

Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

керівник

\_\_\_\_\_ Максим ГОЛОВЧЕНКО

“    ” \_\_\_\_\_ 2024 р.

**Веб-сервіс формування новин на основі публічних сторінок у соціальних  
мережах**

**Технічне завдання**

КПІ.ІІ-3324.045440.02.81

“ПОГОДЖЕНО”

Керівник роботи:

\_\_\_\_\_ Максим ГОЛОВЧЕНКО

Виконавець:

\_\_\_\_\_ Олександр СОКОЛОВ

Київ – 2025

## ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ .....	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ.....	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	7
4.1	Вимоги до функціональних характеристик .....	7
4.1.1	Користувацького інтерфейсу .....	7
4.1.2	Для користувача: .....	12
4.1.3	Для адміністратора системи (якщо він передбачений): .....	12
4.1.4	Додаткові вимоги: .....	13
4.2	Вимоги до надійності .....	13
4.3	Умови експлуатації .....	14
4.3.1	Вид обслуговування.....	14
4.3.2	Обслуговуючий персонал.....	14
4.4	Вимоги до складу і параметрів технічних засобів .....	15
4.5	Вимоги до інформаційної та програмної сумісності .....	15
4.5.1	Вимоги до вхідних даних .....	16
4.5.2	Вимоги до вихідних даних .....	16
4.5.3	Вимоги до мови розробки .....	16
4.5.4	Вимоги до середовища розробки.....	17
4.5.5	Вимоги до представленню вихідних кодів.....	17
4.6	Вимоги до маркування та пакування.....	17
4.7	Вимоги до транспортування та зберігання .....	17
4.8	Спеціальні вимоги .....	18
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ .....	19
5.1	Попередній склад програмної документації.....	19
5.2	Спеціальні вимоги до програмної документації .....	19
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	20
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ .....	21

## 1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Веб-сервіс формування новин на основі публічних сторінок у соціальних мережах

Наведене технічне завдання поширюється на розробку веб-сервісу формування новин на основі публічних сторінок у соціальних мережах **<Найменування >** [КП.ІІІ-3324.045440.02.81], котра використовується для створення новин на основі публічної інформації, доступної в соціальних мережах та інших онлайн-ресурсах та призначена для використання в галузі інформаційних технологій, зокрема в медіа, новинних агрегаторах і платформах для контент-менеджменту. Основними користувачами є медіа-агенції, журналісти, контент-менеджери, які потребують швидкого доступу до актуальних новин, а також адміністратори платформ, що забезпечують модерацію контенту. Система також може бути корисною для індивідуальних користувачів, зацікавлених у створенні персоналізованих новинних стрічок, та організацій, що займаються моніторингом соціальних мереж.

## 2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки <Найменування> є індивідуальний навчальний план студента та РНП затверджена кафедрою інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

### 3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Веб-сервіс для формування новин на основі публічних сторінок у соціальних мережах призначений для автоматизації збору, обробки та публікації новинного контенту з різноманітних джерел, таких як соціальні мережі (Telegram, Instagram) та RSS-канали. Функціонально система забезпечує користувачам можливість переглядати сторінку привітання, реєструватися, авторизуватися, додавати та перевіряти джерела, фільтрувати й редагувати контент, створювати та публікувати новини, а також залишати скарги на некоректний контент. Адміністратори отримують інструменти для управління джерелами та розгляду скарг, що сприяє підтримці якості інформації. Експлуатаційно веб-сервіс розроблено для роботи на настільних і мобільних пристроях, забезпечуючи зручний доступ через адаптивний інтерфейс і оптимізовану взаємодію з HTTP API соціальних мереж і RSS-каналів. Це полегшує роботу контент-менеджерів, журналістів і медіа-агенцій, дозволяючи ефективно обробляти великі обсяги даних у реальному часі.

Метою розробки є створення високопродуктивного та безпечного програмного продукту, який оптимізує процеси збору й обробки новин, забезпечуючи високу швидкість відповіді (до 2 секунд для 95% запитів), безпеку даних користувачів і контенту, а також доступність сервісу на рівні 99.9%. Якість досягається завдяки монолітній архітектурі з використанням Domain-Driven Design для структуризації коду, інтеграції з REST API через бібліотеку axios, застосуванню PostgreSQL для надійного зберігання даних і контейнеризації для масштабованості. Система підтримує адаптивний інтерфейс і локалізацію, що підвищує зручність використання для широкої аудиторії.

Мета дослідження полягає у створенні суспільно корисного веб-сервісу, який покращує якість і ефективність формування новинного контенту порівняно з традиційними методами ручної обробки. Це досягається шляхом автоматизації збору даних через HTTP API, забезпечення

інтуїтивного інтерфейсу для користувачів із різним рівнем технічної підготовки та реалізації надійних механізмів модерації контенту. Результатом є продукт, який пропонує швидший доступ до актуальних новин, підвищену безпеку та можливість масштабування для обробки зростаючих обсягів даних, що має бути чітко підтверджено у висновках курсової роботи.

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

#### 4.1.1 Користувацького інтерфейсу

- Перегляд сторінки привітання (Рисунок 4.1);
- Реєстрація користувача (Рисунок 4.2);
- Авторизація користувача (Рисунок 4.3);
- Додавання джерела новин (Рисунок 4.4);
- Перегляд і фільтрація контенту (Рисунок 4.5);
- Створення та редагування новин (Рисунок 4.6);
- Публікація новин (Рисунок 4.7);
- Залишення скарги (Рисунок 4.8);
- Управління джерелами адміністратором (Рисунок 4.9);
- Обробка скарг адміністратором (Рисунок 4.10).

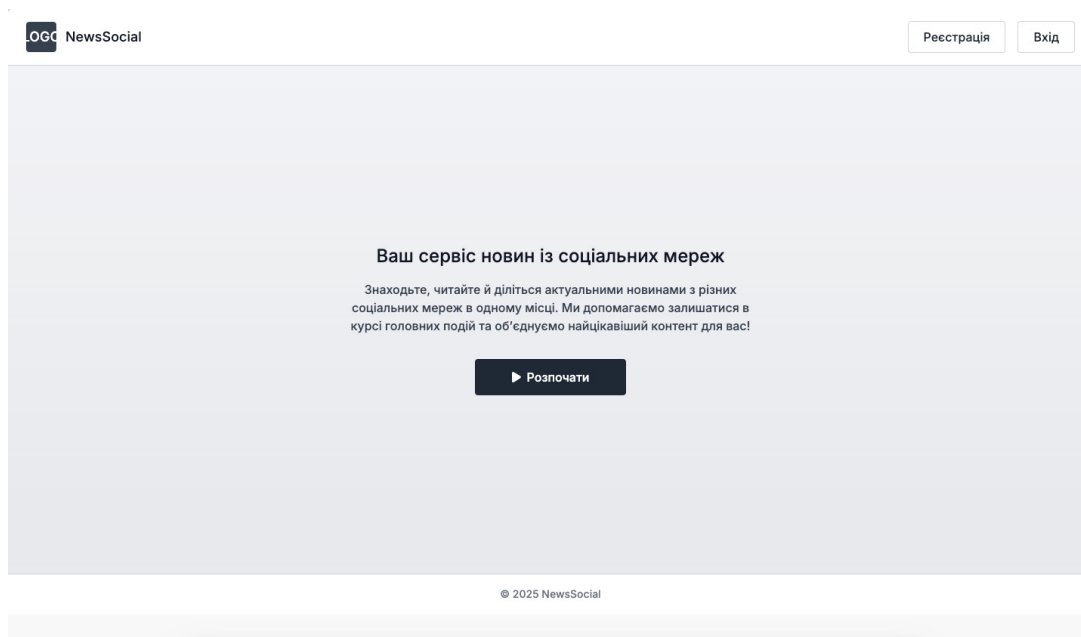


Рисунок 4.1 – Перегляд сторінки привітання

The screenshot shows the 'Створити обліковий запис' (Create account) page of the NewsSocial application. At the top left is the 'LOGO NewsSocial' header. At the top right are two buttons: 'Реєстрація' (Registration) and 'Вхід' (Login). The main content area features a white registration form with the following elements: a title 'Створити обліковий запис', an 'Email' field with the placeholder 'Введіть ваш email', a 'Пароль' (Password) field with the placeholder 'Введіть пароль', a 'Підтвердження пароля' (Confirm password) field with the placeholder 'Повторіть пароль', a dark grey 'Зареєструватися' (Register) button, and a link 'Уже маєте акаунт? Увійдіть' (Already have an account? Log in) at the bottom.

Рисунок 4.2 – Реєстрація користувача

The screenshot shows the 'Вхід до системи / Login' (Login) page of the NewsSocial application. The top header includes a gear icon and the text 'Авторизація / Authorization' on the left, and a 'Гість' (Guest) button on the right. The main content area features a white login form with the following elements: a title 'Вхід до системи / Login', an 'Email' field with the placeholder 'your@email.com', a 'Пароль / Password' field with masked characters '\*\*\*\*\*', a dark blue 'Увійти / Sign In' button, and a link 'Забули пароль? / Forgot password?' at the bottom.

Рисунок 4.3 – Авторизація користувача



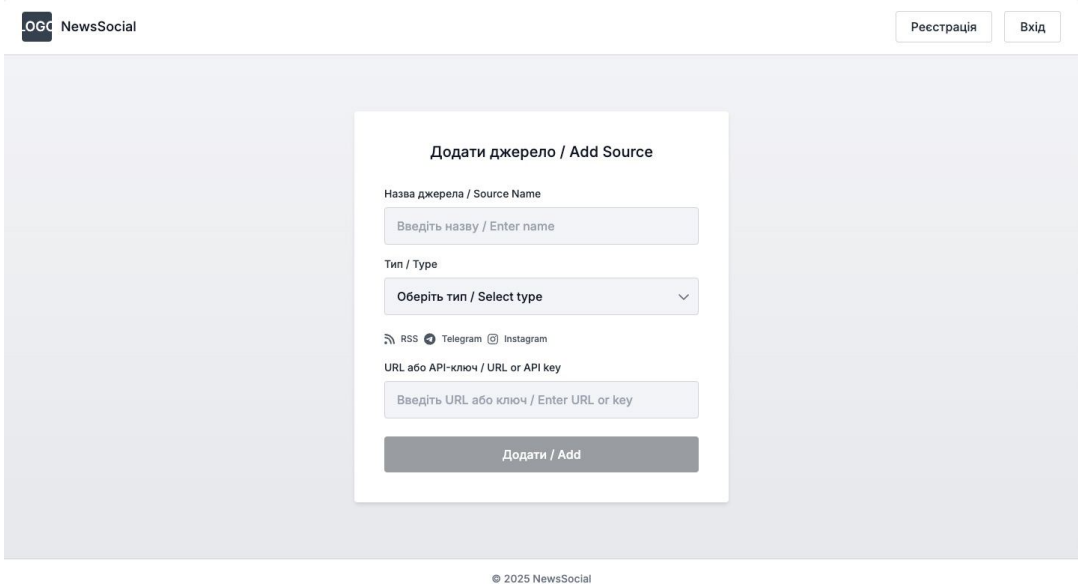


Рисунок 4.4 – Додавання джерела новин

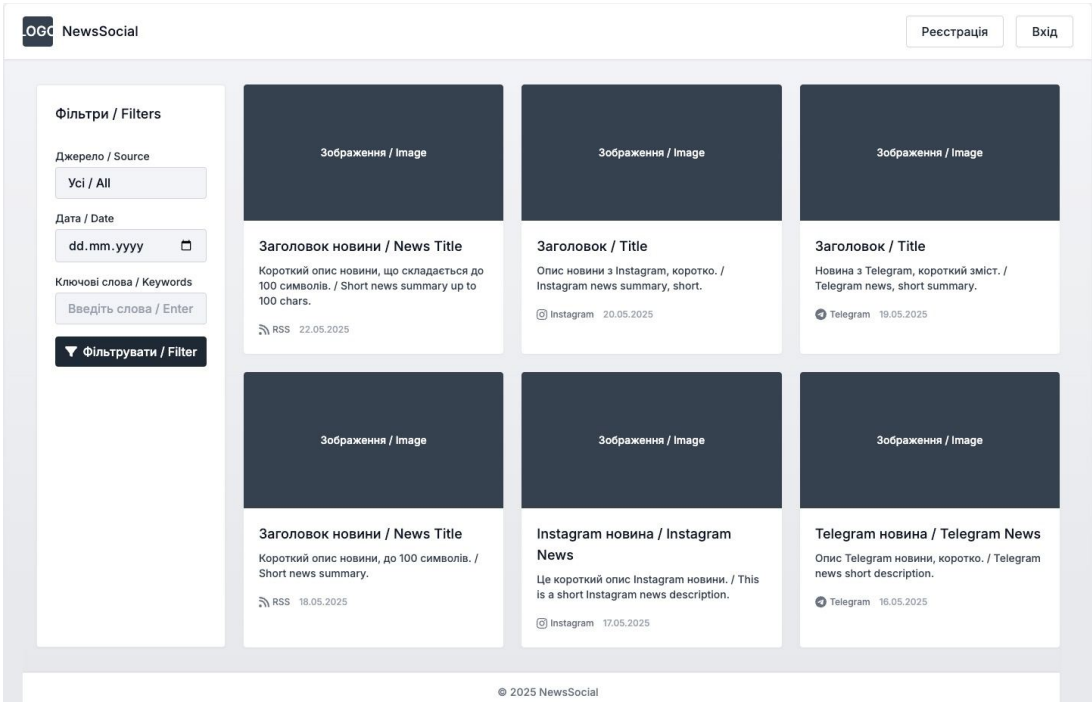


Рисунок 4.5 – Перегляд і фільтрація контенту

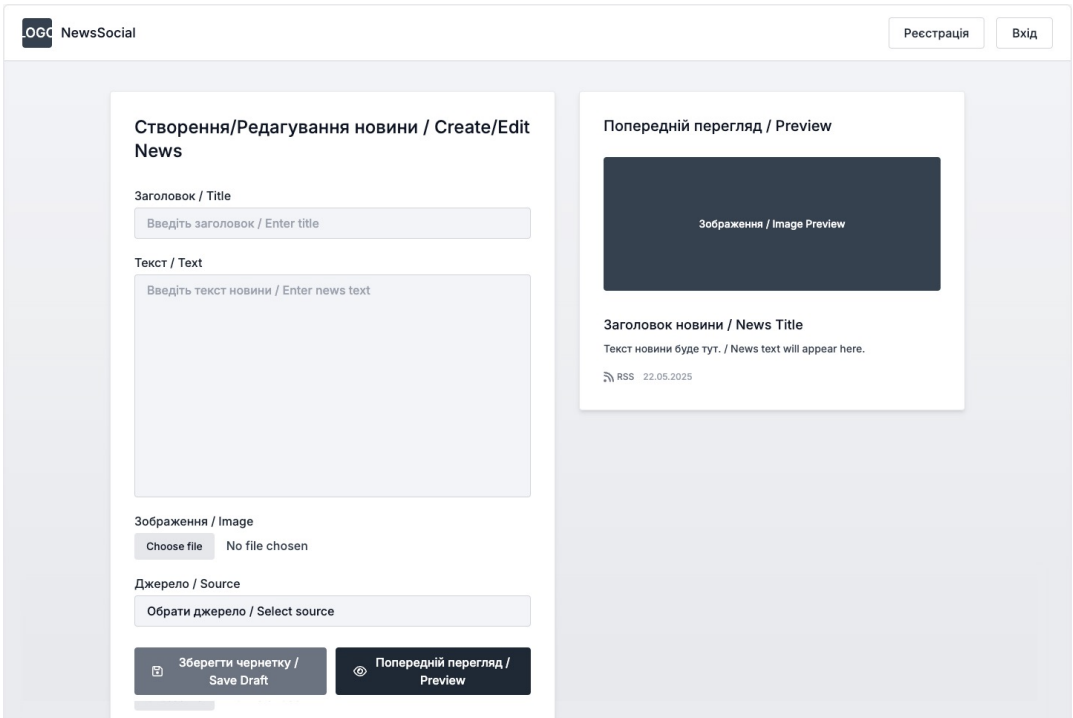


Рисунок 4.6 – Створення та редагування новин

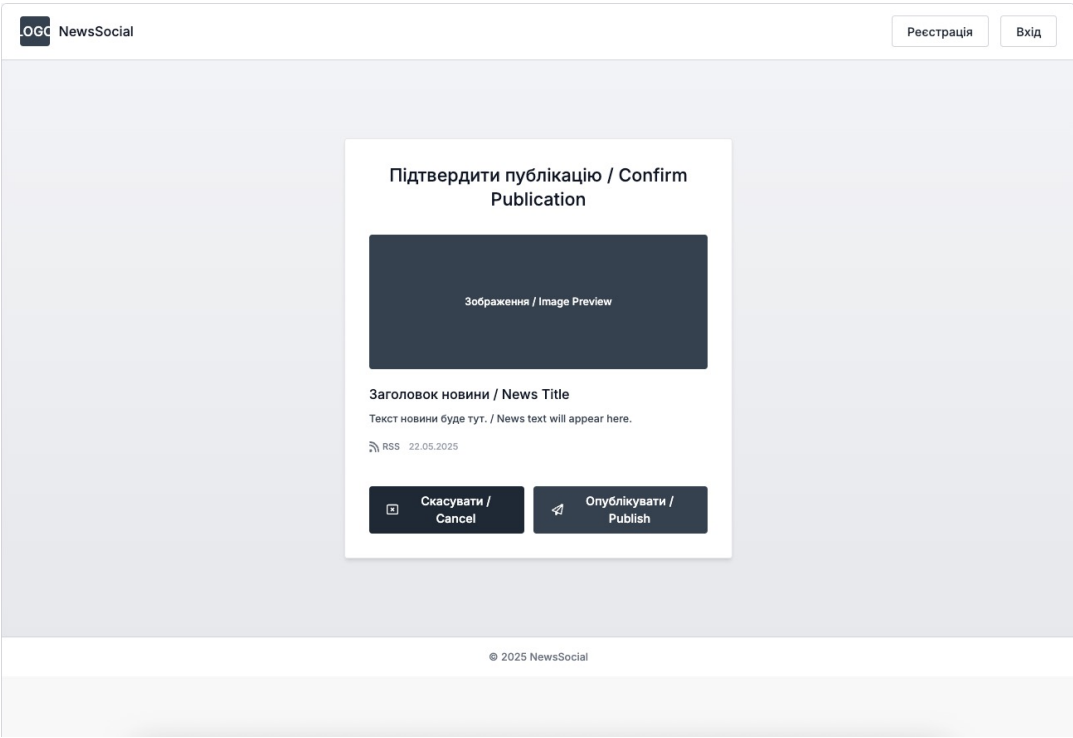


Рисунок 4.7 – Публікація новин

The screenshot shows a web application interface for NewsSocial. At the top left is the 'LOGO NewsSocial' and at the top right are buttons for 'Регистрація' (Registration) and 'Вхід' (Login). The main content area features a central form titled 'Залишити скаргу / Submit a Complaint'. The form includes a dropdown menu for 'Причина / Reason' with the option 'Оберіть причину / Select reason', a text area for 'Опис / Description' with the placeholder 'Опишіть проблему... / Describe the issue...', and a 'Надіслати / Submit' button. The footer contains the copyright notice '© 2025 NewsSocial'.

Рисунок 4.8 – Залишення скарги

The screenshot displays the 'Управління джерелами / Source management' interface. It includes a settings icon and a user profile button labeled 'Адміністратор'. The main section is titled 'Джерела / Sources' and contains a '+ Додати джерело / Add source' button. Below this is a table listing three sources: 'Новини UA' (RSS, Active), 'TechBlog' (API, Inactive), and 'World Updates' (Manual, Active). Each row has edit and delete icons.

Назва / Name	Тип / Type	URL	Статус / Status	Дії / Actions
Новини UA	RSS	news-ua.com/rss	Активний / Active	
TechBlog	API	api.techblog.com/v1	Неактивний / Inactive	
World Updates	Manual	worldupdates.org	Активний / Active	

Рисунок 4.9 – Управління джерелами адміністратором

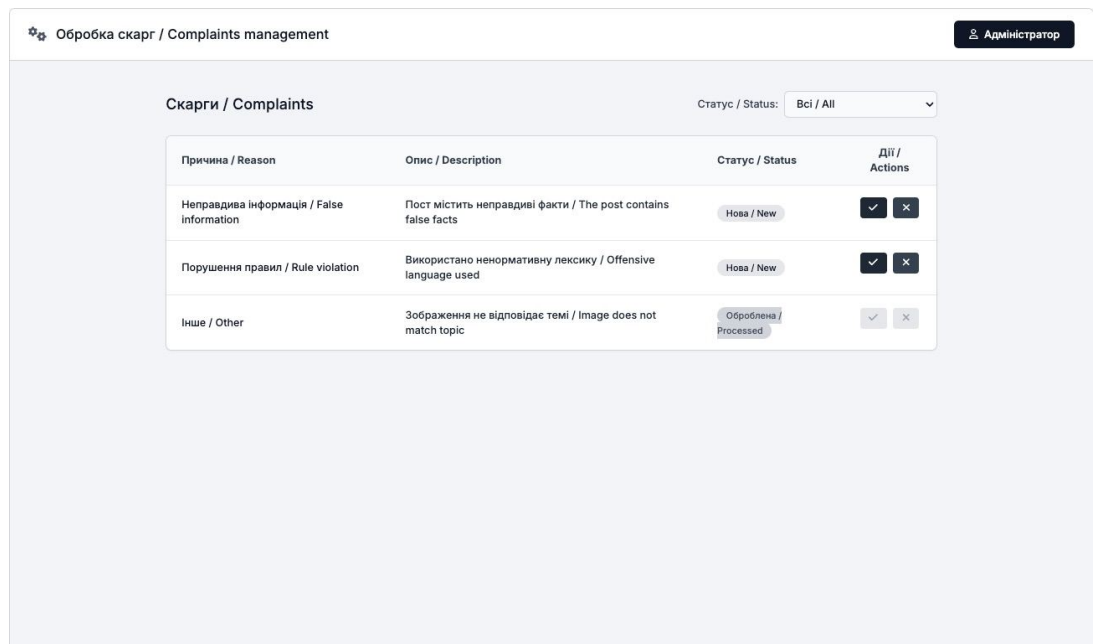


Рисунок 4.10 – Обробка скарг адміністратором

#### 4.1.2 Для користувача:

- Реєстрація облікового запису;
- Авторизація в системі;
- Вихід із системи;
- Додавання нового джерела новин;
- Перевірка валідності джерела;
- Пошук джерел за назвою або типом;
- Збір контенту з джерел;
- Фільтрація контенту за ключовими словами, датою чи джерелом;
- Редагування контенту перед створенням новини;
- Створення новин на основі зібраного контенту;
- Публікація новин;
- Залишення скарги на контент;
- Перегляд власного профілю.

#### 4.1.3 Для адміністратора системи:

- Перегляд списку джерел;

- Редагування або видалення джерел;
- Перегляд списку скарг;
- Обробка скарг (прийняття або відхилення);
- Управління правами доступу користувачів.

#### 4.1.4 Додаткові вимоги:

- Автоматична перевірка валідності джерел перед додаванням до системи
- Забезпечення доступу до системи через сучасні браузер

#### 4.2 Вимоги до надійності

Веб-сервіс для формування новин на основі публічних сторінок у соціальних мережах розроблено з акцентом на забезпечення надійності роботи системи, що є критично важливим для збереження даних і безперебійного функціонування. Для захисту від некоректних дій користувача реалізовано контроль введення інформації: усі поля форм (реєстрація, додавання джерел, створення новин, скарги) проходять валідацію на клієнтській (Vue.js) і серверній (NestJS) сторонах. Наприклад, поля для email перевіряються на відповідність формату, URL джерел – на валідність, а текстові поля – на захист від XSS-атак через екранування. У разі введення некоректних даних користувач отримує чіткі повідомлення про помилку, що мінімізує ризик некоректної поведінки. Цілісність інформації в базі даних PostgreSQL гарантується використанням транзакцій для всіх операцій запису, таких як додавання джерел, створення новин чи обробка скарг. Індеси та обмеження цілісності (foreign keys) забезпечують консистентність даних. Для відновлення після збоїв система створює автоматичні резервні копії бази даних кожні 24 години, які зберігаються в хмарному сховищі. Час відновлення системи після критичного збою не перевищує 10 хвилин завдяки контейнеризації (Docker) і автоматичному переключенню на резервні сервери. Контрольні точки для транзакцій

дозволяють відновлювати проміжні результати, наприклад, збережені чернетки новин, у разі переривання роботи. Логування всіх дій користувачів і системи через NestJS Logger забезпечує можливість аудиту та швидкого виявлення причин збоїв.

### 4.3 Умови експлуатації

Програмне забезпечення призначене для експлуатації на стаціонарних комп'ютерах, ноутбуках, планшетах і смартфонах із доступом до мережі Інтернет через сучасні веб-браузери (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge).

#### 4.3.1 Вид обслуговування

Веб-сервіс для формування новин на основі публічних сторінок у соціальних мережах не потребує спеціального обслуговування, оскільки розроблений як хмарний додаток із автоматизованим управлінням. Система розгортається на платформах, таких як AWS або Google Cloud, де оновлення, моніторинг і масштабування виконуються автоматично через контейнеризацію (Docker) і оркестрацію (Kubernetes). Регулярне оновлення програмного забезпечення, включаючи патчі безпеки та функціональні покращення, здійснюється через CI/CD-пайплайни без залучення спеціалізованого персоналу. Резервне копіювання бази даних PostgreSQL і логування подій налаштовано автоматично, що виключає потребу в ручному обслуговуванні. Таким чином, вимоги до виду обслуговування не висувуються, оскільки система спроектована для автономної роботи з мінімальним втручанням.

#### 4.3.2 Обслуговуючий персонал

Вимоги до обслуговуючого персоналу не висувуються

#### 4.4 Вимоги до складу і параметрів технічних засобів

Мінімальна конфігурація технічних засобів:

- Тип пристрою: ПК, ноутбук, планшет або смартфон із сучасним веб-браузером актуальних версій (Google Chrome, Firefox, Safari, Edge);
- Центральний процесор: 2-ядерний, з тактовою частотою від 1.4 ГГц;
- Оперативна пам'ять: не менше 2 ГБ;
- Інтернет-з'єднання: стабільне підключення зі швидкістю від 5 Мбіт/с;

Рекомендована конфігурація технічних засобів:

- Тип пристрою: ПК, ноутбук, планшет або смартфон із сучасним веб-браузером (останні версії Chrome, Firefox, Safari, Edge);
- Центральний процесор: 4-ядерний, з тактовою частотою від 2.0 ГГц.
- Оперативна пам'ять: не менше 4 ГБ;
- Інтернет-з'єднання: стабільне підключення зі швидкістю від 20 Мбіт/с.

#### 4.5 Вимоги до інформаційної та програмної сумісності

Веб-сервіс для формування новин розроблено для забезпечення широкої сумісності з сучасними платформами та технологіями, що дозволяє використовувати його на різноманітних пристроях і в різних середовищах. Клієнтська частина (веб-додаток на Vue.js із TypeScript) сумісна з сучасними браузерами, включаючи Chrome (версія 120+), Firefox (115+), Safari (16+), Edge (120+), що забезпечує коректну роботу на операційних системах Windows 10+, macOS 11+, Android 10+, iOS 15+. Серверна частина (NestJS, Node.js, PostgreSQL) розгортається на хмарних платформах із Linux-based середовищем (наприклад, Ubuntu 20.04+), що замінює застарілі вимоги до WIN32 чи Unix. Система інтегрується із зовнішніми джерелами (Telegram Bot

API, Instagram Graph API, RSS-канали) через стандартні REST API та XML/JSON формати, гарантуючи інформаційну сумісність.

#### 4.5.1 Вимоги до вхідних даних

Вхідні дані для системи представлені у структурованому форматі, залежно від джерела. Для соціальних мереж (Telegram, Instagram) дані надходять у форматі JSON через REST API, де кожен об'єкт містить поля, такі як текст повідомлення, URL зображення, дата публікації та метадані автора. RSS-канали надають дані у форматі XML або JSON, із структурою, що включає заголовок, опис, посилання та дату. Користувацькі дані, такі як email, пароль чи URL джерел, вводяться через веб-форми та валідуються на відповідність формату (наприклад, RFC 5322 для email, валідний URL для джерел). Некоректні дані відхиляються з повідомленням про помилку.

#### 4.5.2 Вимоги до вихідних даних

Вихідні дані системи формуються у вигляді опублікованих новин, представлених у форматі JSON для внутрішнього використання (API) та HTML для відображення у веб-інтерфейсі. Кожна новина містить заголовок (строка до 255 символів), текст (строка без обмеження довжини), опціональне зображення (URL або base64), дату публікації (ISO 8601) і джерело (посилання). У веб-інтерфейсі новини відображаються як картки з адаптивним дизайном, стилізовані через Tailwind CSS. Скарги та звіти адміністратора експортуються у форматі JSON або CSV для аналізу, із полями: ідентифікатор, причина, опис, статус.

#### 4.5.3 Вимоги до мови розробки

Розробка виконана на мовах програмування TypeScript (для клієнтської та серверної частин) і SQL (для запитів до PostgreSQL). TypeScript обрано через його строгую типізацію, що підвищує надійність коду, а SQL – для ефективної роботи з реляційною базою даних. Додаткові бібліотеки, такі як



axios (для REST API) і xml2js (для RSS), використовуються для обробки вхідних даних.

#### 4.5.4 Вимоги до середовища розробки

Середовище розробки включає платформу Node.js (версія 18+) для серверної частини та Vue.js (версія 3+) для клієнтської. NestJS слугує фреймворком для бекенду, забезпечуючи модульну структуру, а TypeORM – для взаємодії з PostgreSQL. Розробка ведеться в інтегрованому середовищі Visual Studio Code із плагінами для TypeScript, ESLint і Prettier. Для тестування використовується Jest, а для контейнеризації – Docker. CI/CD-пайплайни налаштовані через GitHub Actions для автоматизації розгортання.

#### 4.5.5 Вимоги до представленню вихідних кодів

Вихідний код програми представлений у вигляді структурованих файлів TypeScript, організованих за принципами Domain-Driven Design. Клієнтська частина містить компоненти Vue.js (файли .vue), модулі Pinia для управління станом і конфігурацію Vue Router. Серверна частина включає модулі NestJS (controllers, services, entities), структуровані за доменами (авторизація, джерела, контент). Код супроводжується коментарями англійською мовою, відповідає стандартам ESLint і форматується через Prettier. Усі вихідні файли зберігаються в репозиторії Git із чіткою структурою директорій, включаючи README.md із інструкціями для розгортання.

#### 4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не висуваються.

#### 4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

#### 4.8 Спеціальні вимоги

Згенерувати інсталяційну версію програмного забезпечення.

## **5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ**

### **5.1 Попередній склад програмної документації**

У склад супроводжувальної документації повинні входити наступні документи на аркушах формату А4:

- технічне завдання;
- пояснювальна записка;
- текст програми.

### **5.2 Спеціальні вимоги до програмної документації**

Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.



## **7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ**

Тестування розробленого програмного продукту виконується відповідно до “Розділу тестування програмного забезпечення”.