedly надає можливість підписуватися на інформаційні стрічки різних вебсайтів та автоматично отримувати нові публікації без необхідності самостійного відвідування кожного ресурсу [9], [10], [13];

- інтелектуальна обробка інформації: у програмній системі Feedly використовуються алгоритми штучного інтелекту для аналізу новинних матеріалів, їх узагальнення та визначення релевантності відповідно до інтересів користувача [9], [10], [13];

- автоматична фільтрація контенту: у програмній системі Feedly реалізовано механізми відбору інформації, що дозволяють відсіювати другорядні або нерелевантні матеріали та зменшувати обсяг зайвих даних у новинному потоці [9], [10], [13];

- розширений пошук інформаційних джерел: програмна система Feedly забезпечує можливість знаходження новинних матеріалів і тематичних ресурсів не лише серед підключених підписок, але й у широкому інформаційному просторі Інтернету [9], [10], [13];

- організація матеріалів у тематичні колекції: система Feedly дозволяє структурувати знайдені публікації у спеціальних робочих областях, що полегшує подальший аналіз та спільне використання інформації [9], [10], [13];

- підтримка командної роботи: у програмній системі Feedly передбачено можливість спільного доступу до зібраних матеріалів та обміну інформацією між учасниками робочих груп [9], [10], [13];

- інтеграція з іншими сервісами: програмна система Feedly підтримує взаємодію з різними інструментами комунікації, системами керування проектами та сервісами автоматизації для обміну інформацією та організації робочих процесів [9], [10], [13];

- хмарна архітектура: функціонування системи реалізовано у вигляді онлайн-сервісу, що дозволяє отримувати доступ до інформаційних потоків через веббраузер або мобільні застосунки без встановлення локального програмного забезпечення [9], [10], [13];

– персоналізація інформаційного середовища: користувачеві програмної системи Feedly надаються інструменти для формування власних інформаційних стрічок, налаштування джерел новин та управління параметрами відображення контенту [9], [10], [13].

Серед недоліків програмної системи Feedly можна відзначити обмеженість функціональних можливостей у безкоштовному режимі використання, необхідність оформлення платної підписки для доступу до розширених інструментів аналітики та автоматизації, а також певну складність інтерфейсу для нових користувачів через велику кількість доступних функцій і параметрів налаштування [9], [10], [13].

Програмна система RSSOwl (рис. 1.3) призначена для отримання, структурування та перегляду інформаційних матеріалів, що надходять із різних Інтернет ресурсів у вигляді новинних стрічок [9], [14].

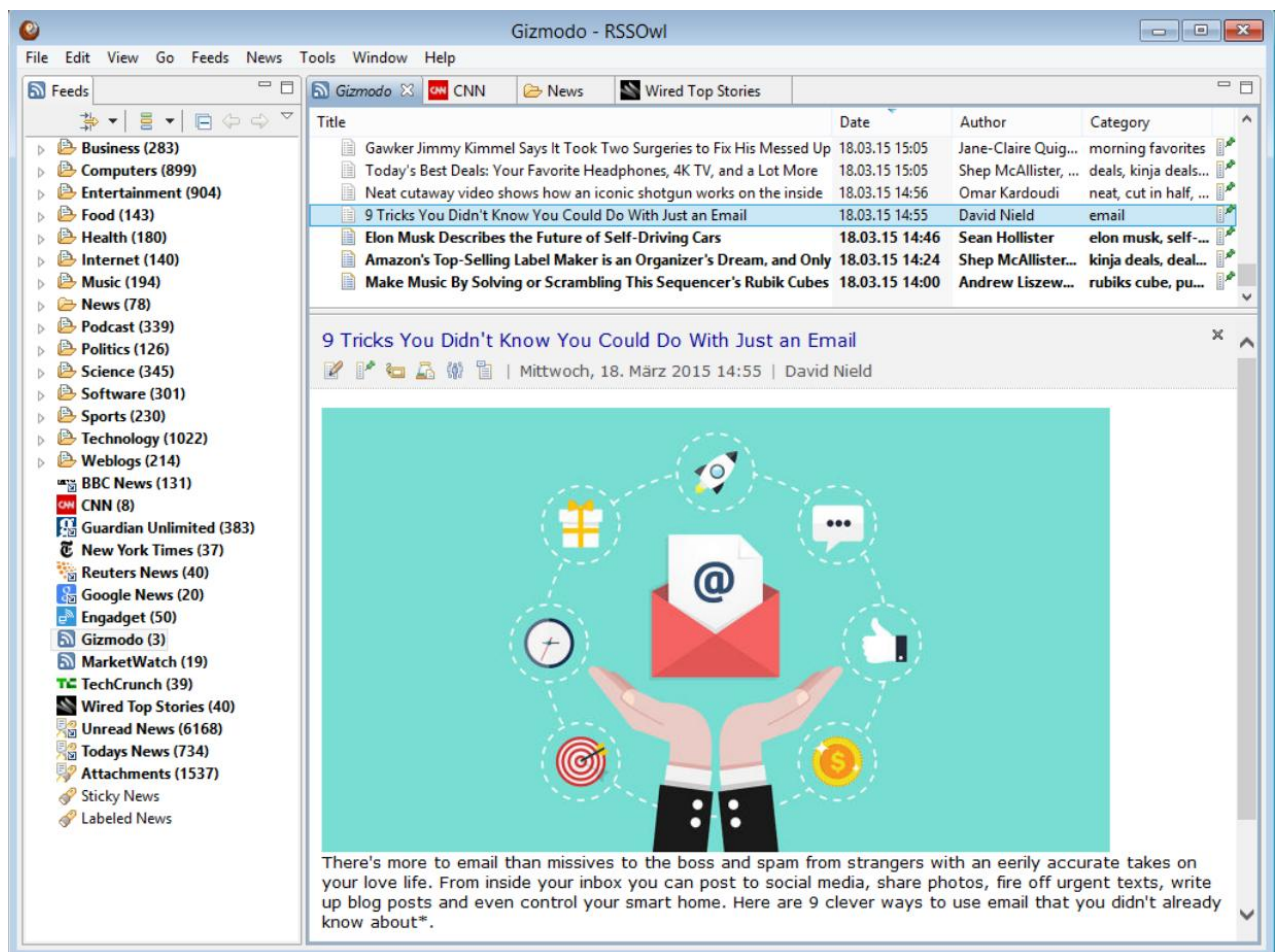


Рисунок 1.3 – Програмна система RSSOwl [14]

Програмна система RSSOwl орієнтована на автоматизацію процесів, пов'язаних із регулярним відстеженням оновлень вебсайтів, блогів та інформаційних порталів. Завдяки використанню програмної системи RSSOwl забезпечується централізоване отримання новинних повідомлень і публікацій, що дозволяє значно спростити роботу з великою кількістю джерел інформації. Функціонування системи ґрунтується на принципі автоматичного отримання оновлень із підключених інформаційних каналів. Усі нові матеріали завантажуються до внутрішнього середовища програми, де вони відображаються у впорядкованому вигляді. Такий підхід дозволяє уникнути необхідності самостійно відвідувати кожен окремий вебресурс для перевірки появи нових публікацій. У результаті користувач отримує доступ до узагальненої інформаційної стрічки, яка формується на основі вибраних джерел [9], [14].

Важливою особливістю програмної системи RSSOwl є наявність механізмів, що дозволяють автоматизувати типові операції з новинними матеріалами. У межах програмного середовища передбачено можливість виконання різних дій із отриманими повідомленнями, зокрема їх упорядкування, маркування та подальше збереження для майбутнього перегляду. Це дозволяє більш ефективно організовувати інформаційні потоки та швидко знаходити потрібні матеріали серед великої кількості новинних повідомлень [9], [14].

У системі RSSOwl реалізовано інструменти пошуку, що дають змогу оперативно знаходити інформацію серед зібраних новинних матеріалів. Механізми пошукового аналізу дозволяють виконувати миттєве опрацювання наявного контенту та відображати результати відповідно до введених критеріїв. При цьому знайдені результати можуть зберігатися як окремі інформаційні потоки, що формуються автоматично на основі заданих параметрів пошуку. Такий підхід значно спрощує відстеження публікацій, що відповідають певній тематиці або ключовим словам [9], [14].

Додатково у програмному середовищі RSSOwl передбачено інструменти організації та класифікації інформаційних матеріалів. Отримані новини можуть супроводжуватися спеціальними позначками, що дозволяє групувати їх за тематичними ознаками або за рівнем важливості. Окрім цього, система підтримує механізми сповіщень, які інформують про появу нових повідомлень у підключених джерелах. Завдяки цьому користувач має можливість своєчасно отримувати інформацію про актуальні події та оперативно реагувати на нові публікації [9], [14].

Ще однією характерною рисою програмної системи RSSOwl є можливість впорядкування великої кількості інформаційних джерел у межах єдиного робочого середовища. Користувач може створювати власну структуру підписок, групувати новинні стрічки за темами та організовувати їх у зручному порядку [9], [14].

З технічної точки зору програмна система RSSOwl реалізована як багатоплатформне рішення, що забезпечує можливість використання на різних операційних системах. Це дозволяє запускати програму на різних типах комп'ютерних пристроїв без суттєвих обмежень, що розширює сферу її застосування. Усі основні функції, пов'язані з отриманням, переглядом та організацією новинних повідомлень, виконуються у межах локального програмного середовища, що забезпечує стабільність роботи та незалежність від зовнішніх сервісів [9], [14].

Таким чином, програмна система RSSOwl є ефективним інструментом для централізованого керування інформаційними потоками, який поєднує можливості автоматичного отримання новин, пошуку та впорядкування матеріалів, а також забезпечує зручні механізми організації великої кількості джерел інформації в одному програмному середовищі.

Програмна система RSSOwl має такі особливості [9], [14]:

- підтримка RSS-стрічок: програмна система RSSOwl забезпечує отримання та відображення новинних повідомлень із різних вебресурсів у

форматі RSS, що дозволяє централізовано переглядати інформаційні матеріали в одному програмному середовищі [9], [14];

- автоматичне оновлення новин: у системі RSSOwl реалізовано механізми регулярної перевірки підключених джерел, завдяки чому нові публікації автоматично додаються до стрічки новин без необхідності ручного оновлення [9], [14];

- швидкий пошук інформації: програмна система RSSOwl підтримує інструменти миттєвого пошуку серед отриманих матеріалів, що дозволяє користувачам оперативно знаходити новини за ключовими словами або тематикою [9], [14];

- збереження результатів пошуку: знайдені матеріали можуть бути збережені у вигляді окремих інформаційних потоків, що формуються на основі заданих параметрів і автоматично оновлюються при появі нових відповідних публікацій [9], [14];

- система міток та категорій: програмна система RSSOwl дозволяє користувачам позначати новинні повідомлення спеціальними тегами та групувати їх відповідно до тематики або інших критеріїв організації інформації [9], [14];

- підтримка сповіщень: користувач системи RSSOwl може отримувати повідомлення про появу нових матеріалів у підключених джерелах, що дозволяє своєчасно відстежувати важливі інформаційні події [9], [14];

- організація новин у колекції: програмна система RSSOwl надає можливість впорядковувати інформаційні стрічки у тематичні групи, що спрощує роботу з великою кількістю джерел новин [9], [14];

- локальне збереження даних: усі новинні матеріали та параметри підписок можуть зберігатися у локальному середовищі користувача, що забезпечує автономність роботи програмного забезпечення [9], [14];

- багатоплатформна архітектура: програмна система RSSOwl розроблена таким чином, щоб забезпечити можливість використання на різних операційних системах та комп'ютерних платформах [9], [14];

– автоматизація роботи з новинами: програмна система RSSOwl підтримує виконання типових операцій із новинними повідомленнями, зокрема їх сортування, маркування та впорядкування у зручній для користувача структурі [9], [14].

Серед недоліків програмної системи RSSOwl можна відзначити обмежений розвиток програмного продукту у сучасних умовах, відсутність активної підтримки та регулярних оновлень, а також менш сучасний користувацький інтерфейс у порівнянні з новітніми хмарними сервісами для агрегації новин. Крім того, використання локального програмного середовища може створювати певні обмеження щодо синхронізації інформації між різними пристроями користувача [9], [14].

Порівняльну характеристику програмних систем для агрегації та оброблення новинних потоків наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмних систем для агрегації та оброблення новинних потоків

Критерій порівняння	Feedspot [12]	Feedly [13]	RSSOwl [14]
агрегування великої кількості джерел	+	+	+—
інтелектуальна фільтрація контенту	+—	+	—
автоматичне відстеження нових джерел	+	+—	—
розширений пошук інформації	+	+	+—
підтримка командної роботи	+—	+	—
інтеграція з іншими сервісами	+—	+	+—
персоналізація стрічок	+—	+	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних систем для агрегації та оброблення новинних потоків. Проте використання деяких програмних систем для агрегації та оброблення новинних потоків пов'язано ризиком формування інформаційної однобічності. Алгоритми відбору новин можуть орієнтуватися на певні критерії популярності або частоти згадування тем, що призводить до домінування окремих джерел або тематик. У результаті користувач може отримувати обмежений спектр інформації, тоді як менш популярні, але потенційно важливі повідомлення залишаються поза увагою. Така ситуація може впливати на об'єктивність інформаційної картини, що формується у межах системи. Не менш важливим аспектом є технічні та ресурсні витрати, пов'язані з підтримкою функціонування подібних програмних систем. Збирання та оброблення великої кількості новинних повідомлень потребує значних обчислювальних ресурсів, ефективних механізмів зберігання даних і постійного оновлення програмних компонентів. У разі масштабування системи, коли кількість джерел та обсяг інформації постійно зростають, підвищуються вимоги до продуктивності серверної інфраструктури та оптимізації алгоритмів оброблення інформації. Тому актуальною є розробка програмної системи для агрегації та оброблення новинних потоків.

При розробці програмної системи для агрегації та оброблення новинних потоків необхідно забезпечити такі функціональні вимоги:

- підтримка можливості отримання новин із зовнішніх інформаційних ресурсів на основі новинних стрічок;
- можливість відображення новин у зручному для користувача вигляді через вебінтерфейс;
- можливість реєстрації та авторизації користувачів у програмній системі;
- підтримка можливості створення та керування підписками на новинні стрічки;
- підтримка можливості додавання нових підписок на новинні ресурси;

- підтримка можливості перегляду новин, що ще не були прочитані користувачем;
- можливість налаштування параметрів відображення та зберігання новин у системі;
- підтримка можливості автоматичного оновлення новинних стрічок.

1.3 Висновки за розділом 1

Визначено, що програмна система для агрегації та оброблення новинних потоків є спеціалізованим засобом, призначеним для автоматизованого отримання, накопичення, впорядкування та аналізу інформаційних повідомлень, що надходять із різних мережових джерел. У межах такого програмного рішення забезпечується централізоване збирання новинних матеріалів, публікацій, повідомлень інформаційних порталів, блогів та інших вебресурсів. У результаті формується єдиний інформаційний простір, у якому всі отримані матеріали відображаються у впорядкованому вигляді та можуть бути переглянуті, відсортовані або проаналізовані за певними критеріями. Використання подібних програмних систем дозволяє суттєво спростити процес відстеження великої кількості інформаційних джерел і підвищити ефективність роботи з інформаційними потоками.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних систем для агрегації та оброблення новинних потоків. Проте використання деяких програмних систем для агрегації та оброблення новинних потоків пов'язано ризиком формування інформаційної однобічності. Алгоритми відбору новин можуть орієнтуватися на певні критерії популярності або частоти згадування тем, що призводить до домінування окремих джерел або тематик. У результаті користувач може отримувати обмежений спектр інформації, тоді як менш популярні, але потенційно важливі повідомлення залишаються поза увагою. Така ситуація може впливати на об'єктивність інформаційної картини, що формується у

межах системи. Не менш важливим аспектом є технічні та ресурсні витрати, пов'язані з підтримкою функціонування подібних програмних систем. Збирання та оброблення великої кількості новинних повідомлень потребує значних обчислювальних ресурсів, ефективних механізмів зберігання даних і постійного оновлення програмних компонентів. У разі масштабування системи, коли кількість джерел та обсяг інформації постійно зростають, підвищуються вимоги до продуктивності серверної інфраструктури та оптимізації алгоритмів оброблення інформації. Тому актуальною є розробка програмної системи для агрегації та оброблення новинних потоків.

Сформульовано функціональні вимоги до програмної системи для агрегації та оброблення новинних потоків.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмної системи для агрегації та оброблення новинних потоків було використано мову програмування PHP [15], [16].

Під час проєктування програмних систем, призначених для збирання, об'єднання та подальшого аналізу інформаційних повідомлень із різноманітних мережевих джерел, особливого значення набуває вибір технологічної платформи, яка забезпечує стабільність функціонування, ефективність оброблення даних і зручність подальшої підтримки програмного продукту. У цьому контексті доцільність використання мови програмування PHP пояснюється її широким застосуванням у сфері веброзроблення та наявністю значної кількості інструментів, що полегшують створення серверних застосунків, здатних взаємодіяти з великою кількістю зовнішніх інформаційних ресурсів [15], [16].

Мова програмування PHP займає помітне місце у сфері створення веборієнтованих програмних продуктів та протягом тривалого часу залишається однією з найбільш поширених технологій серверної розробки. Її розвиток розпочався як інструмент для створення динамічних вебсторінок, проте з часом функціональні можливості суттєво розширилися, що дозволило використовувати її для побудови повноцінних інформаційних систем різного рівня складності. Сучасні версії цієї мови підтримують об'єктно-орієнтовану парадигму програмування, що відкриває можливість створення складних програмних архітектур, придатних для реалізації масштабних вебпроєктів та корпоративних застосунків [15], [16].

Важливою характеристикою PHP є тісна інтеграція з технологіями вебсередовища. Програмний код виконується на стороні сервера, після чого результати оброблення передаються у вигляді сформованих вебсторінок або структурованих даних, які відображаються у браузері користувача. Такий підхід дозволяє створювати інтерактивні вебресурси, здатні реагувати на

запити користувачів, обробляти введену інформацію, взаємодіяти з базами даних та виконувати різноманітні обчислювальні операції. Завдяки цьому PHP активно застосовується під час розроблення інформаційних порталів, систем електронної комерції, корпоративних вебплатформ і численних інтернет-сервісів [15], [16].

Суттєвим фактором популярності цієї технології є її відкритість і доступність. PHP поширюється як програмне забезпечення з відкритим вихідним кодом, що забезпечує можливість вільного використання, модифікації та інтеграції з іншими програмними компонентами. Такий підхід сприяв формуванню широкої екосистеми інструментів, бібліотек і допоміжних програмних модулів, які значно розширюють функціональні можливості мови. У межах цієї екосистеми створено численні фреймворки, системи керування вмістом, бібліотеки для роботи з мережевими протоколами, інструменти безпеки та засоби автоматизації розроблення [15], [16].

Окрему увагу привертає здатність PHP ефективно взаємодіяти з різними операційними системами та серверними платформами. Виконання програмного коду можливе як у середовищі Linux, так і на платформах Windows або macOS, що забезпечує гнучкість у виборі інфраструктури для розгортання програмних систем. Завдяки цьому програмні рішення, створені за допомогою цієї мови, можуть бути інтегровані у різні серверні середовища без суттєвих змін у програмному коді [15], [16].

Сучасні версії PHP також характеризуються підвищеним рівнем продуктивності та оптимізації. Значна увага приділяється вдосконаленню механізмів виконання програмного коду, управлінню пам'яттю та оптимізації оброблення запитів. Це дозволяє створювати системи, здатні обслуговувати значну кількість користувачів і виконувати оброблення великих обсягів інформації у режимі реального часу. У результаті мова програмування PHP залишається актуальним інструментом для створення високонавантажених вебресурсів [15], [16].

Не менш важливим аспектом є наявність розвинених засобів забезпечення безпеки програмних систем. У межах цієї технології реалізовано механізми захисту від типових загроз, пов'язаних із вебзастосунками, зокрема контролю доступу до даних, перевірки коректності введеної інформації, захисту від ін'єкційних атак та інших вразливостей. Використання відповідних підходів до розроблення дозволяє створювати програмні рішення з високим рівнем надійності та захищеності [15], [16].

Значну роль у поширенні PHP відіграє також активна спільнота розробників, яка постійно вдосконалює інструменти програмування, розробляє нові бібліотеки та ділиться практичним досвідом використання цієї технології. Завдяки цьому формуються численні навчальні матеріали, приклади реалізації програмних рішень та рекомендації щодо оптимізації програмного коду. Така підтримка сприяє підвищенню якості програмних продуктів і полегшує процес освоєння мови новими розробниками [15], [16].

Отже, PHP є гнучкою та функціонально насиченою мовою програмування, яка поєднує можливості створення динамічних вебзастосунків, ефективну взаємодію з базами даних, високу сумісність із різними серверними середовищами та широкі можливості для розширення функціональності. Сукупність цих характеристик забезпечує її тривалу популярність у сфері веброботизації та робить цю технологію придатною для реалізації різноманітних програмних систем, орієнтованих на оброблення та представлення інформації в мережевому середовищі [15], [16].

Однією з важливих передумов використання цієї мови є її орієнтація на створення веборієнтованих програмних систем. Значна частина новинної інформації розміщується на Інтернет ресурсах, тому програмне забезпечення, яке виконує функції агрегації, повинно ефективно працювати у середовищі мережових протоколів та взаємодіяти з вебсервером. У межах цієї архітектури мова програмування PHP виступає як зручний інструмент для реалізації серверної логіки, що забезпечує оброблення запитів, отримання даних із

зовнішніх джерел, їх структуризацію та подальше передавання до клієнтської частини системи [15], [16].

Важливою характеристикою цієї технології є наявність великої кількості готових бібліотек та розширень, які значно спрощують реалізацію механізмів отримання інформації з мережі. У середовищі PHP передбачено інструменти для роботи з HTTP-запитами, оброблення форматів XML і JSON, а також взаємодії з RSS-каналами, що широко використовуються для поширення новинного контенту. Завдяки таким можливостям створюється сприятливе середовище для реалізації функцій автоматичного збирання інформації з різних джерел та її подальшого впорядкування [15], [16].

Суттєвим фактором також є висока сумісність PHP із більшістю популярних систем керування базами даних. Під час розроблення системи агрегації новин виникає необхідність зберігання значних обсягів текстової інформації, метаданих, часових позначок та інших характеристик повідомлень. У межах цієї технології передбачено ефективні механізми взаємодії з базами даних, що дозволяє організувати надійне зберігання інформації та забезпечити швидкий доступ до неї під час виконання пошукових або аналітичних операцій [15], [16].

Варто також враховувати, що PHP характеризується достатньо простою структурою синтаксису та зрозумілою логікою побудови програмного коду. Це створює сприятливі умови для швидкої реалізації функціональних компонентів системи та подальшого розширення її можливостей. У процесі розроблення програмного забезпечення для роботи з новинними потоками важливим є можливість оперативного внесення змін до алгоритмів збирання, фільтрації або класифікації інформації. Простота інтеграції нових модулів і можливість швидкого модифікування програмного коду сприяють гнучкості програмної архітектури [15], [16].

Додатковою перевагою є значна поширеність PHP у середовищі вебхостингу. Більшість серверних платформ підтримує виконання PHP-застосунків без необхідності складної конфігурації або встановлення

додаткового програмного забезпечення. Це спрощує процес розгортання системи агрегації новин та знижує витрати на її технічну підтримку. У результаті програмний продукт може бути швидко інтегрований у наявну інфраструктуру вебресурсів [15], [16].

Не менш важливим аспектом є активний розвиток екосистеми цієї мови програмування. Існування великої спільноти розробників, значної кількості фреймворків, інструментів тестування та бібліотек створює умови для використання сучасних підходів до створення програмного забезпечення. Це дозволяє підвищити якість програмного коду, забезпечити стабільність роботи системи та полегшити її подальшу модернізацію [15], [16].

Обґрунтування вибору мови програмування для розробки програмної системи для агрегації та оброблення новинних потоків наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмної системи для агрегації та оброблення новинних потоків

Критерій порівняння мов програмування	Мова програмування		
	PHP	Python	C#
Орієнтація на веброзробку	+	+-	+-
Простота розгортання на вебсервері	+	+-	-
Наявність вбудованих засобів роботи з HTTP та вебзапитами	+	+-	+-
Простота інтеграції з вебсторінками	+	+-	-
Зручність роботи з базами даних	+	+	+-
Швидкість створення веборієнтованих систем	+	+	-

Отже, для реалізації програмної системи для агрегації та оброблення новинних потоків обрано мову програмування PHP, яка забезпечує ефективну інтеграцію з вебтехнологіями, наявність засобів для отримання інформації з мережевих джерел, зручну взаємодію з базами даних, простоту реалізації серверної логіки та широке поширення у середовищі вебінфраструктури. Сукупність цих характеристик мови PHP створює сприятливі умови для розроблення функціональної, масштабованої та зручної у підтримці програмної системи, здатної ефективно працювати з великими потоками новинної інформації.

2.2 Вибір середовища розробки для створення програмної системи для агрегації та оброблення новинних потоків

В якості середовища розробки для створення програмної системи для агрегації та оброблення новинних потоків було обрано середовище Visual Studio Code [17], [18].

Під час створення програмної системи, призначеної для агрегації та подальшого опрацювання новинних потоків, важливе значення має вибір середовища розробки, яке забезпечує зручність написання програмного коду, ефективне тестування функціональних модулів та можливість швидкого внесення змін до програмної логіки. У контексті розроблення веборієнтованих застосунків мовою PHP доцільним рішенням може вважатися використання середовища Visual Studio Code. Це програмне забезпечення набуло широкого поширення серед розробників завдяки поєднанню високої функціональності, гнучкості налаштувань та відносної простоти використання [17], [18].

Середовище розробки Visual Studio Code посідає важливе місце серед сучасних інструментів програмування, які використовуються для створення програмних систем різного призначення. Воно поєднує можливості текстового редактора з функціями повноцінного середовища розробки, що робить його придатним для реалізації складних програмних проєктів. У межах цього

програмного інструмента забезпечується підтримка великої кількості мов програмування та технологій, що дозволяє використовувати його як універсальне середовище для створення вебзастосунків, серверних сервісів та інших програмних продуктів [17], [18].

Однією з характерних особливостей цього середовища є його модульна архітектура. Основний функціонал може бути розширений за допомогою додаткових модулів, що встановлюються через спеціальну систему розширень. Такий підхід дозволяє адаптувати середовище до конкретних потреб розроблення, додаючи інструменти для роботи з певними мовами програмування, фреймворками або технологіями. Завдяки цьому забезпечується можливість створення індивідуально налаштованого робочого програмного середовища, яке відповідає вимогам конкретного програмного проєкту [17], [18].

Суттєвою перевагою Visual Studio Code є підтримка інтелектуальних механізмів аналізу програмного коду. Під час написання програм автоматично пропонуються підказки щодо завершення команд, відображаються можливі параметри функцій та здійснюється перевірка синтаксису. Це дозволяє підвищити швидкість програмування та зменшити кількість помилок, що виникають у процесі розроблення. Такі можливості особливо корисні під час створення великих програмних систем, у яких програмний код складається з багатьох взаємопов'язаних компонентів [17], [18].

Окремої уваги заслуговують засоби налагодження програм. У середовищі Visual Studio Code передбачено можливість виконання програмного коду у режимі покрокового аналізу, встановлення точок зупинки та перегляду значень змінних під час виконання програми. Це дозволяє ефективно виявляти помилки та досліджувати поведінку програмного забезпечення у різних ситуаціях. Завдяки таким інструментам значно спрощується процес тестування програмних модулів та оптимізації алгоритмів оброблення даних [17], [18].

Важливим аспектом є також можливість інтеграції з різними інструментами автоматизації розроблення. Середовище підтримує використання систем керування пакетами, інструментів збірки програм, механізмів тестування та інших програмних компонентів, що застосовуються під час створення сучасного програмного забезпечення. Це дозволяє організувати цілісний процес розроблення, у межах якого програмування, тестування та розгортання програмних систем здійснюється у межах єдиного програмного середовища [17], [18].

Не менш важливою характеристикою є зручність організації робочого простору. У Visual Studio Code передбачено можливість роботи з багатьма файлами та проєктами одночасно, що полегшує навігацію у структурі програмного коду. Інтерфейс середовища дозволяє швидко переходити між різними частинами програмного проєкту, здійснювати пошук по файлах та переглядати структуру каталогів. Це особливо корисно під час розроблення великих інформаційних систем, у яких програмний код може складатися з сотень або навіть тисяч файлів [17], [18].

Значну роль відіграє також підтримка різних операційних систем. Visual Studio Code може використовуватися у середовищах Windows, Linux та macOS, що забезпечує універсальність цього інструмента. Така особливість дозволяє застосовувати його у різних технологічних середовищах без необхідності зміни інструментів розроблення [17], [18].

Таким чином, Visual Studio Code характеризується широкими функціональними можливостями, гнучкістю налаштування та високим рівнем інтеграції з сучасними технологіями програмування. Сукупність цих характеристик робить це середовище ефективним інструментом для створення програмних систем різного рівня складності, зокрема веборієнтованих застосунків і сервісів, що працюють з великими обсягами інформації [17], [18].

Застосування цього середовища розробки забезпечує можливість ефективної роботи з програмним кодом РНР, оскільки підтримується велика кількість розширень, які розширюють базові функції редактора та дозволяють

адаптувати його до потреб конкретного проєкту. За допомогою спеціалізованих плагінів забезпечується автоматичне підсвічування синтаксису, перевірка коректності програмного коду, інтеграція інструментів форматування та підтримка інтелектуальних підказок під час написання програм. Це сприяє підвищенню якості програмного продукту та зменшує ймовірність виникнення помилок у процесі розроблення [17], [18].

Важливою характеристикою цього середовища є можливість інтеграції з сучасними інструментами керування версіями програмного коду. Підтримка системи контролю версій дозволяє організувати ефективне відстеження змін у програмному проєкті, що особливо важливо під час розроблення складних інформаційних систем, які містять значну кількість модулів і компонентів. У випадку створення програмної системи для роботи з новинними потоками така функціональність дозволяє контролювати розвиток програмного продукту та спрощує процес колективної роботи над кодом [17], [18].

Окремої уваги заслуговує здатність цього середовища ефективно взаємодіяти з різноманітними вебтехнологіями. Оскільки програмна система агрегації новин передбачає використання не лише серверної логіки на PHP, але й клієнтських технологій, пов'язаних із розробленням вебінтерфейсу, важливо забезпечити підтримку мов розмітки та сценаріїв, що використовуються у веброзробці. У цьому контексті середовище розробки Visual Studio Code забезпечує зручну роботу з HTML, CSS, JavaScript та іншими технологіями, що дозволяє реалізувати комплексний підхід до створення програмної системи [17], [18].

Значною перевагою середовища розробки Visual Studio Code є його висока швидкодія та відносно невеликі вимоги до системних ресурсів. Це дозволяє використовувати його навіть на комп'ютерах із середнім рівнем продуктивності без суттєвого зниження ефективності роботи. У процесі створення програмного забезпечення для оброблення новинних потоків, коли необхідно постійно тестувати та досліджувати алгоритми оброблення даних та

модифікувати програмний код, стабільна робота середовища розробки відіграє важливу роль [17], [18].

Суттєвим фактором є також активний розвиток екосистеми цього програмного інструмента. Постійне оновлення функціональності, розширення бібліотеки плагінів і підтримка з боку великої спільноти розробників сприяють підвищенню ефективності використання середовища у процесі створення програмних продуктів. Завдяки цьому забезпечується доступ до сучасних інструментів налагодження, тестування та оптимізації програмного коду [17], [18].

Обґрунтування вибору середовища розробки для створення програмної системи для агрегації та оброблення новинних потоків наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмної системи для агрегації та оброблення новинних потоків

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio Code	NetBeans	Eclipse
Простота встановлення та налаштування	+	+—	—
Швидкість роботи середовища	+	+—	+—
Зручність роботи з вебтехнологіями	+	+	+—
Наявність великої кількості розширень	+	+—	+
Зручність навігації по проекту	+	+—	+—

Таким чином, для створення програмної системи для агрегації та оброблення новинних потоків обрано середовище розробки Visual Studio Code через поєднання функціональності, гнучкості налаштувань, високої продуктивності та широких можливостей інтеграції з сучасними інструментами веброботів. Це середовище дозволяє організувати ефективний процес створення програмного забезпечення, забезпечує зручну роботу з програмним кодом та сприяє підвищенню якості кінцевого програмного продукту.

2.3 Вибір засобів зберігання та опрацювання даних

Для зберігання та опрацювання даних було обрано систему управління базами даних SQLite [19], [20].

Під час розроблення програмної системи, призначеної для агрегації та оброблення новинних потоків, важливим етапом є вибір системи управління базами даних, яка забезпечуватиме надійне зберігання інформації, швидкий доступ до даних та ефективне виконання операцій пошуку й оброблення. У цьому контексті доцільним рішенням може вважатися використання SQLite, оскільки ця система поєднує простоту інтеграції, високу продуктивність та мінімальні вимоги до серверної інфраструктури [19], [20].

Однією з ключових особливостей SQLite є її вбудована архітектура. На відміну від багатьох інших систем управління базами даних, вона не потребує встановлення окремого серверного програмного забезпечення або налаштування складної серверної конфігурації. Уся база даних зберігається у вигляді одного файлу, що значно спрощує процес розгортання програмної системи та її подальше використання. Такий підхід дозволяє інтегрувати механізми зберігання даних безпосередньо у програмний застосунок, що є особливо зручним під час створення веборієнтованих систем, які працюють з потоками інформаційних повідомлень [19], [20].

Важливою перевагою використання системи управління базами даних SQLite є ефективність роботи з відносно великими обсягами текстових даних. Програмні системи, що виконують агрегацію новин, зазвичай оперують значною кількістю текстових матеріалів, які надходять із різних інформаційних джерел. У межах SQLite реалізовано механізми індексування, що дозволяють прискорити процес пошуку та сортування даних. Завдяки цьому забезпечується швидке отримання необхідних новинних повідомлень відповідно до визначених критеріїв, наприклад за тематикою, датою публікації або джерелом інформації [19], [20].

Суттєвою характеристикою цієї системи є її компактність і низькі вимоги до апаратних ресурсів. Відсутність окремого серверного процесу зменшує навантаження на систему та дозволяє використовувати базу даних навіть у середовищах з обмеженими ресурсами. Для програмних систем, які виконують періодичне отримання та оброблення новинних потоків, це є важливою перевагою, оскільки дозволяє ефективно зберігати інформацію без необхідності використання складної серверної інфраструктури [19], [20].

Додатковою перевагою є висока швидкість виконання операцій читання даних. У системах агрегації новин значна частина операцій пов'язана саме з отриманням та аналізом інформації, яка вже збережена у базі. SQLite оптимізована для виконання таких запитів, що сприяє швидкому відображенню новинної інформації у програмному інтерфейсі та підвищує загальну ефективність роботи програмної системи [19], [20].

Не менш важливою характеристикою є сумісність цієї системи з різними мовами програмування та програмними платформами. Система управління базами даних SQLite підтримується великою кількістю програмних середовищ, зокрема мовою PHP, що забезпечує зручну інтеграцію з вебзастосунками. У результаті створюється можливість реалізувати ефективну взаємодію між серверною логікою програмної системи та механізмами зберігання інформації без необхідності використання складних програмних інструментів [19], [20].

Важливо також враховувати високий рівень надійності цієї системи управління базами даних. У процесі роботи SQLite забезпечує підтримку транзакцій, що дозволяє гарантувати цілісність даних під час виконання операцій запису або оновлення інформації. Для програмних систем, які регулярно отримують нові новинні повідомлення з різних джерел, така функціональність дозволяє уникнути втрати даних або виникнення некоректних записів у базі [19], [20].

Обґрунтування вибору засобів зберігання та опрацювання даних наведено у таблиці 2.3.

Таблиця 2.3 – Обґрунтування вибору засобів зберігання та опрацювання даних

Критерій порівняння систем управління базами даних	Системи управління базами даних		
	SQLite	MySQL	PostgreSQL
Відсутність необхідності окремого серверу	+	—	—
Компактність та легкість розгортання	+	+—	+—
Швидкість роботи з невеликими та середніми обсягами даних	+	+—	+—
Низькі вимоги до апаратних ресурсів	+	+—	—
Легкість резервного копіювання та переносу	+	+—	+—

Таким чином, для зберігання та опрацювання даних обрано систему управління базами даних SQLite, яка дозволяє ефективно організувати зберігання текстової інформації, забезпечити швидкий доступ до новинних

матеріалів і створити стабільну основу для функціонування програмного застосунку, орієнтованого на оброблення великих обсягів інформаційних повідомлень.

2.4 Висновки за розділом 2

Для реалізації програмної системи для агрегації та оброблення новинних потоків обрано мову програмування PHP, яка забезпечує ефективну інтеграцію з вебтехнологіями, наявність засобів для отримання інформації з мережеских джерел, зручну взаємодію з базами даних, простоту реалізації серверної логіки та широке поширення у середовищі вебінфраструктури. Сукупність цих характеристик мови PHP створює сприятливі умови для розроблення функціональної, масштабованої та зручної у підтримці програмної системи, здатної ефективно працювати з великими потоками новинної інформації.

Для створення програмної системи для агрегації та оброблення новинних потоків обрано середовище розробки Visual Studio Code через поєднання функціональності, гнучкості налаштувань, високої продуктивності та широких можливостей інтеграції з сучасними інструментами веброботів. Це середовище дозволяє організувати ефективний процес створення програмного забезпечення, забезпечує зручну роботу з програмним кодом та сприяє підвищенню якості кінцевого програмного продукту.

Для зберігання та опрацювання даних обрано систему управління базами даних SQLite, яка дозволяє ефективно організувати зберігання текстової інформації, забезпечити швидкий доступ до новинних матеріалів і створити стабільну основу для функціонування програмного застосунку, орієнтованого на оброблення великих обсягів інформаційних повідомлень.

3 ОПИС ПРОГРАМНОЇ СИСТЕМИ

3.1 Структура програмної системи для агрегації та оброблення новинних потоків

Структуру програмної системи для агрегації та оброблення новинних потоків наведено на рис. 3.1.

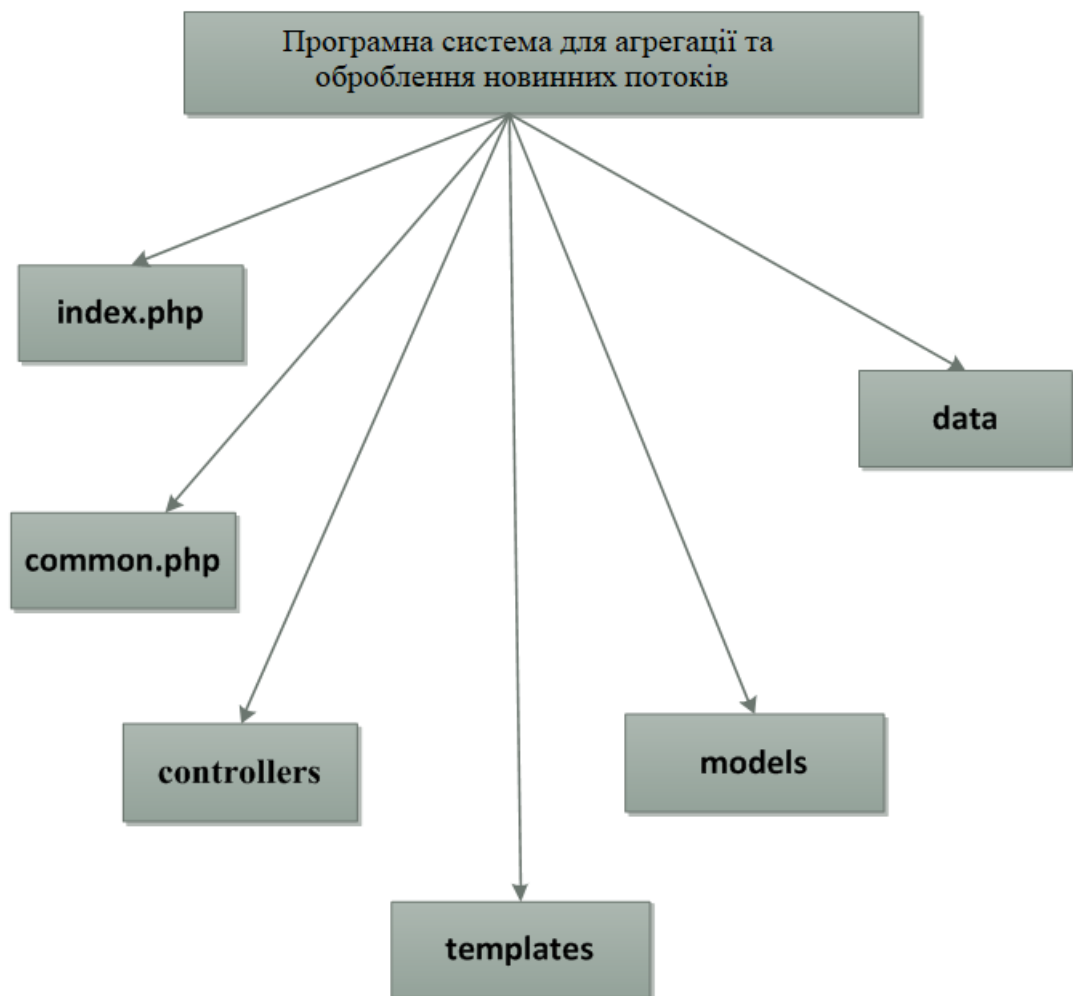


Рисунок 3.1 – Структура програмної системи для агрегації та оброблення новинних потоків

Побудова програмної системи для агрегації та оброблення новинних потоків потребує чітко визначеної внутрішньої організації програмних компонентів. Формування структури програмної системи для агрегації та оброблення новинних потоків здійснено з урахуванням принципів

модульності та розподілу функціональних обов'язків між окремими частинами системи. Такий підхід дозволяє забезпечити зрозумілу логіку взаємодії між програмними модулями, підвищити гнучкість програмного коду та спростити подальше розширення функціональності системи. Архітектурною основою розроблення програмного рішення для агрегації новин слугувала концепція Model–View–Controller (MVC), відповідно до якої функції зберігання даних, оброблення інформації та відображення результатів розділяються між різними компонентами програмного забезпечення.

У межах такої архітектури важливу роль відіграє підсистема, що відповідає за роботу з даними. Для цього передбачено спеціальний програмний модуль, який реалізує логіку взаємодії з базою даних і забезпечує виконання запитів до сховища інформації. У цьому компоненті реалізуються механізми отримання новинних записів, збереження інформації про джерела новин, оброблення запитів пошуку та формування результатів для подальшої передачі іншим частинам системи. Завдяки такій організації програмна логіка, пов'язана зі зберіганням і вибіркою даних, ізольована від інших функціональних частин програми, що сприяє підвищенню зрозумілості структури програмного коду та полегшує його супровід.

Зберігання інформації, необхідної для функціонування системи, здійснюється у спеціальному програмному каталозі, у межах якого розміщується база даних та допоміжні файли, що використовуються програмою. Основним елементом цього компонента є файл бази даних, створений із використанням системи управління базами даних SQLite. У цьому сховищі зберігаються новинні повідомлення, відомості про джерела інформації, параметри підписок користувачів, а також інші службові дані, необхідні для функціонування програмної системи. Окрім основного файлу бази даних, у цій частині структури можуть розміщуватися резервні копії інформації, архівні дані та інші допоміжні файли, які забезпечують збереження та відновлення інформації у разі потреби.

Окреме місце у структурі програмної системи для агрегації та оброблення новинних потоків займає компонент, відповідальний за відображення інформації користувачеві. Цей модуль містить набір шаблонів вебінтерфейсу, які формують зовнішній вигляд сторінок програмної системи. У таких шаблонах використовуються елементи мови розмітки HTML разом із вбудованими сценаріями PHP, що дозволяє поєднувати статичну структуру вебсторінок із динамічним формуванням інформаційного наповнення. У межах цього компонента реалізується відображення різних елементів інтерфейсу системи, зокрема перегляд списку новин, відображення історії перегляду, виконання пошуку інформації, управління підписками на новинні джерела, редагування параметрів підписки, імпорт новинних стрічок та налаштування параметрів відображення інформації. Таким чином, зазначений компонент забезпечує візуалізацію результатів оброблення даних і формує інтерфейс взаємодії користувача з програмною системою.

Центральною ланкою програмної структури виступає компонент, який координує взаємодію між користувацьким інтерфейсом та системою зберігання даних. У цьому модулі реалізується логіка оброблення запитів користувача, а також здійснюється керування передаванням інформації між іншими складовими програмної системи. Саме тут відбувається приймання введених користувачем даних, визначення необхідних дій та формування відповідних запитів до підсистеми роботи з базою даних. Після отримання результатів оброблення ці дані передаються до модулів відображення, які формують вебсторінки з відповідною інформацією. Завдяки такій організації контролюючий компонент виконує роль проміжного рівня між інтерфейсом користувача та внутрішніми механізмами оброблення інформації.

Загальна структура програмної системи для агрегації та оброблення новинних потоків сформована таким чином, щоб кожен її компонент виконував чітко визначені функції та взаємодівав з іншими частинами програмного забезпечення через визначені механізми обміну даними. Такий підхід дозволяє досягти високого рівня впорядкованості програмного коду,

забезпечити можливість модифікації окремих частин системи без необхідності суттєвого втручання в інші модулі та створити умови для подальшого розширення функціональності програмного продукту.

Отже, структура програмної системи для агрегації та оброблення новинних потоків базується на принципах розподілу функцій між окремими компонентами, що відповідають за роботу з даними, їх зберігання, оброблення запитів користувача та відображення результатів. Така організація програмного забезпечення забезпечує логічну впорядкованість програмної архітектури, спрощує процес розроблення та створює передумови для подальшого розвитку і вдосконалення системи.

3.2 Функціонування програмної системи для агрегації та оброблення новинних потоків

Функціонування програмної системи для агрегації та оброблення новинних потоків подамо за допомогою схеми, зображеної на рис. 3.2.

Функціонування програмної системи, призначеної для автоматизованого збирання та оброблення новинної інформації, організовано таким чином, щоб забезпечити зручний доступ до новинних повідомлень, отриманих із різноманітних інформаційних джерел. Робота програми базується на використанні кількох основних режимів взаємодії з користувачем, кожен з яких відповідає за виконання певних завдань, пов'язаних із переглядом, пошуком, збереженням та керуванням новинним контентом. Такий підхід дозволяє впорядкувати інформаційний потік і забезпечити ефективне опрацювання великої кількості новинних повідомлень, що надходять із RSS-джерел.



Рисунок 3.2 – Схема функціонування програмної системи для агрегації та оброблення новинних потоків

Одним із ключових режимів функціонування програмної системи для агрегації та оброблення новинних потоків є відображення новин, які ще не були переглянуті користувачем. У цьому режимі програма автоматично формує список нових інформаційних повідомлень, отриманих із підписаних джерел, та представляє їх у зручному для перегляду вигляді. Завдяки цьому користувач отримує можливість швидко ознайомитися з актуальними новинами, не витрачаючи час на повторний перегляд уже опрацьованих матеріалів. Система відстежує стан кожного повідомлення та фіксує факт його перегляду, що дозволяє відокремлювати нові повідомлення від тих, які вже були прочитані.

Іншим важливим режимом роботи є можливість виконання пошуку інформації. У межах цього функціонального механізму користувач може знаходити необхідні новинні матеріали серед тих джерел, на які оформлено підписку. Пошук може здійснюватися за ключовими словами або іншими параметрами, що дозволяє швидко отримати доступ до потрібної інформації навіть у разі значної кількості новинних повідомлень. Така можливість є особливо корисною у випадках, коли необхідно знайти новини, пов'язані з певною подією або тематикою.

Програмна система також передбачає можливість збереження окремих новин для подальшого використання. Для цього реалізовано режим роботи із закладками, який дозволяє позначати важливі або цікаві інформаційні матеріали. Збережені у такий спосіб новини формують окремий список, що забезпечує швидкий доступ до них у будь-який момент часу. Завдяки цьому користувач може повертатися до вибраних матеріалів без необхідності повторного пошуку у загальному потоці новин.

Окрім цього, система підтримує механізм перегляду історії взаємодії з новинними джерелами. У відповідному режимі відображається перелік повідомлень, які були переглянуті раніше. Така функціональність дозволяє відстежувати інформацію, з якою вже було ознайомлено користувача, а також за потреби повторно звертатися до раніше прочитаних новинних матеріалів.

Наявність історії перегляду сприяє більш впорядкованій роботі з інформаційними потоками та забезпечує можливість аналізу вже отриманої інформації.

Важливе місце у функціонуванні програмної системи займає механізм керування підписками на новинні джерела. Саме завдяки цьому компоненту користувач отримує можливість формувати перелік інформаційних ресурсів, з яких буде здійснюватися отримання новин. У межах відповідного режиму реалізовано перегляд списку підключених RSS-каналів, а також виконання операцій з їх додавання або редагування. Крім того, передбачено можливість імпорту та експорту переліку підписок, що дозволяє переносити налаштування системи між різними пристроями або створювати резервні копії конфігурації новинних джерел.

Функціональність керування підписками доповнюється спеціальним допоміжним меню, яке забезпечує доступ до окремих операцій роботи з RSS-стрічками. За допомогою цього меню користувач може оформлювати нову підписку на інформаційне джерело, переглядати перелік уже підключених стрічок, а також виконувати операції імпорту або експорту списку підписок. Такий механізм дозволяє значно спростити процес управління джерелами новин і забезпечує зручне налаштування інформаційного середовища користувача.

Ще одним важливим елементом функціонування програмної системи є режим налаштування. У межах цього режиму здійснюється конфігурація параметрів відображення новинної інформації та визначення умов її зберігання. Користувач може змінювати параметри інтерфейсу, визначати спосіб відображення новинних повідомлень, а також встановлювати правила оброблення новинних стрічок. Така можливість дозволяє адаптувати роботу програмної системи до індивідуальних потреб користувача та створити найбільш зручне середовище для роботи з інформаційними потоками.

У процесі функціонування програмна система регулярно отримує нові повідомлення з підписаних RSS-джерел, зберігає їх у базі даних та забезпечує

їх подальше відображення у відповідних режимах роботи. Користувач взаємодіє з програмою через вебінтерфейс, який надає доступ до всіх функціональних можливостей системи. У результаті забезпечується централізований доступ до новинної інформації, її впорядкування та зручне використання.

Отже, функціонування програмної системи для агрегації та оброблення новинних потоків базується на поєднанні механізмів автоматичного отримання інформації з RSS-джерел та зручних інструментів взаємодії користувача з новинними даними. Реалізовані режими роботи дозволяють ефективно переглядати, шукати, зберігати та впорядковувати новинні повідомлення, а також керувати джерелами їх отримання. Така організація роботи забезпечує зручне використання системи та сприяє більш ефективному опрацюванню інформаційних потоків.

3.3 Розробка бази даних програмної системи для агрегації та оброблення новинних потоків

Оскільки програмна система працює з інформаційними потоками, що постійно оновлюються, новини повинні накопичуватися у структурованому сховищі даних. Це необхідно для забезпечення можливості перегляду нових повідомлень, повторного доступу до вже прочитаних матеріалів, створення закладок, а також для реалізації механізмів автоматичного видалення застарілої інформації відповідно до параметрів, встановлених користувачем. З цією метою розроблено реляційну базу даних, яка створена з використанням системи керування базами даних SQLite. Такий підхід дозволяє організувати ефективне зберігання інформації та забезпечити швидке виконання запитів до даних.

Схему бази даних програмної системи для агрегації та оброблення новинних потоків наведено на рис. 3.3.

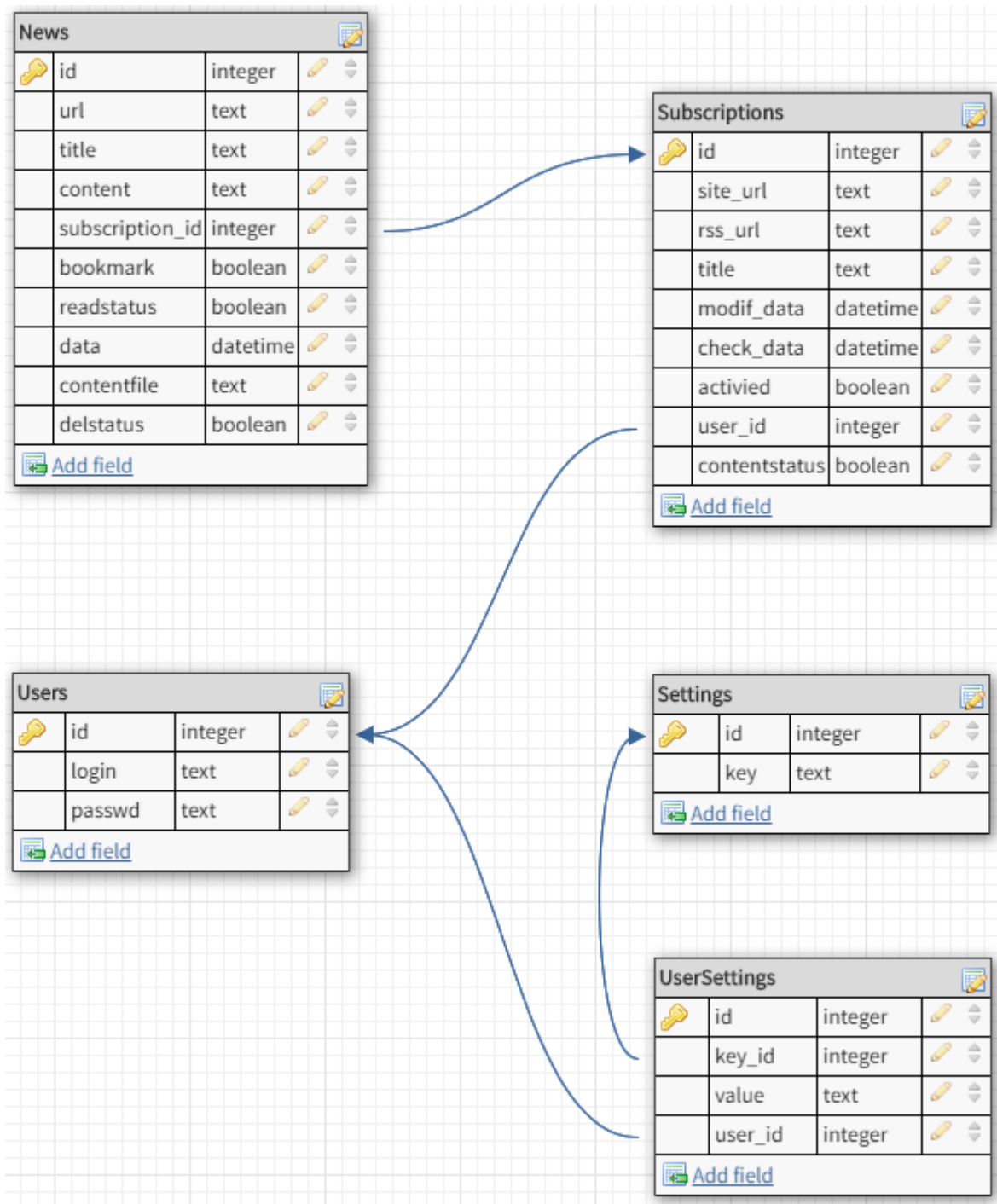


Рисунок 3.3 – Схема бази даних програмної системи для агрегації та оброблення новинних потоків

Структура бази даних сформована на основі кількох ключових інформаційних об'єктів, які відображають основні елементи функціонування системи. Центральним елементом є таблиця, що містить інформацію про новинні повідомлення. У цій таблиці зберігаються всі новини, отримані із зовнішніх джерел, а також метадані, необхідні для їх оброблення та

відображення. Для кожного запису передбачено унікальний ідентифікатор, який дозволяє однозначно ідентифікувати новину в межах бази даних. Окрім цього, фіксується веб-адреса новинного матеріалу, його заголовок і текстовий зміст. З метою організації зручної роботи з інформаційним потоком у таблиці також передбачено ознаки, що визначають, чи було створено для новини закладку та чи переглянув її користувач. Для збереження інформації про час появи новинного повідомлення використовується поле, у якому фіксується дата та час його створення або отримання. Окремо передбачено поле, що містить посилання на файл із повним текстом новини, якщо основний вміст зберігається у файловій системі. Додатково використовується спеціальний параметр, який сигналізує про те, що запис підготовлено до видалення з бази даних відповідно до політики очищення інформації.

Новини в системі тісно пов'язані з джерелами їх отримання. Для цього створено окрему таблицю, яка містить інформацію про підписки на новинні ресурси. У межах цієї таблиці зберігаються відомості про інтернет-ресурс, з якого здійснюється отримання новин, а також адреса RSS-стрічки, що використовується для автоматичного завантаження нових повідомлень. Кожна підписка має власний ідентифікаційний номер, назву та часові параметри, які відображають момент останньої зміни конфігурації підписки та час останнього оновлення новин. Додатково фіксується інформація про те, чи є підписка активною в поточний момент часу. Окремий параметр визначає, чи повинна система завантажувати повний текст новинного матеріалу, що дозволяє керувати обсягом інформації, яка зберігається у базі даних. Зв'язок між новинами та підписками реалізовано за допомогою зовнішнього ключа, що забезпечує належність кожного новинного запису до конкретного джерела інформації.

Для організації персоналізованого доступу до програмної системи використовується таблиця, яка містить інформацію про користувачів. У цьому компоненті бази даних зберігаються відомості, необхідні для ідентифікації облікових записів. Кожен користувач має власний унікальний ідентифікатор, а

також дані для авторизації у системі, зокрема логін і пароль. Наявність такого механізму дозволяє організувати індивідуальне середовище роботи з новинними потоками, у межах якого кожен користувач може формувати власний перелік підписок і налаштовувати параметри відображення інформації. Зв'язок між користувачами та підписками реалізовано таким чином, що один користувач може бути пов'язаний із багатьма підписками, тоді як кожна підписка належить лише одному користувачеві.

Окрему роль у структурі бази даних відіграє таблиця, призначена для зберігання параметрів налаштування системи. У ній містяться записи, що визначають назви різних параметрів конфігурації, які можуть використовуватися програмою. Ці параметри задають правила функціонування системи, наприклад умови зберігання новин або особливості їх відображення у користувацькому інтерфейсі. Оскільки кожен користувач може встановлювати власні значення для таких параметрів, виникає необхідність реалізації механізму зберігання індивідуальних налаштувань. Для цього створюється додаткова таблиця, яка пов'язує користувачів із відповідними параметрами конфігурації. У цьому компоненті бази даних фіксується ідентифікатор конкретного налаштування, значення, встановлене користувачем, а також ідентифікатор облікового запису, до якого це налаштування належить. Така структура дозволяє забезпечити гнучку систему персоналізації роботи програмного забезпечення.

Організація взаємозв'язків між таблицями бази даних базується на використанні відношень типу один-до-багатьох та багато-до-багатьох. Зокрема, одне джерело новин може бути пов'язане з великою кількістю новинних записів, тоді як кожна новина належить лише до однієї підписки. Аналогічно, один користувач може створювати багато підписок на різні новинні ресурси. Щодо параметрів налаштування, то для реалізації взаємодії між користувачами та конфігураційними параметрами використовується проміжна таблиця, яка дозволяє пов'язувати кожного користувача з набором налаштувань і зберігати індивідуальні значення цих параметрів.

Таким чином, база даних програмної системи побудована на основі логічно взаємопов'язаних таблиць, які відображають основні елементи роботи з новинними потоками. Вона забезпечує зберігання інформації про новини, джерела їх отримання, користувачів та параметри конфігурації системи, а також реалізує необхідні зв'язки між цими елементами.

Отже, розроблена структура бази даних створює надійну основу для функціонування програмної системи агрегації новин. Вона дозволяє ефективно організувати накопичення інформації, забезпечує можливість швидкого доступу до новинних матеріалів та підтримує гнучку систему налаштувань для користувачів, що сприяє стабільній та зручній роботі програмного забезпечення.

3.4 Проєктування інтерфейсу програмної системи для агрегації та оброблення новинних потоків

Проєктування інтерфейсу взаємодії користувача з програмною системою для агрегації та оброблення новинних потоків виконано з урахуванням вимог технічного завдання. При розробленні програмної системи для агрегації та оброблення новинних потоків враховані функціональні вимоги до програми:

- підтримка можливості отримання новин із зовнішніх інформаційних ресурсів на основі новинних стрічок;
- можливість відображення новин у зручному для користувача вигляді через вебінтерфейс;
- можливість реєстрації та авторизації користувачів у програмній системі;
- підтримка можливості створення та керування підписками на новинні стрічки;
- підтримка можливості додавання нових підписок на новинні ресурси;

- підтримка можливості перегляду новин, що ще не були прочитані користувачем;
- можливість налаштування параметрів відображення та зберігання новин у системі;
- підтримка можливості автоматичного оновлення новинних стрічок.

Інтерфейс головної сторінки програмної системи для агрегації та оброблення новинних потоків наведено на рис. 3.4.

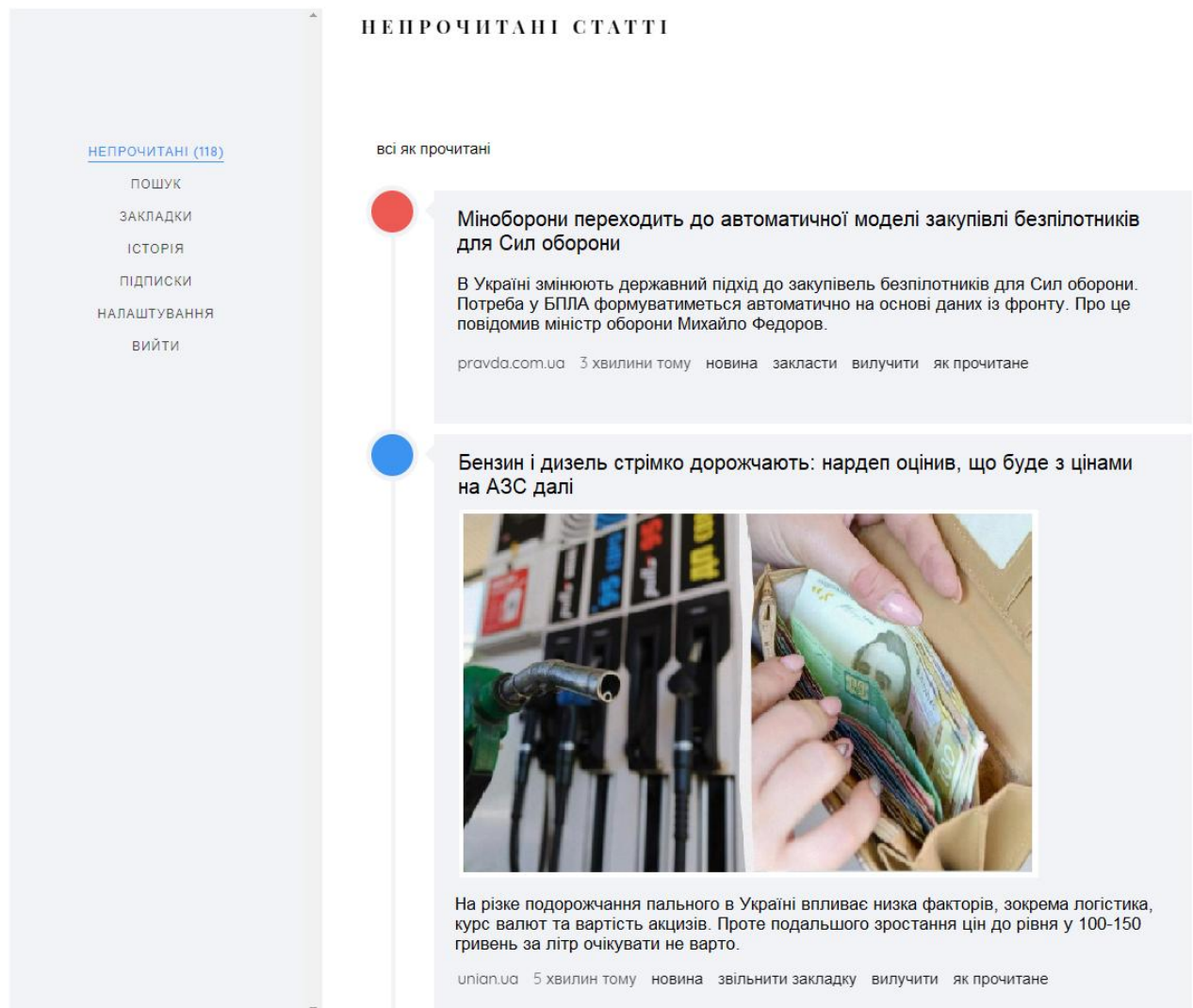


Рисунок 3.4 – Інтерфейс головної сторінки програмної системи для агрегації та оброблення новинних потоків

Інтерфейс вкладки додавання новинного каналу наведено на рис. 3.5.

Рисунок 3.5 – Інтерфейс вкладки додавання новинного каналу

Форму авторизації користувача наведено на рис. 3.6.

Рисунок 3.6 – Форма авторизації користувача

3.5 Висновки за розділом 3

Розроблено програмну систему для агрегації та оброблення новинних потоків.

Запропоновано структуру програмної системи для агрегації та оброблення новинних потоків. При розробці структури програми враховано принципи модульності та розподілу функціональних обов’язків між окремими частинами програмної системи, що дозволяє забезпечити зрозумілу логіку взаємодії між програмними модулями, підвищити гнучкість програмного коду

та спростити подальше розширення функціональності системи. Архітектурною основою розроблення програмної системи для агрегації та оброблення новинних потоків слугувала концепція Model–View–Controller (MVC), відповідно до якої функції зберігання даних, оброблення інформації та відображення результатів розділяються між різними компонентами програмного забезпечення.

Описано функціонування програмної системи для агрегації та оброблення новинних потоків, що базується на поєднанні механізмів автоматичного отримання інформації з RSS-джерел та зручних інструментів взаємодії користувача з новинними даними. Реалізовані режими роботи дозволяють ефективно переглядати, шукати, зберігати та впорядковувати новинні повідомлення, а також керувати джерелами їх отримання. Така організація роботи забезпечує зручне використання системи та сприяє більш ефективному опрацюванню інформаційних потоків.

Розроблено базу даних програмної системи для агрегації та оброблення новинних потоків, яка побудована на основі логічно взаємопов'язаних таблиць, що відображають основні елементи роботи з новинними потоками. Вона забезпечує зберігання інформації про новини, джерела їх отримання, користувачів та параметри конфігурації системи, а також реалізує необхідні зв'язки між цими елементами. Розроблена база даних дозволяє ефективно організувати накопичення інформації, забезпечує можливість швидкого доступу до новинних матеріалів та підтримує гнучку систему налаштувань для користувачів, що сприяє стабільній та зручній роботі програмного забезпечення.

Виконано проєктування інтерфейсу взаємодії користувача з програмною системою для агрегації та оброблення новинних потоків.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ

4.1 Призначення й умови застосування програмної системи

Програмна система призначена для підтримки процесу агрегації та оброблення новинних потоків. Програмна система написана на мові PHP, яка забезпечує ефективну інтеграцію з вебтехнологіями, наявність засобів для отримання інформації з мережових джерел, зручну взаємодію з базами даних, простоту реалізації серверної логіки та широке поширення у середовищі вебінфраструктури. В якості середовища розробки для створення програмної системи для агрегації та оброблення новинних потоків обрано Visual Studio Code через поєднання функціональності, гнучкості налаштувань, високої продуктивності та широких можливостей інтеграції з сучасними інструментами веброзроблення.

Програмна система для агрегації та оброблення новинних потоків забезпечує виконання таких функцій:

- підтримка можливості отримання новин із зовнішніх інформаційних ресурсів на основі новинних стрічок;
- можливість відображення новин у зручному для користувача вигляді через вебінтерфейс;
- можливість реєстрації та авторизації користувачів у програмній системі;
- підтримка можливості створення та керування підписками на новинні стрічки;
- підтримка можливості додавання нових підписок на новинні ресурси;
- підтримка можливості перегляду новин, що ще не були прочитані користувачем;
- можливість налаштування параметрів відображення та зберігання новин у системі;
- підтримка можливості автоматичного оновлення новинних стрічок.

Для роботи програмної системи для агрегації та оброблення новинних потоків потрібні такі технічні та програмні ресурси: центральний процесор повинен мати тактову частоту не нижче 2,0 ГГц, що дозволяє забезпечити стабільне виконання програмного коду, оброблення запитів користувачів і формування динамічних вебсторінок. Оперативна пам'ять комп'ютера повинна становити не менше 4 ГБ, оскільки під час функціонування програмної системи одночасно виконуються процеси оброблення RSS-стрічок, формування запитів до бази даних та відображення інформації у веб-інтерфейсі. Для зберігання програмних файлів, бази даних і допоміжних ресурсів необхідно передбачити наявність вільного дискового простору не менше 1 ГБ. Щодо програмного забезпечення, то функціонування системи передбачає наявність операційної системи, яка підтримує роботу вебсерверів і середовищ виконання сценаріїв. Для цього можуть використовуватися сучасні операційні системи, такі як Microsoft Windows, Ubuntu або інші дистрибутиви на базі Linux. Використання таких операційних систем забезпечує стабільне середовище для запуску серверних компонентів і програмних модулів системи.

Для забезпечення роботи вебчастини програми необхідно встановити вебсервер, який оброблятиме HTTP-запити користувачів. Оскільки логіка роботи програмної системи реалізується за допомогою серверних сценаріїв, необхідною умовою є встановлення середовища виконання мови програмування PHP версії не нижче 7.4. Використання сучасної версії мови забезпечує підтримку актуальних можливостей програмування, підвищений рівень безпеки та кращу продуктивність під час виконання програмного коду.

Для зберігання та оброблення інформації про новини, підписки та користувачів застосовується система керування базами даних SQLite. Вона не потребує встановлення окремого серверного компонента та працює безпосередньо з файлом бази даних, що спрощує розгортання програмної системи та зменшує вимоги до системних ресурсів.

Для взаємодії користувача з програмною системою необхідний сучасний веббраузер, який підтримує роботу з HTML-розміткою, каскадними таблицями стилів та клієнтськими сценаріями. Для цієї мети можуть використовуватися такі браузери, як Google Chrome, Mozilla Firefox або Microsoft Edge.

Функціональні характеристики розробленої програмної системи для агрегації та оброблення новинних потоків наведено у технічному завданні (додаток А). Фрагмент тексту програмного коду програмної системи для агрегації та оброблення новинних потоків наведено у додатку Б.

4.2 Характеристики програмної системи для агрегації та оброблення новинних потоків

Програмна система для агрегації та оброблення новинних потоків характеризується як інформаційний інструмент, призначений для автоматизованого отримання, збереження, аналізу та відображення новинних повідомлень, що надходять із різних мережевих джерел. Основною особливістю такої системи є можливість централізованого збирання інформації, завдяки чому користувач отримує доступ до великої кількості новин у межах єдиного інтерфейсу. У процесі функціонування програмного забезпечення відбувається регулярне звернення до зовнішніх інформаційних ресурсів, з яких завантажуються нові повідомлення, після чого вони проходять етапи збереження, впорядкування та підготовки до подальшого відображення.

Важливою характеристикою програмної системи є здатність організовувати новинний контент у структурованому вигляді. Інформація зберігається у впорядкованому сховищі даних, що забезпечує швидкий доступ до матеріалів, а також дозволяє виконувати різноманітні операції з новинами.

Функціональні можливості системи передбачають опрацювання новинних повідомлень після їх отримання. Під час цього процесу відбувається

аналіз метаданих, формування внутрішньої структури записів і підготовка інформації до подальшого використання.

Ще однією характерною рисою програмної системи є можливість збереження історії взаємодії користувача з новинами. У процесі перегляду повідомлень фіксується інформація про виконані дії, що дозволяє формувати перелік вже переглянутих матеріалів.

Особлива увага приділяється можливості персоналізації роботи системи. Передбачається наявність механізмів налаштування, за допомогою яких користувач може змінювати параметри відображення інформації, керувати підписками на новинні ресурси та визначати правила зберігання новин у системі. Характерною властивістю програмної системи є також підтримка механізмів імпорту та експорту підписок. Значну роль у функціонуванні системи відіграє інтерфейс взаємодії з користувачем. Він реалізується у вигляді веборієнтованого середовища, що дозволяє отримувати доступ до новинної інформації через стандартні програмні засоби перегляду вебсторінок. Інтерфейс орієнтований на зручність використання, що проявляється у логічній організації елементів керування, зрозумілій структурі сторінок і швидкому доступі до основних функцій системи.

Таким чином, програмна система для агрегації та оброблення новинних потоків характеризується комплексом властивостей, спрямованих на ефективне збирання, впорядкування та подання інформації з різних джерел.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмної системи для агрегації та оброблення новинних потоків на серверній частині необхідно запустити відповідний ру-файл. Для звертання до клієнтської сторони необхідно перейти на відповідну вебсторінку.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмної системи для агрегації та оброблення новинних потоків є інформація, яка надходить як від користувача, так і з зовнішніх інформаційних джерел. До таких даних належать адреси новинних стрічок, що вводяться користувачем під час оформлення підписки на інформаційні ресурси. Крім того, вхідними даними виступають відомості, які користувач зазначає під час взаємодії із системою, зокрема дані для входу до облікового запису, параметри налаштування відображення новин, запити для пошуку інформації, а також команди, пов'язані з керуванням підписками або збереженням новин у закладках.

Вихідними даними програмної системи для агрегації та оброблення новинних потоків є впорядкована інформація, що відображається користувачеві через інтерфейс програми. До цієї інформації належать заголовки новин, їх короткий опис або повний текст, посилання на першоджерела, а також дані про джерела отримання новинних повідомлень.

4.3.3 Повідомлення

Користувач програмної системи для агрегації та оброблення новинних потоків може отримати такі повідомлення: інформаційні повідомлення, які сигналізують про успішне виконання певних дій у системі, наприклад підтвердження додавання нової підписки, повідомлення про збереження налаштувань або інформування про успішне оновлення новинної стрічки. У випадках, коли виконання дії є неможливим через некоректно введені дані або інші технічні причини, система може формувати повідомлення про помилки, які повідомляють користувача про необхідність перевірити введену інформацію або повторити спробу виконання операції.

4.4 Виконання програмної системи для агрегації та оброблення новинних потоків

Після запуску програмної системи для агрегації та оброблення новинних потоків користувачу відображається сторінка із формою входу до системи (рис. 4.1), у якому передбачено поля для введення облікових даних користувача. У ці поля вводяться логін та пароль, після чого для підтвердження введеної інформації необхідно активувати кнопку входу.

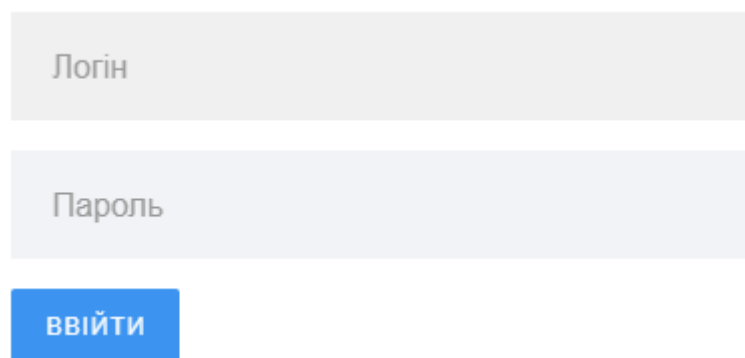


Рисунок 4.1 – Інтерфейс сторінки із формою входу до програмної системи для агрегації та оброблення новинних потоків

Після успішного проходження процедури автентифікації у веббраузері відкривається основний інтерфейс програмної системи (рис. 4.2), де відображається перелік новин, що ще не були переглянуті користувачем. У лівій частині сторінки розміщується основне навігаційне меню. За допомогою цього меню здійснюється доступ до основних функціональних можливостей системи, пов'язаних із переглядом новин, виконанням пошуку, роботою із закладками, аналізом історії переглядів, управлінням підписками та налаштуванням параметрів функціонування програми.

У пункті меню, що відповідає за відображення непрочитаних матеріалів, додатково демонструється інформація про поточну кількість новин, які ще не були відкриті користувачем. Справа розташовано безпосередній перелік новинних повідомлень, отриманих із підписаних

інформаційних джерел. Для кожного елемента такого списку демонструється його заголовок, короткий опис змісту, а також інформаційний рядок із додатковими параметрами. У цьому рядку відображається джерело новини, яке подається у вигляді гіперпосилання на основний сайт, зазначається час, що минув з моменту її оприлюднення, а також доступні дії, які можна виконати з відповідним матеріалом. Серед таких дій передбачено можливість переходу за зовнішнім посиланням для перегляду повного тексту новини, додавання матеріалу до переліку обраних, видалення новини із бази даних або позначення її як прочитаної з подальшим збереженням у системі протягом періоду, визначеного у параметрах налаштування.

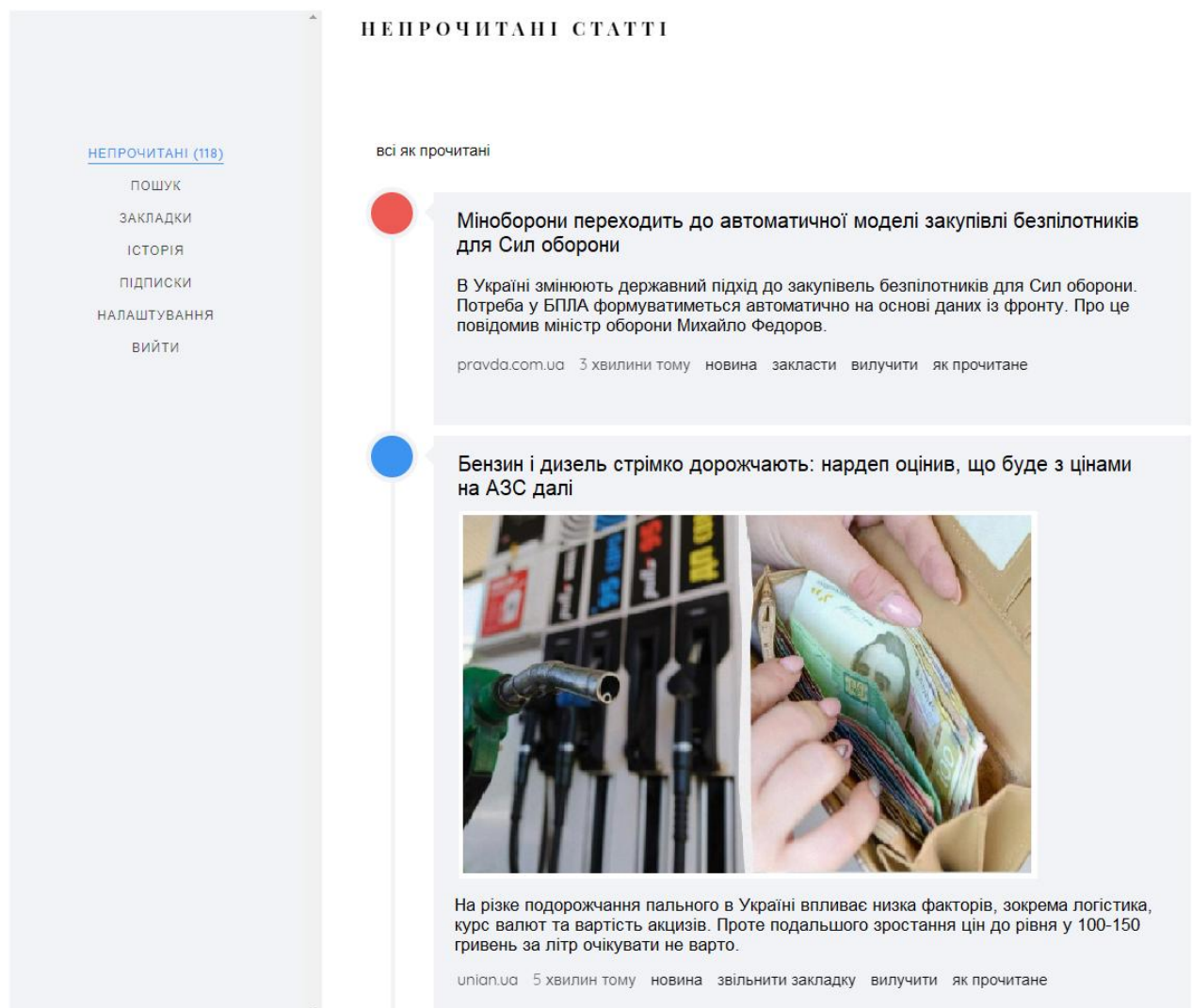


Рисунок 4.2 – Головна сторінка програми

У разі вибору заголовка новини відкривається окрема сторінка (рис. 4.3) , присвячена перегляду відповідного матеріалу. Цей інтерфейс, окрім стандартної верхньої частини сторінки, містить елементи навігації, що дозволяють переходити до попереднього або наступного запису. Такі елементи розташовуються над текстом новини та під ним, забезпечуючи зручне послідовне ознайомлення з матеріалами. На сторінці перегляду відображається назва новини та її зміст.

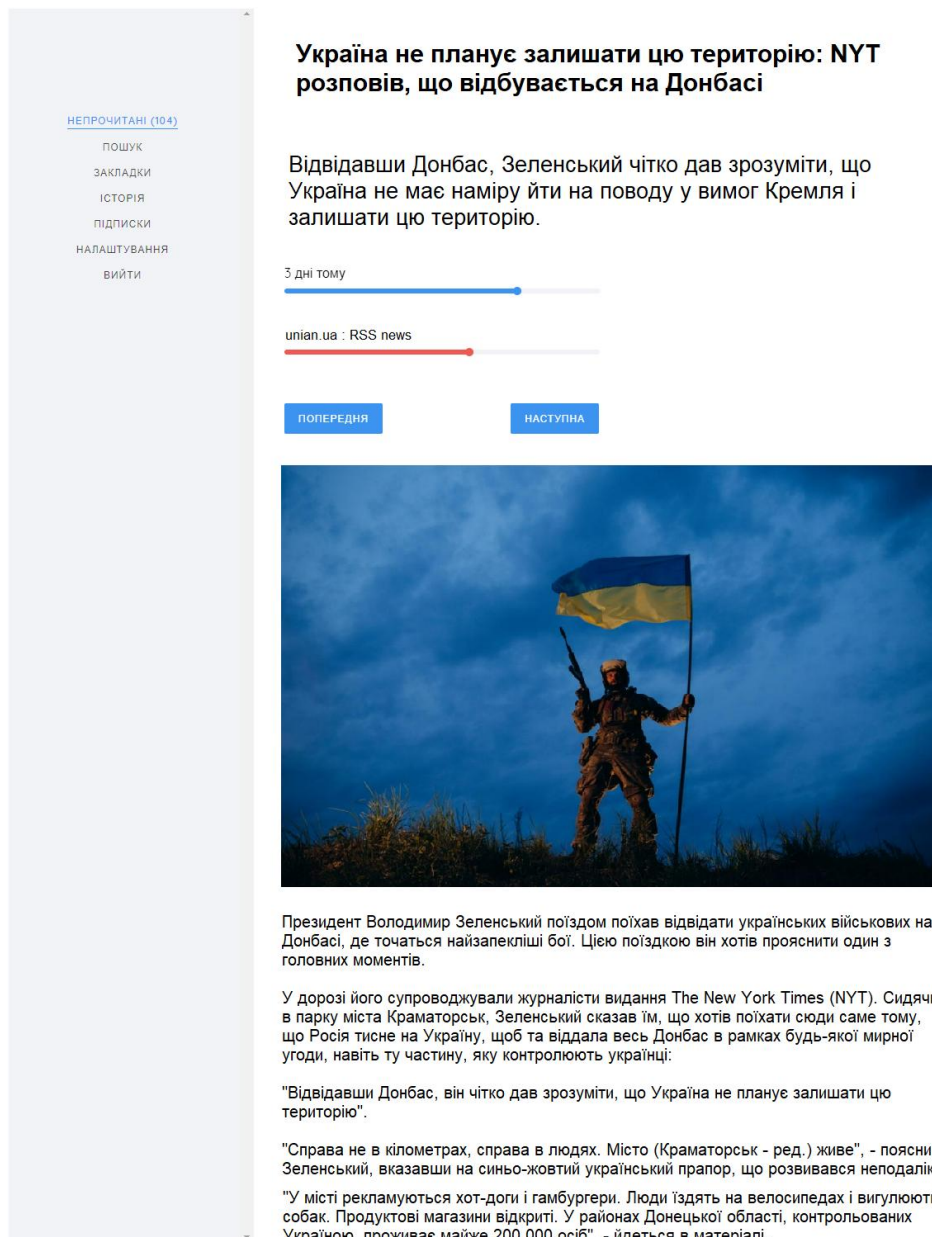


Рисунок 4.3 – Відображення обраної користувачем новини

Окремий розділ системи призначений для керування підписками на інформаційні джерела (рис. 4.4). У межах цього розділу доступне додаткове меню, за допомогою якого можна додавати нові RSS-канали, переглядати перелік уже підключених стрічок, здійснювати імпорт підписок із зовнішніх файлів або виконувати їх збереження у вигляді окремого файлу для подальшого використання. Інтерфейс оформлення нової підписки передбачає відображення спеціальної форми, у якій необхідно зазначити адресу RSS-каналу. Після введення відповідної адреси може бути визначено режим отримання новинної інформації. За необхідності передбачено активацію параметра, який забезпечує завантаження повного тексту новин безпосередньо з вебсайту джерела, а не лише інформації, що міститься у самій RSS-стрічці.

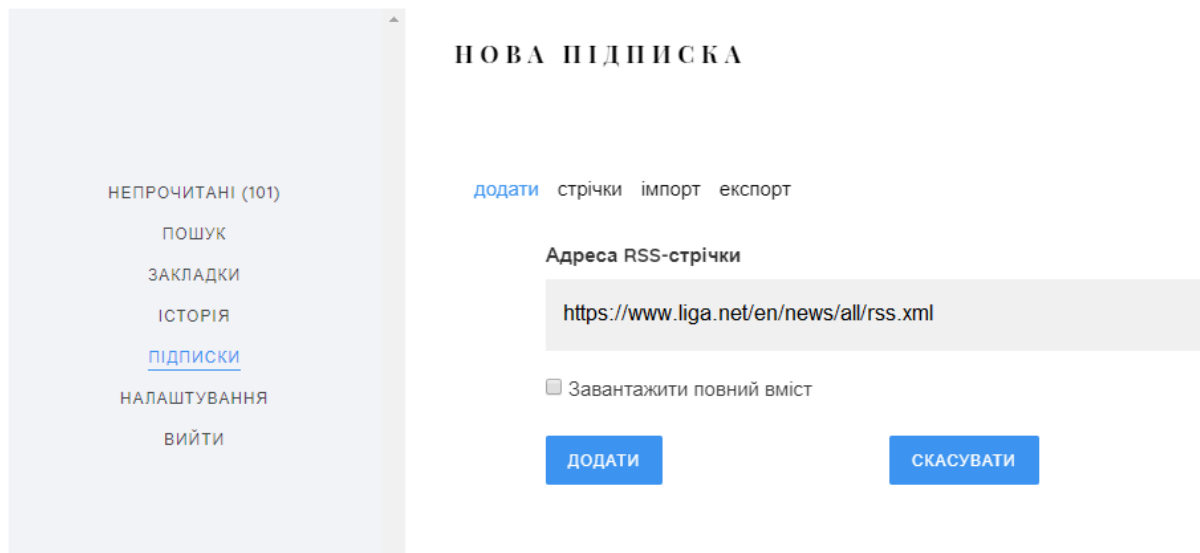


Рисунок 4.4 – Додавання RSS-каналу для оброблення новин

У разі переходу до розділу перегляду стрічок відображається перелік усіх RSS-каналів, на які виконано підписку (рис. 4.5). Для кожного джерела показується його назва, а поруч зазначається інформація про кількість новин, що залишаються непрочитаними, а також загальна кількість матеріалів, доступних у стрічці. Додатково наводяться відомості про час останньої перевірки відповідного інформаційного джерела. Під цими даними розташовується панель керування, яка забезпечує швидкий доступ до вебсайту

джерела, дозволяє ініціювати ручне оновлення стрічки або перейти до інтерфейсу редагування параметрів підписки.

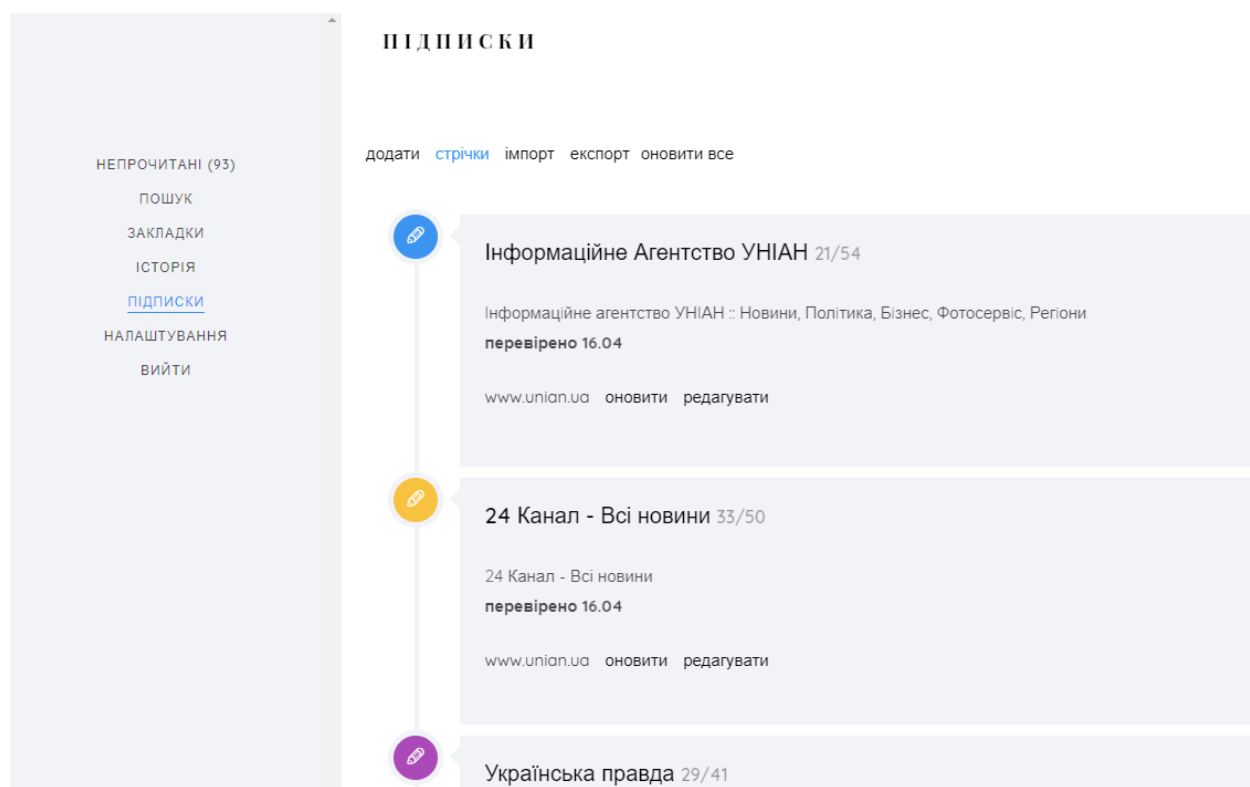
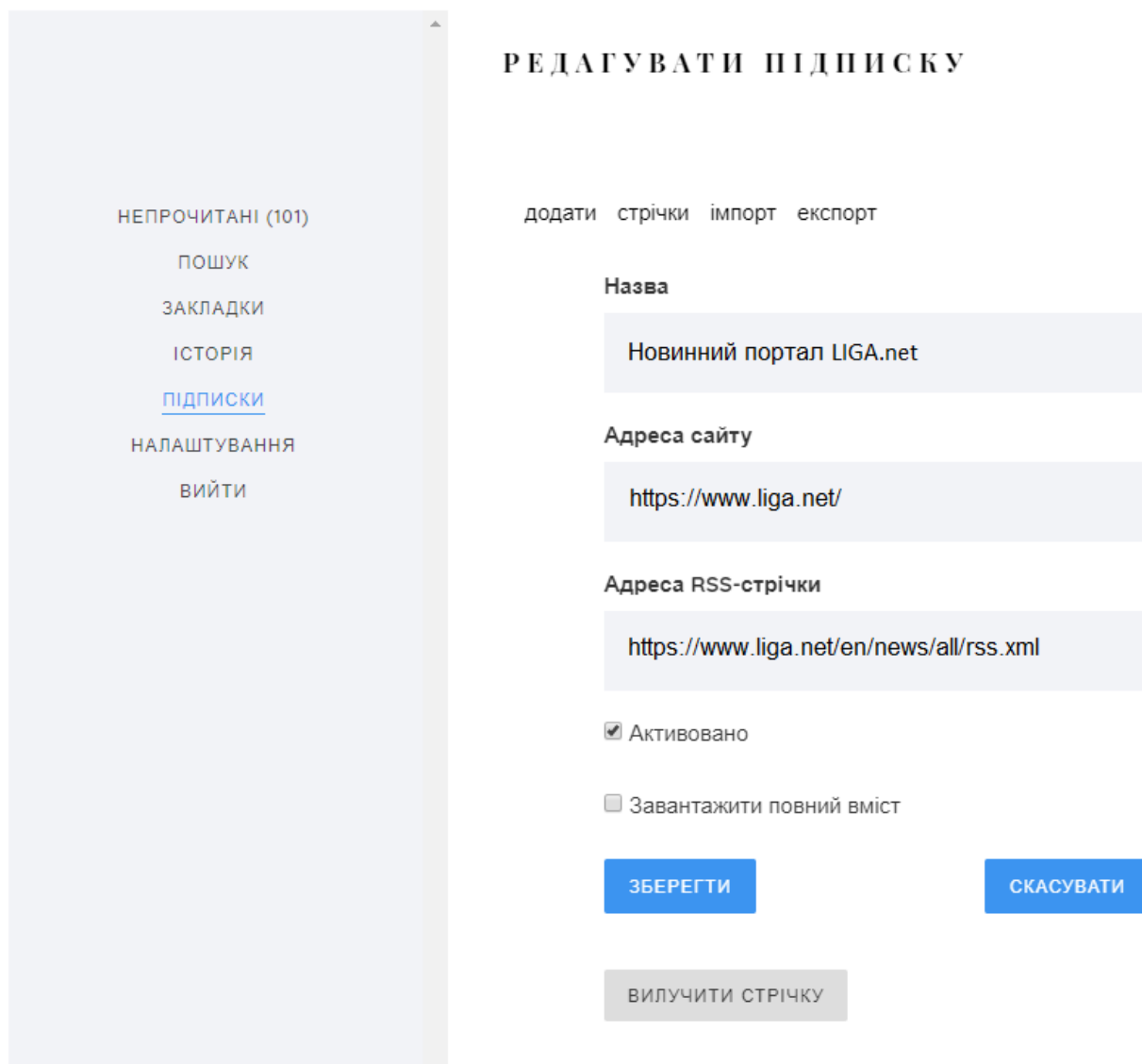


Рисунок 4.5 – Підписки користувача на новинні агентства

Під час редагування підписки відкривається форма (рис. 4.6), у якій можна змінити назву джерела, адресу вебсайту та безпосередню адресу RSS-каналу. У цьому ж інтерфейсі визначається режим завантаження повного тексту новин та стан активності підписки, що впливає на те, чи буде система здійснювати перевірку відповідної стрічки. Після внесення змін передбачено можливість їх збереження або скасування. У разі потреби доступна функція повного видалення відповідного інформаційного джерела зі списку підписок.



РЕДАГУВАТИ ПІДПИСКУ

додати стрічки імпорт експорт

Назва

Новинний портал LIGA.net

Адреса сайту

https://www.liga.net/

Адреса RSS-стрічки

https://www.liga.net/en/news/all/rss.xml

☒ Активовано

☐ Завантажити повний вміст

ЗБЕРЕГТИ

СКАСУВАТИ

ВИЛУЧИТИ СТРИЧКУ

Рисунок 4.6 – Редагування інформації про підписку на RSS-канал

У розділі Налаштувань (рис. 4.7) визначаються правила зберігання новинних матеріалів у базі даних, зокрема тривалість зберігання прочитаних і непрочитаних записів, кількість новин, які відображаються в межах однієї сторінки, а також інтервал оновлення інформації про кількість непрочитаних матеріалів. Окрім цього передбачено можливість зміни пароля користувача та виконання процедури оновлення вебзастосунку з визначеного зовнішнього сервера, що дозволяє підтримувати програмне забезпечення у актуальному стані.

НЕПРОЧИТАНІ (101)

ПОШУК

ЗАКЛАДКИ

ІСТОРІЯ

ПІДПИСКИ

НАЛАШТУВАННЯ

ВИЙТИ

НАЛАШТУВАННЯ

[загальні](#) база даних

Читання

Вилучати прочитані новини

Після 10 днів

Вилучати непрочитані новини

Після 30 днів

Новин на сторінці

10

Оновлювати лічильник новин

5

Аутентифікація

Логін

Пароль

Повторити пароль

ЗБЕРЕГТИ

СКАСУВАТИ

Рисунок 4.7 – Налаштування користувача

4.5 Тестування програмної системи для агрегації та оброблення новинних потоків

Проведено тестування програмної системи для агрегації та оброблення новинних потоків.

Підтверджено, що програмної системи для агрегації та оброблення новинних потоків працює коректно і стабільно, виконуючи всі необхідні функціональні вимоги та успішно справляючись із основною задачею підтримки процесу роботи з новинами.

Розроблена система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів агрегації та оброблення новинних потоків.

4.6 Висновки за розділом 4

Описано програмну систему для агрегації та оброблення новинних потоків.

Проведено тестування створеної програмної системи для агрегації та оброблення новинних потоків. Результати тестування показали, що система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів агрегації та оброблення новинних потоків.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процеси, пов'язані з організацією та опрацюванням даних для агрегації та оброблення новинних потоків.

Визначено, що програмна система для агрегації та оброблення новинних потоків є спеціалізованим засобом, призначеним для автоматизованого отримання, накопичення, впорядкування та аналізу інформаційних повідомлень, що надходять із різних мережесих джерел. У межах такого програмного рішення забезпечується централізоване збирання новинних матеріалів, публікацій, повідомлень інформаційних порталів, блогів та інших вебресурсів. У результаті формується єдиний інформаційний простір, у якому всі отримані матеріали відображаються у впорядкованому вигляді та можуть бути переглянуті, відсортовані або проаналізовані за певними критеріями. Використання подібних програмних систем дозволяє суттєво спростити процес відстеження великої кількості інформаційних джерел і підвищити ефективність роботи з інформаційними потоками.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних систем для агрегації та оброблення новинних потоків. Проте використання деяких програмних систем для агрегації та оброблення новинних потоків пов'язано ризиком формування інформаційної однобічності. Алгоритми відбору новин можуть орієнтуватися на певні критерії популярності або частоти згадування тем, що призводить до домінування окремих джерел або тематик. У результаті користувач може отримувати обмежений спектр інформації, тоді як менш популярні, але потенційно важливі повідомлення залишаються поза увагою. Така ситуація може впливати на об'єктивність інформаційної картини, що формується у межах системи. Не менш важливим аспектом є технічні та ресурсні витрати, пов'язані з підтримкою функціонування подібних програмних систем. Збирання та оброблення великої кількості новинних повідомлень потребує

значних обчислювальних ресурсів, ефективних механізмів зберігання даних і постійного оновлення програмних компонентів. У разі масштабування системи, коли кількість джерел та обсяг інформації постійно зростають, підвищуються вимоги до продуктивності серверної інфраструктури та оптимізації алгоритмів оброблення інформації. Тому актуальною є розробка програмної системи для агрегації та оброблення новинних потоків.

Сформульовано функціональні вимоги до програмної системи для агрегації та оброблення новинних потоків.

Для реалізації програмної системи для агрегації та оброблення новинних потоків обрано мову програмування PHP, яка забезпечує ефективну інтеграцію з вебтехнологіями, наявність засобів для отримання інформації з мережеских джерел, зручну взаємодію з базами даних, простоту реалізації серверної логіки та широке поширення у середовищі вебінфраструктури. Сукупність цих характеристик мови PHP створює сприятливі умови для розроблення функціональної, масштабованої та зручної у підтримці програмної системи, здатної ефективно працювати з великими потоками новинної інформації.

Для створення програмної системи для агрегації та оброблення новинних потоків обрано середовище розробки Visual Studio Code через поєднання функціональності, гнучкості налаштувань, високої продуктивності та широких можливостей інтеграції з сучасними інструментами веброзроблення. Це середовище дозволяє організувати ефективний процес створення програмного забезпечення, забезпечує зручну роботу з програмним кодом та сприяє підвищенню якості кінцевого програмного продукту.

Для зберігання та опрацювання даних обрано систему управління базами даних SQLite, яка дозволяє ефективно організувати зберігання текстової інформації, забезпечити швидкий доступ до новинних матеріалів і створити стабільну основу для функціонування програмного застосунку, орієнтованого на оброблення великих обсягів інформаційних повідомлень.

Розроблено програмну систему для агрегації та оброблення новинних потоків.

Запропоновано структуру програмної системи для агрегації та оброблення новинних потоків. При розробці структури програми враховано принципи модульності та розподілу функціональних обов'язків між окремими частинами програмної системи, що дозволяє забезпечити зрозумілу логіку взаємодії між програмними модулями, підвищити гнучкість програмного коду та спростити подальше розширення функціональності системи. Архітектурною основою розроблення програмної системи для агрегації та оброблення новинних потоків слугувала концепція Model–View–Controller (MVC), відповідно до якої функції зберігання даних, оброблення інформації та відображення результатів розділяються між різними компонентами програмного забезпечення.

Описано функціонування програмної системи для агрегації та оброблення новинних потоків, що базується на поєднанні механізмів автоматичного отримання інформації з RSS-джерел та зручних інструментів взаємодії користувача з новинними даними. Реалізовані режими роботи дозволяють ефективно переглядати, шукати, зберігати та впорядковувати новинні повідомлення, а також керувати джерелами їх отримання. Така організація роботи забезпечує зручне використання системи та сприяє більш ефективному опрацюванню інформаційних потоків.

Розроблено базу даних програмної системи для агрегації та оброблення новинних потоків, яка побудована на основі логічно взаємопов'язаних таблиць, що відображають основні елементи роботи з новинними потоками. Вона забезпечує зберігання інформації про новини, джерела їх отримання, користувачів та параметри конфігурації системи, а також реалізує необхідні зв'язки між цими елементами. Розроблена база даних дозволяє ефективно організувати накопичення інформації, забезпечує можливість швидкого доступу до новинних матеріалів та підтримує гнучку систему налаштувань для

користувачів, що сприяє стабільній та зручній роботі програмного забезпечення

Виконано проєктування інтерфейсу взаємодії користувача з програмною системою для агрегації та оброблення новинних потоків.

Проведено тестування створеної програмної системи для агрегації та оброблення новинних потоків. Результати тестування показали, що система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів агрегації та оброблення новинних потоків.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is News Aggregation? Understanding Its Importance. URL: <https://ultra.news/2025/09/03/what-is-news-aggregation-understanding-its-importance/> (date of access: 04.05.2026).
2. News Aggregator History, Facts & Types | What is News Aggregation. URL: <https://study.com/academy/lesson/news-aggregator-history-facts-types-aggregation.html> (date of access: 04.05.2026).
3. Role of News Aggregator Smartest Way To Consume News. URL: <https://newsdata.io/blog/role-of-news-aggregator/> (date of access: 04.05.2026).
4. News Aggregators Definition & Detailed Explanation. URL: <https://simplyglobalmedia.com/media-distribution-channels-glossary/news-aggregators/> (date of access: 04.05.2026).
5. News aggregators. URL: <https://fiveable.me/mass-media-society/key-terms/news-aggregators> (date of access: 04.05.2026).
6. Navigating the Complex World of Aggregators and Algorithms in News. URL: <https://journalism.university/reporting-techniques/news-aggregators-algorithms-complex-navigation/> (date of access: 04.05.2026).
7. Best RSS Feed Reader And News Aggregation Apps. URL: <https://contentstudio.io/blog/rss-feed-reader-apps> (date of access: 04.05.2026).
8. Mastering the Stream: The Ultimate Guide to RSS Feed News Aggregators. URL: <https://rssdailynews.com/2024/03/13/ultimate-guide-to-rss-feed-news-aggregators/> (date of access: 04.05.2026).
9. Top Free Windows RSS Feed Readers News Aggregators. URL: <https://www.lifewire.com/free-windows-rss-feed-readers-1173954> (date of access: 04.05.2026).
10. Top RSS Aggregators: Features, Pros, Cons & Comparison. URL: <https://www.bestdevops.com/top-10-rss-aggregators-features-pros-cons-comparison/> (date of access: 04.05.2026).

11. Top rss-aggregator Open-Source Projects. URL: <https://www.libhunt.com/topic/rss-aggregator> (date of access: 04.05.2026).
12. Feedspot. URL: <https://www.feedspot.com/> (date of access: 04.05.2026).
13. Feedly. URL: <https://feedly.com/news-reader> (date of access: 04.05.2026).
14. RSSOwl. URL: <https://www.rssowl.org/> (date of access: 04.05.2026).
15. PHP Tutorial. URL: <https://www.geeksforgeeks.org/php/php-tutorial/> (date of access: 04.05.2026).
16. Manual PHP. URL: <https://www.manualweb.net/php/> (date of access: 04.05.2026).
17. Visual Studio Code Documentation. URL: <https://code.visualstudio.com/docs> (date of access: 04.05.2026).
18. Visual Studio Code. URL: <https://www.freecodecamp.org/news/tag/visual-studio-code/> (date of access: 04.05.2026).
19. SQLite Tutorial. URL: <https://www.geeksforgeeks.org/sqlite/sqlite-tutorial/> (date of access: 04.05.2026).
20. SQLite Tutorial. URL: <https://www.sqlitetutor.com/> (date of access: 04.05.2026).

ДОДАТОК А
Технічне завдання

Вступ

Програмна система може використовуватися для підтримки процесу оброблення новинних потоків.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Створення програмної системи для агрегування та опрацювання новинних потоків», затверджене наказом Національного університету «Запорізька політехніка» № 139 від 7 квітня 2026 р.

А.2 Призначення розробки

Програмна система призначена для агрегації та оброблення новинних потоків.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Програмна система для агрегації та оброблення новинних потоків забезпечує виконання таких функцій:

- підтримка можливості отримання новин із зовнішніх інформаційних ресурсів на основі новинних стрічок;
- можливість відображення новин у зручному для користувача вигляді через вебінтерфейс;
- можливість реєстрації та авторизації користувачів у програмній системі;
- підтримка можливості створення та керування підписками на новинні стрічки;

- підтримка можливості додавання нових підписок на новинні ресурси;
- підтримка можливості перегляду новин, що ще не були прочитані користувачем;
- можливість налаштування параметрів відображення та зберігання новин у системі;
- підтримка можливості автоматичного оновлення новинних стрічок.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмної системи для агрегації та оброблення новинних потоків повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмна система для агрегації та оброблення новинних потоків повинна забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмної системи для агрегації та оброблення новинних потоків необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічних засобів

Для роботи програмної системи для агрегації та оброблення новинних потоків потрібні такі технічні та програмні ресурси: центральний процесор повинен мати тактову частоту не нижче 2,0 ГГц, що дозволяє забезпечити

стабільне виконання програмного коду, оброблення запитів користувачів і формування динамічних вебсторінок. Оперативна пам'ять комп'ютера повинна становити не менше 4 ГБ, оскільки під час функціонування програмної системи одночасно виконуються процеси оброблення RSS-стрічок, формування запитів до бази даних та відображення інформації у веб-інтерфейсі. Для зберігання програмних файлів, бази даних і допоміжних ресурсів необхідно передбачити наявність вільного дискового простору не менше 1 ГБ. Щодо програмного забезпечення, то функціонування системи передбачає наявність операційної системи, яка підтримує роботу вебсерверів і середовищ виконання сценаріїв. Для цього можуть використовуватися сучасні операційні системи, такі як Microsoft Windows, Ubuntu або інші дистрибутиви на базі Linux. Використання таких операційних систем забезпечує стабільне середовище для запуску серверних компонентів і програмних модулів системи.

Для забезпечення роботи вебчастини програми необхідно встановити вебсервер, який оброблятиме HTTP-запити користувачів. Оскільки логіка роботи програмної системи реалізується за допомогою серверних сценаріїв, необхідною умовою є встановлення середовища виконання мови програмування PHP версії не нижче 7.4. Використання сучасної версії мови забезпечує підтримку актуальних можливостей програмування, підвищений рівень безпеки та кращу продуктивність під час виконання програмного коду.

Для зберігання та оброблення інформації про новини, підписки та користувачів застосовується система керування базами даних SQLite. Вона не потребує встановлення окремого серверного компонента та працює безпосередньо з файлом бази даних, що спрощує розгортання програмної системи та зменшує вимоги до системних ресурсів.

Для взаємодії користувача з програмною системою необхідний сучасний веббраузер, який підтримує роботу з HTML-розміткою, каскадними таблицями стилів та клієнтськими сценаріями. Для цієї мети можуть

використовуватися такі браузери, як Google Chrome, Mozilla Firefox або Microsoft Edge.

А.3.6 Вимоги до маркування і пакування

Програмна система для агрегації та оброблення новинних потоків може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва – «Програмна система для агрегації та оброблення новинних потоків».

А.4 Вхідні дані до роботи

Вхідними даними до програмної системи для агрегації та оброблення новинних потоків є інформація, яка надходить як від користувача, так і з зовнішніх інформаційних джерел. До таких даних належать адреси новинних стрічок, що вводяться користувачем під час оформлення підписки на інформаційні ресурси. Крім того, вхідними даними виступають відомості, які користувач зазначає під час взаємодії із системою, зокрема дані для входу до облікового запису, параметри налаштування відображення новин, запити для пошуку інформації, а також команди, пов'язані з керуванням підписками або збереженням новин у закладках.

Вихідними даними програмної системи для агрегації та оброблення новинних потоків є впорядкована інформація, що відображається користувачеві через інтерфейс програми. До цієї інформації належать заголовки новин, їх короткий опис або повний текст, посилання на першоджерела, а також дані про джерела отримання новинних повідомлень.

ДОДАТОК Б
Фрагмент тексту програми

```

$pth_dt = __DIR__ . "/dt";

if(!file_exists($pth_dt)){
    mkdir($pth_dt,0777,true);
}

$pth_db = $pth_dt . "/nws.db";

function cnct_db($pth_db)
{
    $dbh = new PDO("sqlite:" . $pth_db);
    $dbh->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
    $dbh->exec("PRAGMA journal_mode=WAL");
    $dbh->exec("PRAGMA synchronous=NORMAL");
    $dbh->exec("PRAGMA temp_store=MEMORY");
    return $dbh;
}

$dbh = cnct_db($pth_db);

function db_nd($dbh,$sql,$prm)
{
    $stm=$dbh->prepare($sql);
    foreach($prm as $k=>$v){
        $stm->bindValue($k,$v);
    }
    $stm->execute();
}

function db_gt($dbh,$sql,$prm)
{
    $stm=$dbh->prepare($sql);

    foreach($prm as $k=>$v){
        $stm->bindValue($k,$v);
    }

    $stm->execute();

    $rs=$stm->fetchAll(PDO::FETCH_ASSOC);

    if(!$rs){
        return [];
    }

    return $rs;
}

```

```

function db_gt_rw($dbh,$sql,$prm)
{
    $stm=$dbh->prepare($sql);

    foreach($prm as $k=>$v){
        $stm->bindValue($k,$v);
    }

    $stm->execute();

    $rw=$stm->fetch(PDO::FETCH_ASSOC);

    if(!$rw){
        return [];
    }

    return $rw;
}

function db_rm($dbh,$sql,$prm)
{
    $stm=$dbh->prepare($sql);

    foreach($prm as $k=>$v){
        $stm->bindValue($k,$v);
    }

    $stm->execute();
}

function db_up($dbh,$sql,$prm)
{
    $stm=$dbh->prepare($sql);

    foreach($prm as $k=>$v){
        $stm->bindValue($k,$v);
    }

    $stm->execute();
}

function fch_rss($lnk)
{
    $ctx=stream_context_create([
        "http"=>[
            "timeout">15,
            "header"=>"User-Agent: rss-ngg"
        ]
    ])

```

```

    ]);

    $dt=file_get_contents($lnk,false,$ctx);

    if($dt===false){
        return "";
    }

    return $dt;
}

function chk_rss($dt)
{
    if(strlen($dt)<10){
        return false;
    }

    if(strpos($dt,"<rss")===false
    strpos($dt,"<feed")===false){
        return false;
    }

    return true;
}

function fch_all_sbosc($dbh)
{
    $sql="SELECT * FROM sbosc";

    $rs=db_get($dbh,$sql,[]);

    if(!$rs){
        return [];
    }

    return $rs;
}

function fch_rss_frm_sbosc($dbh)
{
    $sbosc=fch_all_sbosc($dbh);

    $lst=[];

    foreach($sbosc as $rw){

        $lnk=$rw["lnk"];

        $dt=fch_rss($lnk);
    }
}

```

```

        if(!chk_rss($dt)){
            continue;
        }

        $lst[]=[
            "sbsc"=>$rw,
            "rss"=>$dt
        ];
    }

    return $lst;
}

function prs_rss($dt)
{
    $xml=@simplexml_load_string($dt);

    if(!$xml){
        return [];
    }

    $lst=[];

    if(isset($xml->channel)){

        foreach($xml->channel->item as $it){

            $ttl=(string)$it->title;
            $lnk=(string)$it->link;
            $txt=(string)$it->description;

            $lst[]=[
                "ttl"=>$ttl,
                "lnk"=>$lnk,
                "txt"=>$txt
            ];
        }
    }

    if(isset($xml->entry)){

        foreach($xml->entry as $it){

            $ttl=(string)$it->title;
            $lnk=(string)$it->link['href'];
            $txt=(string)$it->summary;

            $lst=[]=

```

```

        "ttl"=>$ttl,
        "lnk"=>$lnk,
        "txt"=>$txt
    ];
}
}

return $lst;
}

function prs_all_rss($dbh)
{
    $rss_lst=fch_rss_frm_sbcs($dbh);

    $nws_lst=[];

    foreach($rss_lst as $rw){

        $sbcs=$rw["sbcs"];
        $rss=$rw["rss"];

        $prs=prs_rss($rss);

        foreach($prs as $it){

            $nws_lst[]= [
                "sbcs_d"=>$sbcs["d"],
                "ttl"=>$it["ttl"],
                "lnk"=>$it["lnk"],
                "txt"=>$it["txt"]
            ];
        }
    }

    return $nws_lst;
}

function sv_nws($dbh,$nws_lst)
{
    foreach($nws_lst as $it){

        $sql="SELECT d FROM nws WHERE lnk=:lnk";

        $rw=db_gt_rw($dbh,$sql,["lnk"=>$it["lnk"]]);

        if($rw){
            continue;
        }
    }
}

```

```

        $sql2="INSERT INTO nws(sbsc_d,ttl,lnk,txt,rd,bkm)
VALUES(:sbsc_d,:ttl,:lnk,:txt,','')";

        db_nd($dbh,$sql2,[
            ":sbsc_d"=>$it["sbsc_d"],
            ":ttl"=>$it["ttl"],
            ":lnk"=>$it["lnk"],
            ":txt"=>$it["txt"]
        ]);
    }
}

function gt_nws_ll($dbh)
{
    $sql="SELECT * FROM nws ORDER BY d DESC";
    return db_gt($dbh,$sql,[]);
}

function gt_nws_nr($dbh)
{
    $sql="SELECT * FROM nws WHERE rd=' ' ORDER BY d DESC";
    return db_gt($dbh,$sql,[]);
}

function gt_nws_bkm($dbh)
{
    $sql="SELECT * FROM nws WHERE bkm='1' ORDER BY d DESC";
    return db_gt($dbh,$sql,[]);
}

function rd_nws($dbh,$d)
{
    $sql="UPDATE nws SET rd='1' WHERE d=:d";
    db_up($dbh,$sql,[":d"=>$d]);

    $sql2="INSERT INTO hst(nws_d) VALUES(:d)";
    db_nd($dbh,$sql2,[":d"=>$d]);
}

function tg_bkm($dbh,$d)
{
    $sql="SELECT bkm FROM nws WHERE d=:d";
    $rw=db_gt_rw($dbh,$sql,[":d"=>$d]);

    if(!$rw){
        return;
    }

    $b=$rw["bkm"];

```

```

    if ($b=="1") {
        $sql2="UPDATE nws SET bkm='' WHERE d=:d";
        db_up($dbh,$sql2,[":d"=>$d]);
    }else{
        $sql3="UPDATE nws SET bkm='1' WHERE d=:d";
        db_up($dbh,$sql3,[":d"=>$d]);
    }
}

function rm_nws($dbh,$d)
{
    $sql="DELETE FROM nws WHERE d=:d";
    db_rm($dbh,$sql,[":d"=>$d]);
}

function cln_nws($dbh)
{
    $sql="SELECT d FROM nws";
    $rs=db_gt($dbh,$sql,[]);

    foreach($rs as $rw){
        if(strlen($rw["d"])>0){
            continue;
        }
    }
}

function gt_sbse($dbh)
{
    $sql="SELECT * FROM sbse ORDER BY d DESC";
    return db_gt($dbh,$sql,[]);
}

function dd_sbse($dbh,$ttl,$lnk)
{
    $sql="INSERT INTO sbse(ttl,lnk,lst) VALUES(:ttl,:lnk,'')";
    db_nd($dbh,$sql,[
        ":ttl"=>$ttl,
        ":lnk"=>$lnk
    ]);
}

function rm_sbse($dbh,$d)
{
    $sql="DELETE FROM sbse WHERE d=:d";
    db_rm($dbh,$sql,[":d"=>$d]);
}

```

```

function gt_sbsec_rw($dbh,$d)
{
    $sql="SELECT * FROM sbsec WHERE d=:d";
    return db_gt_rw($dbh,$sql,[":d"=>$d]);
}

function md_sbsec($dbh,$d,$ttl,$lnk)
{
    $sql="UPDATE sbsec SET ttl=:ttl,lnk=:lnk WHERE d=:d";
    db_up($dbh,$sql,[
        ":ttl"=>$ttl,
        ":lnk"=>$lnk,
        ":d"=>$d
    ]);
}

function srch_nws($dbh,$trm)
{
    $sql="SELECT * FROM nws WHERE ttl LIKE :trm OR txt LIKE :trm
ORDER BY d DESC";

    $rs=db_gt($dbh,$sql,[":trm"=>"%".$trm."%"]);

    $sql2="INSERT INTO srch(trm) VALUES(:trm)";
    db_nd($dbh,$sql2,[":trm"=>$trm]);

    return $rs;
}

function gt_srch_hst($dbh)
{
    $sql="SELECT * FROM srch ORDER BY d DESC";
    return db_gt($dbh,$sql,[]);
}

function mk_bkm($dbh,$d)
{
    $sql="UPDATE nws SET bkm='1' WHERE d=:d";
    db_up($dbh,$sql,[":d"=>$d]);
}

function rm_bkm($dbh,$d)
{
    $sql="UPDATE nws SET bkm='' WHERE d=:d";
    db_up($dbh,$sql,[":d"=>$d]);
}

function gt_bkm($dbh)
{

```

```

        $sql="SELECT * FROM nws WHERE bkm='1'";
        return db_gt($dbh,$sql,[]);
    }

function gt_hst($dbh)
{
    $sql="SELECT nws.* FROM nws
    JOIN hst ON hst.nws_d=nws.d
    ORDER BY hst.d DESC";

    return db_gt($dbh,$sql,[]);
}

function clr_hst($dbh)
{
    $sql="DELETE FROM hst";
    db_rm($dbh,$sql,[]);
}

function rt_rq($dbh)
{
    $md=$_SERVER["REQUEST_METHOD"];

    if($md=="GET"){

        if(isset($_GET["gt_nws"])){
            return ctrl_gt_nws($dbh);
        }

        if(isset($_GET["gt_nr"])){
            return ctrl_gt_nr($dbh);
        }

        if(isset($_GET["gt_bkm"])){
            return ctrl_gt_bkm($dbh);
        }

        if(isset($_GET["gt_sbse"])){
            return ctrl_gt_sbse($dbh);
        }

        if(isset($_GET["srch"])){
            $trm=$_GET["srch"];
            return ctrl_srch($dbh,$trm);
        }

        if(isset($_GET["hst"])){
            return gt_hst($dbh);
        }
    }
}

```

```

}

if ($md=="POST") {

    if (isset($_POST["rd"])) {
        ctrl_rd($dbh,$_POST["rd"]);
    }

    if (isset($_POST["bkm"])) {
        ctrl_bkm($dbh,$_POST["bkm"]);
    }

    if (isset($_POST["rm"])) {
        ctrl_rm($dbh,$_POST["rm"]);
    }

    if (isset($_POST["dd_sbse"])) {

        $ttl=$_POST["ttl"];
        $lnk=$_POST["lnk"];

        ctrl_dd_sbse($dbh,$ttl,$lnk);
    }

    if (isset($_POST["rm_sbse"])) {
        ctrl_rm_sbse($dbh,$_POST["rm_sbse"]);
    }
}

return [];
}

```

ДОДАТОК В
Слайди презентації

Створення програмної системи для агрегування та опрацювання новинних потоків

Виконав	Олександр МАКСИМЕНКО ст.гр. КНТ-212
Керівник	Тетяна ЗАЙКО к.т.н., доцент

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процеси, пов'язані з організацією та опрацюванням даних для агрегації та оброблення новинних потоків.

Предмет дослідження – програмні системи для агрегації та оброблення новинних потоків.

Мета роботи – розробка програмної системи для агрегації та оброблення новинних потоків для зменшення часу дослідження новин.

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних систем для агрегації та оброблення новинних потоків;
- здійснити проєктування програмної системи для агрегації та оброблення новинних потоків;
- створити програмну систему для агрегації та оброблення новинних потоків;
- дослідити розроблену програмну систему для агрегації та оброблення новинних потоків.

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Feedspot	Feedly	RSSOwl
агрегування великої кількості джерел	+	+	+—
інтелектуальна фільтрація контенту	+—	+	—
автоматичне відстеження нових джерел	+	+—	—
розширений пошук інформації	+	+	+—
підтримка командної роботи	+—	+	—
інтеграція з іншими сервісами	+—	+	+—
персоналізація стрічок	+—	+	+

Рисунок В.4 – Слайд 4

Постановка завдання

При розробці програмної системи для агрегації та оброблення новинних потоків необхідно забезпечити такі функціональні вимоги:

- підтримка можливості отримання новин із зовнішніх інформаційних ресурсів на основі новинних стрічок;
- можливість відображення новин у зручному для користувача вигляді через вебінтерфейс;
- можливість реєстрації та авторизації користувачів у програмній системі;
- підтримка можливості створення та керування підписками на новинні стрічки;
- підтримка можливості додавання нових підписок на новинні ресурси;
- підтримка можливості перегляду новин, що ще не були прочитані користувачем;
- можливість налаштування параметрів відображення та зберігання новин у системі;
- підтримка можливості автоматичного оновлення новинних стрічок.

Рисунок В.5 – Слайд 5

Вибір мови програмування

Критерій порівняння мов програмування	Мова програмування		
	PHP	Python	C#
Орієнтація на веброботку	+	+-	+-
Простота розгортання на вебсервері	+	+-	-
Наявність вбудованих засобів роботи з HTTP та вебзапитами	+	+-	+-
Простота інтеграції з вебсторінками	+	+-	-
Зручність роботи з базами даних	+	+	+-
Швидкість створення веборієнтованих систем	+	+	-

Рисунок В.6 – Слайд 6

Вибір середовища розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio Code	NetBeans	Eclipse
Простота встановлення та налаштування	+	+—	—
Швидкість роботи середовища	+	+—	+—
Зручність роботи з вебтехнологіями	+	+	+—
Наявність великої кількості розширень	+	+—	+
Зручність навігації по проекту	+	+—	+—

Рисунок В.7 – Слайд 7

Вибір засобів зберігання та опрацювання даних

Критерій порівняння систем управління базами даних	Системи управління базами даних		
	SQLite	MySQL	PostgreSQL
Відсутність необхідності окремого серверу	+	—	—
Компактність та легкість розгортання	+	+—	+—
Швидкість роботи з невеликими та середніми обсягами даних	+	+—	+—
Низькі вимоги до апаратних ресурсів	+	+—	—
Легкість резервного копіювання та переносу	+	+—	+—

Рисунок В.8 – Слайд 8

Структура програми

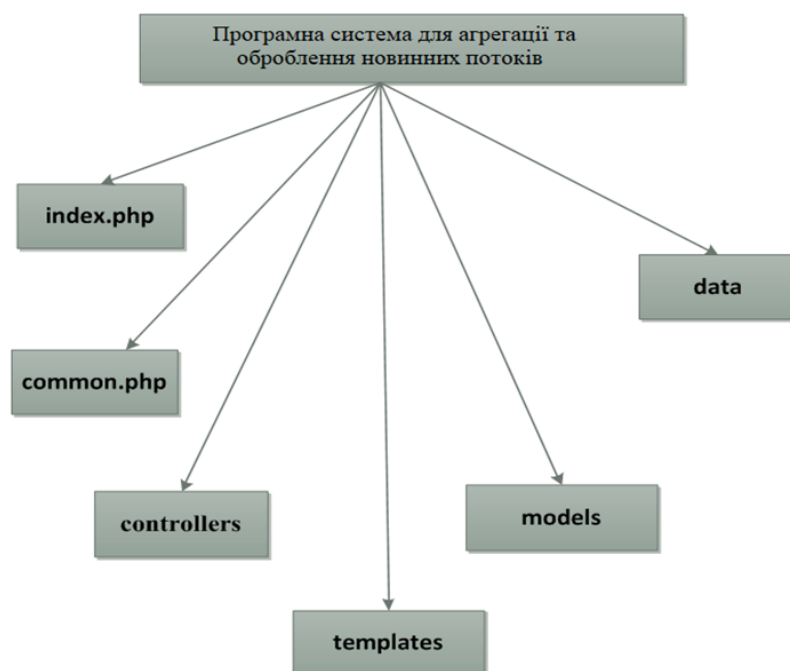


Рисунок В.9 – Слайд 9

Функціонування програми

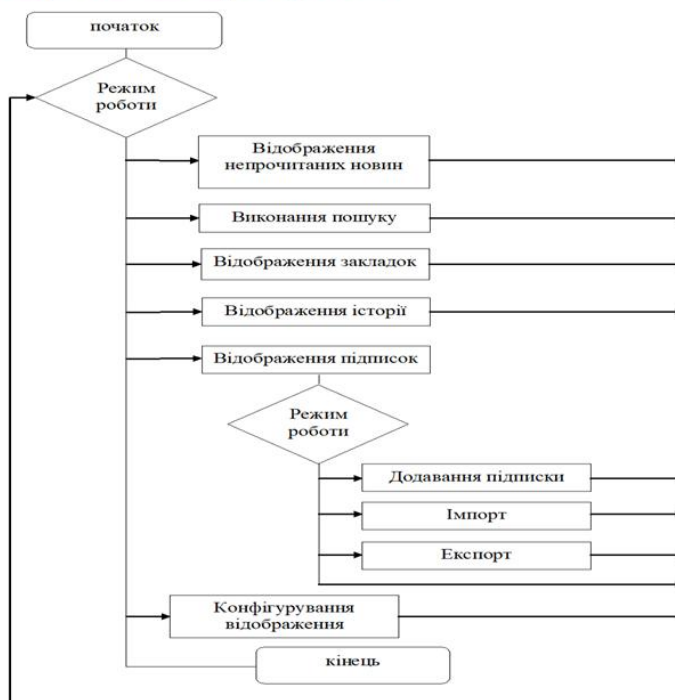


Рисунок В.10 – Слайд 10

Схема бази даних

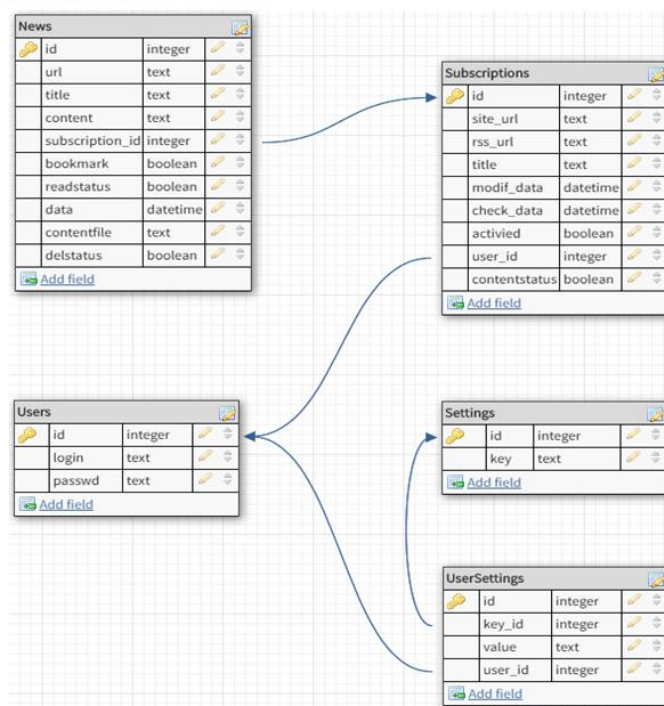


Рисунок В.11 – Слайд 11

Робота з програмою. Відображення новин

НЕПРОЧИТАНІ (118)

пошук

закладки

історія

підписки

налаштування

вийти

НЕПРОЧИТАНІ СТАТТІ

всі як прочитані

Міноборони переходить до автоматичної моделі закупівлі безпілотників для Сил оборони

В Україні змінюють державний підхід до закупівель безпілотників для Сил оборони. Потреба у БПЛА формуватиметься автоматично на основі даних із фронту. Про це повідомив міністр оборони Михайло Федоров.

pravo.com.ua · 3 хвилини тому · новина · закласти · видалити · як прочитати

Бензин і дизель стрімко дорожчають: нардеп оцінив, що буде з цінами на АЗС далі



На ризик подорожчання пального в Україні впливає низка факторів, зокрема логістика, курс валют та вартість акцизів. Проте подальшого зростання ціни до рівня у 100-150 гривень за літр очікувати не варто.

upl.ua · 5 хвилини тому · новина · збільшити закладку · видалити · як прочитати

Логін

Пароль

ВВІЙТИ

Україна не планує залишати цю територію: NYT розповів, що відбувається на Донбасі

Віддавши Донбас, Зеленський чітко дав зрозуміти, що Україна не має наміру йти на поводу у вимог Кремля і залишати цю територію.

Тільки новина

назад · далі



Президент Володимир Зеленський під час візиту вказує українським військовим на Донбасі, де точиться військовий бій. Цією поведінкою він хоче продемонструвати один з основних принципів.

У дорозі його супроводжували журналісти видання The New York Times (NYT). Сьогодні вранці міста Краматорськ, Зелений склав їм, що хотів показати своїм самим тоном, що Росія несе на території, щоб та віддала весь Донбас в рамках Брестської угоди, навіть ту частину, яку контролює Україна.

"Віддавши Донбас, він чітко дав зрозуміти, що Україна не планує залишати цю територію".

"Справа не в кілометрах, справа в людях. Місто (Краматорськ - ред.) німає", - пояснив Зеленський, вказавши на сиво-чорний український прапор, що розвівався на вершині. "У місті розміщені війська і танкери. Якщо їхати на велосипеді - випадемо з собою. Продуктів нагадати відсут. У районі Донецької області, контрольовані Україною, проживає майже 200 000 осіб", - додав в інтерв'ю.

Рисунок В.12 – Слайд 12

Робота з програмою. Підписки та налаштування

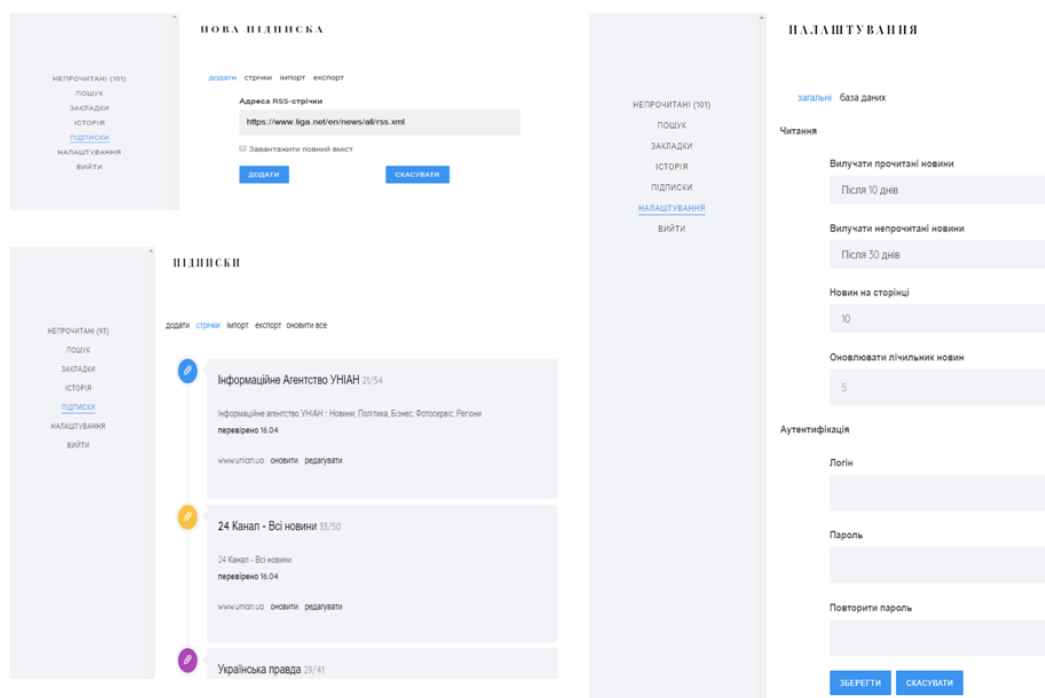


Рисунок В.13 – Слайд 13

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процеси, пов'язані з організацією та опрацюванням даних для агрегації та оброблення новинних потоків.

Сформульовано функціональні вимоги до програмної системи для агрегації та оброблення новинних потоків.

Для реалізації програмної системи для агрегації та оброблення новинних потоків обрано мову програмування PHP, середовище розробки Visual Studio Code та систему управління базами даних SQLite.

Розроблено програмну систему для агрегації та оброблення новинних потоків. Запропоновано структуру програмної системи для агрегації та оброблення новинних потоків. Розроблено базу даних програмної системи для агрегації та оброблення новинних потоків.

Виконано тестування розробленої програмної системи для агрегації та оброблення новинних потоків.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

Рисунок В.14 – Слайд 14