

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 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ СТВОРЕННЯ ТА ПУБЛІКАЦІЇ
СТАТЕЙ У БАГАТОКОРИСТУВАЦЬКОМУ БЛОЗІ
SOFTWARE FOR CREATING AND PUBLISHING
ARTICLES ON A MULTIUSER BLOG

Виконав(ла): студент(ка) 4 курсу, групи КНТ-113сп
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ІВАНЮК І.М.

(ПРІЗВИЩЕ та ініціали)

Керівник ОЛІЙНИК А.О.

(ПРІЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(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)

ІВАНЮКА Івана Михайловича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для створення та публікації статей у багатокористувацькому блозі. Software for Creating and Publishing Articles on a Multiuser Blog

керівник проєкту (роботи) д.т.н., професор, ОЛІЙНИК Андрій Олександрович,
(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 програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

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	ОЛІЙНИК А.О., професор		
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 програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	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) Іван ІВАНЮК
(Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Андрій ОЛІЙНИК
(Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
88 с., 3 табл., 29 рис., 3 дод., 21 джерело.

PYTHON, VISUAL STUDIO CODE, БАЗА ДАНИХ, БЛОГ, МОВА
ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес розробки програмного забезпечення для багатокористувацького блогу.

Предмет дослідження – програмні засоби для створення та публікації статей у багатокористувацькому блозі.

Мета роботи – розробка програмного забезпечення для створення та публікації статей у багатокористувацькому блозі, що забезпечує зменшення часу публікації статей та підвищення швидкості взаємодії авторів у багатокористувацькому блозі.

Матеріали, методи та технічні засоби: мова програмування Python, середовище розробки Visual Studio Code.

Результати. Розроблено проєктні рішення для створення програмного забезпечення для створення та публікації статей у багатокористувацькому блозі. Створено програмне забезпечення для створення та публікації статей у багатокористувацькому блозі. Проведено тестування розробленої програмної системи для створення та публікації статей у багатокористувацькому блозі.

Висновки. Мету роботи досягнуто. Розроблено програмне забезпечення для створення та публікації статей у багатокористувацькому блозі за допомогою мови програмування Python та середовища розробки Visual Studio Code.

Галузь використання – ведення блогів.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 88 pages, 3 tables, 29 figures, 3 appendixes, 21 sources.

PYTHON, VISUAL STUDIO CODE, DATABASE, BLOG, PROGRAMMING LANGUAGE, SOFTWARE.

The object of research is the process of developing software for a multi-user blog.

The subject of the research is software for creating and publishing articles in a multi-user blog.

The purpose of this work is to develop the software for creating and publishing articles in a multi-user blog, which ensures a reduction in the time of publishing articles and an increase in the speed of interaction between authors in a multi-user blog.

Materials, methods and technical tools: Python programming language, Visual Studio Code development environment.

Results. Project solutions for the creation of software for creating and publishing articles in a multi-user blog has been designed. The software to support the process of creating and publishing articles in a multi-user blog has been developed. The developed software system for creating and publishing articles in a multi-user blog has been tested.

Conclusions. The goal of the work has been achieved. The software tools for creating and publishing articles in a multi-user blog using the Python programming language and the Visual Studio Code development environment have been developed.

Scope of use – blogging.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Програмне забезпечення для створення та публікації статей у багатокористувацькому блозі.....	11
1.2 Аналіз програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.....	20
1.3 Висновки за розділом 1	29
2 Матеріали і методи	31
2.1 Вибір мови програмування	31
2.2 Вибір середовища розробки для створення програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.....	35
2.3 Висновки за розділом 2	39
3 Опис програми	40
3.1 Структура програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.....	40
3.2 Функціонування програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.....	42
3.3 Розробка бази даних програмного забезпечення для створення та публікації статей у багатокористувацькому блозі	44
3.4 Проєктування інтерфейсу програмного забезпечення для створення та публікації статей у багатокористувацькому блозі	47
3.5 Висновки за розділом 3	49
4 Експлуатація, тестування та експериментальне дослідження програми.....	51
4.1 Призначення й умови застосування програми	51
4.2 Характеристики програми для створення та публікації статей у багатокористувацькому блозі.....	53
4.3 Інструкція по експлуатації програми.....	54

4.3.1	Звернення до програми	54
4.3.2	Вхідні й вихідні дані	54
4.3.3	Повідомлення.....	55
4.4	Виконання програмного забезпечення для створення та публікації статей у багатокористувацькому блозі	55
4.5	Тестування програмного забезпечення для створення та публікації статей у багатокористувацькому блозі	59
4.6	Висновки за розділом 4	59
	Висновки	60
	Перелік джерел посилання	63
	Додаток А Технічне завдання	65
	Додаток Б Фрагмент тексту програми	69
	Додаток В Слайди презентації	81

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

VS – Visual Studio Code;

БД – база даних;

ПЗ – програмне забезпечення;

СКБД – система керування базами даних.

ВСТУП

Актуальність програмного забезпечення для створення та публікації статей у багатокористувацькому блозі зумовлюється зростанням потреби у швидкому поширенні інформації та розширенні цифрового середовища, у якому взаємодіють як окремі автори, так і цілі спільноти. У сучасних умовах онлайн платформи стають основним простором для ведення професійних дискусій, публікації експертних матеріалів та підтримання постійного діалогу між читачами та авторами, тому інструменти для ефективного створення контенту набувають особливої ваги. Збільшення обсягів інформації та потреба в оперативному її опрацюванні стимулюють використання рішень, які забезпечують зручне управління матеріалами та дають можливість авторам зосереджуватися на змісті без зайвих технічних обмежень [1].

Зростання популярності освітніх, наукових та корпоративних програмних платформ робить такі системи невід'ємним інструментом організації спільної діяльності. Платформи з можливістю багатокористувацької взаємодії дозволяють структурувати знання, впорядковувати теми та забезпечувати прозору комунікацію між авторами різного рівня підготовки. Саме тому програмні рішення, спрямовані на підтримку створення і публікації статей, залишаються важливим елементом цифрової екосистеми, сприяючи розвитку онлайн спільнот, формуванню авторитетних джерел інформації та створенню простору для систематичного обміну думками [1]-[4].

Проте деякі програми для створення та публікації статей у багатокористувацькому блозі за наявності високого навантаження можуть не забезпечувати прийнятної продуктивності, що ускладнює роботу авторів та може спричинити затримки у публікації матеріалів. Технічні обмеження інколи позначаються і на можливостях редагування: відсутність гнучких інструментів, незручний інтерфейс або надлишок другорядних функцій створює додаткові труднощі під час підготовки текстів. Суттєвим недоліком є

й необхідність постійного оновлення програмного забезпечення, що не завжди гарантує бездоганну роботу. Проблеми безпеки також залишаються одним з найвразливіших аспектів таких платформ. За умов багатокористувацького доступу зростає ймовірність несанкціонованих дій, випадкового видалення важливих матеріалів або поширення небажаного контенту. Додатковим недоліком є можливість виникнення конфліктних ситуацій між авторами, коли одночасні зміни у тих самих матеріалах призводять до накладення правок або втрати частини тексту. У таких випадках програмне забезпечення не завжди здатне коректно об'єднати різні варіанти однієї статті, що змушує учасників витрачати час на узгодження та відновлення змісту вручну. Таким чином, попри високий потенціал багатокористувацьких блогівих платформ, їх використання супроводжується низкою обмежень, які потребують постійної уваги, належного технічного супроводу та зваженого адміністрування [1]-[6].

Тому актуальною є розробка програмного забезпечення для створення та публікації статей у багатокористувацькому блозі. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.

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

- виконати аналіз предметної області та програмних засобів для створення та публікації статей у багатокористувацькому блозі;
- здійснити проектування програмного забезпечення для створення та публікації статей у багатокористувацькому блозі;
- створити програмне забезпечення для створення та публікації статей у багатокористувацькому блозі;
- виконати тестування розробленого програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для створення та публікації статей у багатокористувацькому блозі

Блог являє собою різновид онлайн ресурсу, де автор передає власні міркування, досвід або будь-які інші матеріали, що він бажає оприлюднити для широкої аудиторії. Сам термін походить від поняття мережевого журналу, що згодом набуло стислої форми. Йдеться про цифровий простір, у якому здійснюється створення та підтримання сторінки, призначеної для обміну думками, враженнями та ідеями, подібно до персонального щоденника, але у відкритому доступі [1], [2].

Основу такого ресурсу становить текстовий контент, що подається у формі окремих матеріалів. Такі записи зазвичай оформлюються у невимушеному стилі, зорієнтованому на безпосереднє спілкування з читачем. Тематика подібних публікацій може охоплювати будь-який інтерес автора, проте часто блог орієнтується на певну вузьку сферу, пов'язану зі стилем життя, подорожами, кулінарією, порадами, фінансовою грамотністю або іншими напрямками [1], [2].

Окремий матеріал у блозі являє собою текстову публікацію, що може виконувати роль огляду, пояснення чи аналітичного розбору. Такі записи нерідко мають неформальний характер, відрізняються від академічних робіт своєю простотою та доступністю. У межах блогу можуть з'являтися різні типи публікацій, серед яких особисті історії, міркування з приводу актуальних подій, інструктивні матеріали або пояснювальні посібники [1], [2].

Розрізнення між блогом і вебсайтом полягає у їхньому функціональному призначенні та організації контенту. Блог може існувати як частина ширшого ресурсу, що виконує інші завдання, наприклад комерційні або інформаційні. Також блог може бути окремим проєктом, структура якого включає головну сторінку та додаткові розділи, що підтримують його змістову цілісність. Матеріали в блозі розміщуються у зворотній часовій послідовності,

надаючи перевагу найсвіжішим записам. Вебсайти, натомість, формують власну систему сторінок залежно від мети, що може охоплювати представлення компанії, продуктів або сервісів. У блогах зазвичай домінує невимушений стиль подачі, тоді як вебресурси частіше використовують стриману, формальну манеру. Крім того, блоги передбачають активний контакт із читачами через коментарі та можливість поширення матеріалів, тоді як вебсайти можуть не мати такої інтерактивності. Блоги оновлюються значно частіше, що відрізняє їх від ресурсів, зміст яких змінюється лише за потреби. Основна роль блогу полягає у поясненні або розкритті певних тем для аудиторії, тоді як переважна частина вебсайтів сфокусована на наданні послуг або просуванні товарів [1], [2].

Процес ведення блогу пов'язаний із реєстрацією на платформі, що надає засоби для створення та публікації матеріалів. Після активації профілю відкривається можливість формувати власні записи, розміщувати їх у відкритому доступі, додавати візуальні елементи та поширювати контент у соціальних мережах для охоплення ширшої аудиторії. Цей механізм дозволяє організувати власний інформаційний простір та підтримувати постійний зв'язок із читачами [1], [2].

Програмне забезпечення для створення та публікації статей у багатокористувацькому блозі використовується для організації простору, у якому кілька авторів можуть одночасно працювати над контентом, публікувати власні матеріали та взаємодіяти з редакційною системою. Подібні програмні рішення забезпечують єдину платформу, що об'єднує інструменти для написання текстів, керування доступами, узгодження матеріалів і контролю за їхньою якістю. Робота з контентом відбувається в інтерактивному середовищі, де кожен учасник отримує роль із визначеним рівнем повноважень, що дозволяє підтримувати порядок у великій команді та уникати хаотичної подачі матеріалів. Такий підхід спрощує організацію спільних проєктів, оскільки вся інформація зберігається в одному місці й може оброблятися в реальному часі [1], [2].

Основні особливості подібного програмного забезпечення проявляються у можливості централізованого контролю. Редактори отримують інструменти для перевірки матеріалів, автори — зручні засоби написання й форматування текстів, адміністратори — можливість керувати всією системою та стежити за стабільністю роботи платформи. Значна увага приділяється організації колективної взаємодії, тому забезпечується наявність комунікаційних механізмів, журналів активності та функцій погодження матеріалів. Крім того, часто застосовується система керування версіями, завдяки якій з'являється змога відслідковувати зміни, повернути попередню редакцію або порівняти різні варіанти одного тексту [1], [2].

Серед переваг подібних рішень можна відзначити суттєве полегшення командної роботи. Авторам не доводиться обмінюватися файлами різних форматів або узгоджувати правки вручну, оскільки всі операції зосереджені на одній платформі. Спрощується контроль якості, зменшується кількість технічних помилок і покращується узгодженість структури контенту. Завдяки системі ролей зберігається чітка ієрархія, що допомагає уникнути несанкціонованих змін та підтримувати стабільність цілісності блогу. Крім того, централізованість сприяє підвищенню продуктивності, адже робота з матеріалами стає швидшою та передбачуваною [1], [2].

Водночас подібне програмне забезпечення має і свої недоліки. У разі великої кількості авторів може виникати надмірне навантаження на систему, що впливає на швидкість її роботи. Деякі платформи обмежують можливості персоналізації, що знижує гнучкість у створенні унікального стилю блогу. Організація розгалуженої редакційної структури вимагає від користувачів певного рівня технічної підготовки, тому новачки часто потребують додаткового навчання. Крім того, залежність від централізованого сервера створює ризики, пов'язані з можливими збоями, що можуть ускладнити доступ до матеріалів або навіть призвести до тимчасового блокування роботи всього проєкту [1], [2].

Блог використовується як інструмент для досягнення різноманітних цілей і приносить низку вигод. Основною перевагою вважається можливість отримання доходу. Деякі блоги здатні генерувати значні суми щомісяця, а професійні автори, які займаються блогінгом на постійній основі, отримують солідний річний дохід. Інформація про успішних блогерів демонструє потенціал цієї діяльності [3], [4].

Ведення блогу за допомогою відповідних програмних засобів сприяє формуванню онлайн репутації та розвитку особистого бренду. Публікація якісного контенту дозволяє утвердитися в певній сфері та відкриває нові можливості, такі як запрошення для виступів, пропозиції фріланс-проектів або високооплачуваних позицій [3], [4].

Блог також виступає ефективним джерелом заробітку в Інтернеті. Незважаючи на конкуренцію в різних нішах, послідовні дії здатні приносити прибуток, зокрема через партнерські програми, продаж курсів або власних продуктів. Одночасно процес ведення блогу розвиває різноманітні професійні навички. Регулярне написання дописів покращує вміння письма, редагування текстів формує якість контенту, вивчення SEO дозволяє підвищити відвідуваність сайту, а активність у соціальних мережах допомагає ефективніше взаємодіяти з аудиторією [3], [4].

Блог відкриває можливості для налагодження контактів із колегами та лідерами думок у певній сфері. Спілкування через соціальні мережі, коментарі та електронну пошту сприяє професійному зростанню та обміну досвідом. Він також може стати інструментом для генерації трафіку та продажів для бізнесу. Публікація релевантного контенту допомагає залучати цільових відвідувачів і конвертувати їх у клієнтів [3], [4].

Існує велика різноманітність блогів, які відрізняються тематикою та призначенням. Туристичні блоги діляться досвідом подорожей та практичними порадами. Особисті блоги відображають думки, емоції та повсякденне життя авторів. Нішеві блоги зосереджені на конкретних темах, таких як здоров'я, стиль життя або виховання дітей, і часто приносять дохід

через рекламу або партнерські програми. Блоги про моду та стиль життя висвітлюють актуальні тенденції, а блоги про здоров'я акцентують увагу на харчуванні, фітнесі та психічному благополуччі. Корпоративні блоги інформують про новини компанії та її продукти. Вибір теми повинен базуватися на знаннях та інтересах, щоб підтримувати активність протягом тривалого часу без втрати мотивації [3], [4].

Ведення блогу за допомогою відповідних програмних засобів потребує значних зусиль і часу. Регулярне створення контенту та просування сайту часто стає складним завданням, особливо в умовах високої конкуренції. Без належної організації легко заглибитися у рутину і витратити багато часу на дрібні завдання, не помічаючи прогресу. Тому перед початком варто зважити як переваги, так і складнощі цього процесу [4], [5].

Створення власного блогу починається з визначення тематики, що має комерційний потенціал і цікава самому автору. Після вибору ніші важливо залучати аудиторію, пропонуючи корисний і якісний контент у різних форматах, включно з текстами, відео або електронними матеріалами. Коли з'являється стабільна аудиторія, блог можна монетизувати через продаж товарів, рекламу, партнерські програми або спонсорські публікації. Паралельно варто активно просувати контент, щоб збільшити охоплення та залучити нових відвідувачів [4], [5].

Для ефективного ведення блогу важливо враховувати не лише змістові цілі, а й технічну складову. Перед запуском блогу визначаються очікувані результати, цільова аудиторія та стратегія монетизації. Використання програмного забезпечення дозволяє інтегрувати всі необхідні елементи для управління блогом, від налаштування реклами до автоматизації публікацій у соціальних мережах [5], [6].

Початковий етап полягає у плануванні: визначаються цілі блогу та його роль у розвитку особистого бренду або бізнесу. Якщо акцент ставиться на монетизацію, підбираються відповідні плагіни для розміщення реклами або інтегруються сервіси на кшталт Google AdSense, що полегшує процес

заробітку. Для інформаційних блогів основна увага приділяється створенню контенту, який забезпечує високі позиції у пошукових системах та підтримує SEO. Одночасно передбачається взаємодія із соціальними мережами через відповідні модулі, які дозволяють розміщувати стрічки, планувати пости та події [5], [6].

Важливою стає і концепція ніші блогу. Перевага нішевих блогів у концентрації на конкретних темах, що дозволяє більш ефективно залучати аудиторію та керувати контентом, порівняно із загальними блогами про стиль життя. Наступним кроком є вибір домену, який легко запам'ятовується, точно відображає тему блогу та підходить під надійні розширення, найчастіше .com. Для цього використовуються авторитетні сервіси реєстрації доменів, які дозволяють оцінити доступність і навіть придбати зайняті домени за потреби [5], [6].

Технічний аспект ведення блогу включає обрання хостингу, який забезпечує стабільну швидкість завантаження і високий рівень доступності сайту. Керовані платформи хостингу, такі як SiteGround, надають оптимальне поєднання швидкості та функціональності, що позитивно впливає на SEO і користувацький досвід. Після цього відбувається безпосереднє створення блогу на платформі. Найбільш універсальним рішенням залишається WordPress.org завдяки широким можливостям налаштування, доступності безкоштовних тем і плагінів, а також простоті освоєння. У більшості випадків хостинг-провайдери пропонують автоматичне встановлення WordPress, що значно спрощує процес запуску та налаштування блогу [5], [6].

У результаті програмне забезпечення дозволяє не лише організувати контент і налаштувати технічні параметри, а й ефективно управляти монетизацією, взаємодією з аудиторією та просуванням у пошукових системах. Блог стає зручним інструментом, який поєднує креативність і технічну оптимізацію для досягнення поставлених цілей [6], [7].

Для ефективного ведення блогу важливо враховувати не лише змістові цілі, а й технічну складову. Перед запуском блогу визначаються очікувані

результати, цільова аудиторія та стратегія монетизації. Використання програмного забезпечення дозволяє інтегрувати всі необхідні елементи для управління блогом, від налаштування реклами до автоматизації публікацій у соціальних мережах [6], [7].

Початковий етап полягає у плануванні: визначаються цілі блогу та його роль у розвитку особистого бренду або бізнесу. Якщо акцент ставиться на монетизацію, підбираються відповідні плагіни для розміщення реклами або інтегруються сервіси на кшталт Google AdSense, що полегшує процес заробітку. Для інформаційних блогів основна увага приділяється створенню контенту, який забезпечує високі позиції у пошукових системах та підтримує SEO. Одночасно передбачається взаємодія із соціальними мережами через відповідні модулі, які дозволяють розміщувати стрічки, планувати пости та події [6], [7].

Технічний аспект ведення блогу включає обрання хостингу, який забезпечує стабільну швидкість завантаження і високий рівень доступності сайту. Керовані платформи хостингу, такі як SiteGround, надають оптимальне поєднання швидкості та функціональності, що позитивно впливає на SEO і користувацький досвід. Після цього відбувається безпосереднє створення блогу на платформі. Найбільш універсальним рішенням залишається WordPress.org завдяки широким можливостям налаштування, доступності безкоштовних тем і плагінів, а також простоті освоєння. У більшості випадків хостинг провайдери пропонують автоматичне встановлення WordPress, що значно спрощує процес запуску та налаштування блогу [6], [7].

У результаті програмне забезпечення дозволяє не лише організувати контент і налаштувати технічні параметри, а й ефективно управляти монетизацією, взаємодією з аудиторією та просуванням у пошукових системах. Блог стає зручним інструментом, який поєднує креативність і технічну оптимізацію для досягнення поставлених цілей [7], [8].

Ведення блогу за допомогою спеціалізованого програмного забезпечення передбачає одночасне планування контенту та технічних

аспектів. На початковому етапі визначаються цілі блогу, його цільова аудиторія та потенційні шляхи монетизації. Інструменти програмного забезпечення дозволяють інтегрувати рекламу, автоматизувати публікації у соціальних мережах та керувати усіма елементами блогу з однієї платформи. Перш ніж розпочати, уточнюється концепція блогу: вирішується, чи буде він інформаційним, нішевим чи загальним, і як контент впливатиме на побудову бренду або бізнесу. При орієнтації на монетизацію підбираються плагіни для розміщення реклами або підключаються сервіси на кшталт Google AdSense, що спрощує управління доходом. Для інформаційних блогів пріоритет надається створенню відповідних матеріалів, оптимізованих під пошукові системи, що забезпечує видимість та трафік, а інтеграція соціальних мереж через відповідні модулі дозволяє залучати додаткову аудиторію та планувати пости автоматично [7], [8].

Наступним етапом стає обрання домену, який легко запам'ятовується, точно відображає тему блогу та використовує надійні розширення, найчастіше .com. Використовуються авторитетні сервіси для перевірки доступності домену і його придбання, якщо потрібно. Важливим кроком є також вибір хостингу, що забезпечує швидку роботу сайту та стабільність, оскільки це впливає на зручність користування і пошукову оптимізацію. Керовані платформи хостингу, такі як SiteGround, забезпечують високу продуктивність та простоту обслуговування [7], [8].

Процес створення блогу на платформі, наприклад WordPress.org, включає вибір теми, налаштування категорій, тегів і шаблонів, а також підключення плагінів для розширення функціоналу. Перевага WordPress полягає у широких можливостях налаштування, доступності безкоштовних інструментів і простоті освоєння, що робить його оптимальним для початківців. Більшість провайдерів хостингу пропонують автоматичне встановлення WordPress, що значно спрощує запуск і дозволяє зосередитися на створенні контенту [7], [8].

Використання спеціалізованого програмного забезпечення дозволяє поєднати творчий процес і технічне управління, оптимізувати монетизацію, просування і взаємодію з аудиторією, роблячи блог ефективним інструментом для досягнення поставлених цілей [7], [8].

При веденні блогу за допомогою спеціалізованого програмного забезпечення акцент зміщується на планування контенту та його технічну реалізацію. Після підготовки сторінок і шаблонів відбувається формування контент-стратегії, орієнтованої на конкретну нішу. Такий підхід дозволяє залучати увагу аудиторії, зміцнювати репутацію та підвищувати ефективність SEO. Для цього проводиться аналіз конкурентів, вивчаються потреби потенційних читачів і підбираються ключові слова, що допомагають контенту бути помітним у пошукових системах [7], [8].

Для забезпечення регулярності публікацій використовується редакційний календар. Планування частоти постів і підтримка послідовного графіка створює у читачів очікування і стимулює повторні відвідування. Такий підхід допомагає систематично управляти контентом і контролювати досягнення поставлених цілей [7], [8].

У підсумку, успішний блог формується завдяки поєднанню чіткого планування, цільової контент-стратегії, ефективної оптимізації та використання потужного програмного забезпечення для управління всіма процесами. Такий підхід створює міцну основу для зростання блогу, залучення аудиторії та потенційного отримання доходу [7], [8].

Визначено, що програмне забезпечення для створення та публікації статей у багатокористувацькому блозі використовується для організації простору, у якому кілька авторів можуть одночасно працювати над контентом, публікувати власні матеріали та взаємодіяти з редакційною системою. Подібні програмні рішення забезпечують єдину платформу, що об'єднує інструменти для написання текстів, керування доступами, узгодження матеріалів і контролю за їхньою якістю.

1.2 Аналіз програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Проаналізуємо програмне забезпечення для створення та публікації статей у багатокористувацькому блозі [9]-[17].

Програмне забезпечення Ghost (рис. 1.1) є інструментом професійного рівня, створеним для тих, хто працює з текстовим контентом і потребує зручного середовища для розбудови власного видання. Платформа Ghost орієнтується на авторів, які прагнуть встановити безпосередній зв'язок зі своїми читачами через систему членства та платних підписок. У її основі лежить прагнення забезпечити максимально легке та швидке створення матеріалів без перевантаження функціями, що властиві багатфункціональним системам керування контентом. Робочий процес вибудовується так, щоб автор зосереджувався на змісті, а не на технічних деталях [9], [10], [12], [15].

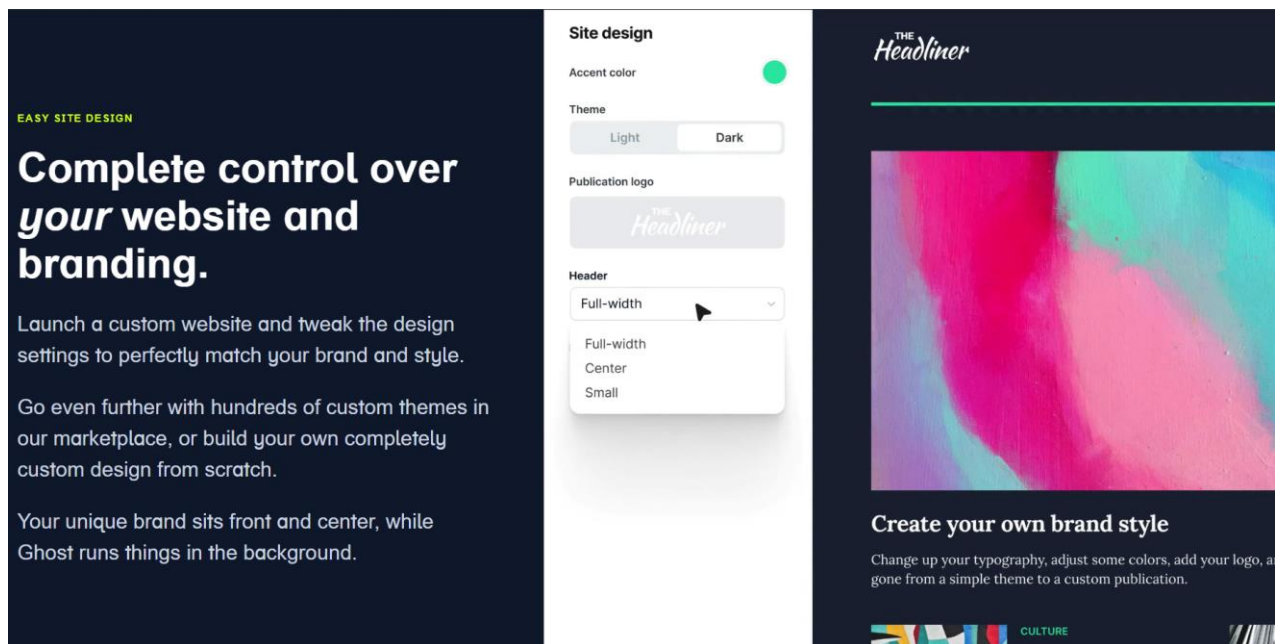


Рисунок 1.1 – Програмне забезпечення Ghost [15]

Однією з ключових можливостей Ghost є наявність вбудованого механізму монетизації, який дає змогу формувати платні матеріали та надсилати ексклюзивні повідомлення без потреби у додаткових розширеннях.

Редактор, створений у стриманому стилі, звільняє від зайвого візуального шуму та сприяє безперервному процесу написання. В основі платформи використано сучасне серверне середовище, яке забезпечує швидке завантаження матеріалів і надає повноцінну оптимізацію для пошукових систем, включно з автоматичним формуванням карт ресурсу, налаштуванням метатегів і підтримкою коректного поширення матеріалів у соціальних мережах [9], [10], [12], [15].

Найкраще середовище застосування Ghost проявляється тоді, коли автор прагне створити незалежне онлайн видання, що ґрунтується на членстві та власній аудиторії. Платформа стає особливо доречною для тих, хто цінує чистий інтерфейс і бажає працювати з уже вбудованими можливостями заробітку, не витрачаючи часу на складні технічні налаштування. Для запуску доступні два підходи. Можна скористатися варіантом керованого хостингу, де всі технічні питання розв'язуються автоматично, або обрати самостійне розгортання, що відкриває повний контроль над системою, але вимагає спеціальних технічних умінь [9], [10], [12], [15].

Основними характеристиками програмного забезпечення Ghost є такі [9], [10], [12], [15]:

- орієнтація на професійний контент: програмне забезпечення Ghost спрямоване на авторів, видавців та редакторів, які працюють із текстовими матеріалами та прагнуть підтримувати постійну взаємодію зі своєю аудиторією [9], [10], [12], [15];

- сучасний інтерфейс публікації: програмне забезпечення Ghost побудоване так, щоб автор міг працювати без відволікаючих елементів і зосереджуватися на змісті [9], [10], [12], [15];

- вбудована монетизація: програмне забезпечення Ghost дозволяє організовувати підписки та приватний доступ до матеріалів без потреби в додаткових розширеннях або налаштуваннях [9], [10], [12], [15];

- оптимізація для пошукових систем: програмне забезпечення Ghost автоматично формує ключові метадані та карти ресурсу, забезпечуючи високий рівень видимості матеріалів у пошукових сервісах [9], [10], [12], [15];
- використання сучасного технологічного середовища: робота побудована на швидкому й ефективному стеку, що дає можливість обробляти запити з мінімальними затримками [9], [10], [12], [15];
- можливість керованого або самостійного розгортання: користувач програмного забезпечення Ghost може обрати готовий хостинг або встановити програмну платформу на власному серверному середовищі для повного контролю [9], [10], [12], [15];
- обмежена кількість розширень і тем: екосистема менша порівняно з популярними універсальними CMS, що звужує можливості масштабного кастомізування [9], [10], [12], [15];
- фокус на публікації, а не на розширених функціях: програмне забезпечення Ghost найбільше підходить для блогів, розсилок і журналів, але не призначена для створення складних портальних систем [9], [10], [12], [15].

Серед недоліків програмного забезпечення Ghost можна відзначити обмежені можливості розширеного налаштування, оскільки система має значно меншу кількість доступних тем і інтеграцій, що ускладнює створення нестандартних рішень. Відчутною є й залежність від сучасного серверного середовища, яка потребує відповідних технічних навичок під час самостійного розгортання, що може стати бар'єром для користувачів без досвіду адміністрування. Також простежується певна спеціалізація програмної платформи Ghost, через яку вона зосереджена переважно на публікації матеріалів і не підходить для розроблення багатofункціональних вебресурсів із широким спектром сервісів [9], [10], [12], [15].

Програмне забезпечення Blogger (рис. 1.2) представляє собою платформу для онлайн публікацій, яка завдяки підтримці великої технологічної компанії набула широкого поширення серед авторів, що ведуть свої записи переважно з особистих мотивів [9], [11], [16].

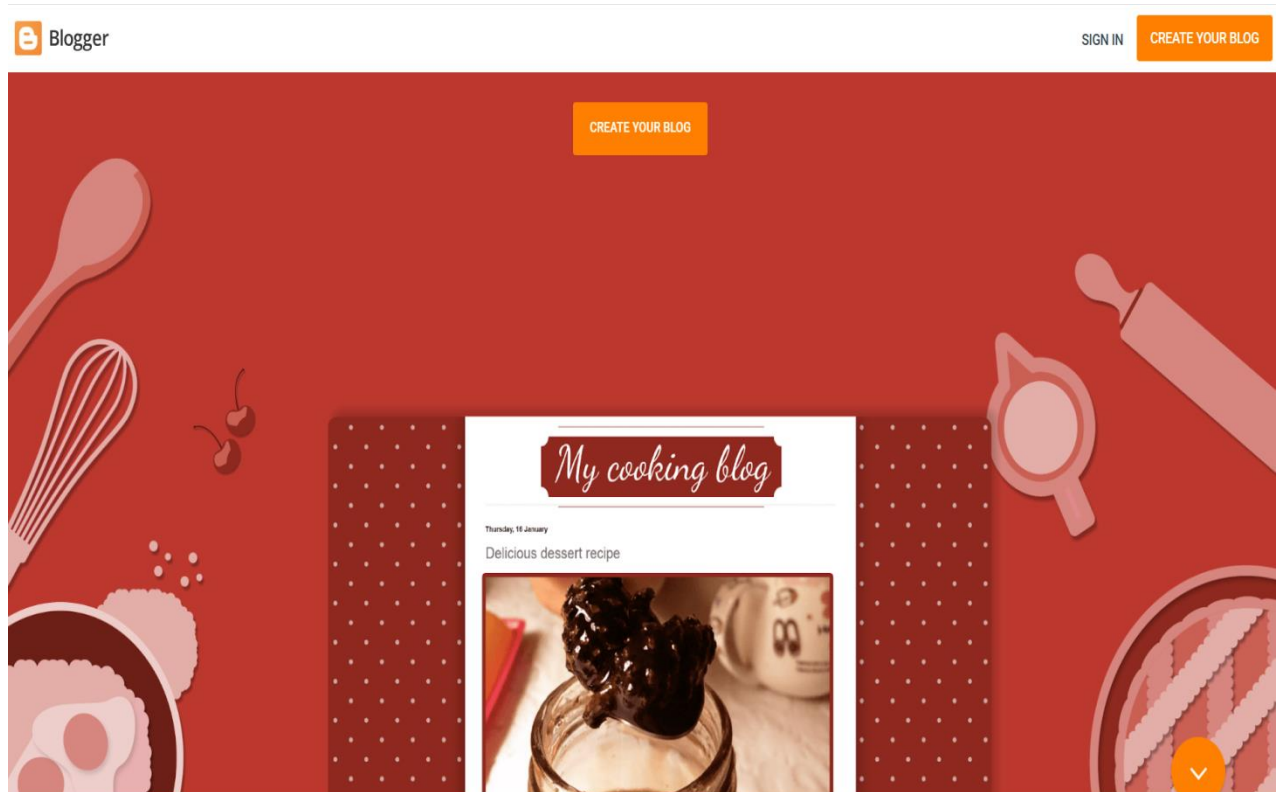


Рисунок 1.2 – Програмне забезпечення Blogger [16]

Доступ до сервісу Blogger відкривається одразу після входу в обліковий запис, пов'язаний із сервісами Google, що забезпечує швидкий старт і не потребує додаткових налаштувань. Робота з платформою не передбачає фінансових витрат, оскільки вона надає можливість користуватися безкоштовною адресою в доменній зоні Blogspot, а також дозволяє під'єднувати власні доменні імена без додаткової комісії [9], [11], [16].

Інтерфейс програмного забезпечення Blogger відзначається простотою та мінімалістичністю, через що освоюється без попереднього досвіду. Автори можуть обирати стандартні теми оформлення або завантажувати варіанти, створені сторонніми розробниками, а також виконувати поверхневі зміни зовнішнього вигляду на рівні шрифтів чи структури сторінок. Оскільки інструмент тісно взаємодіє з іншими продуктами Google, користувач отримує можливість легко під'єднати аналітику чи рекламні модулі, забезпечуючи монетизацію через рекламні оголошення [9], [11], [16].

Разом із тим програмне забезпечення Blogger має певні обмеження. Параметри кастомізації в ній доволі скромні, що ускладнює створення нестандартних макетів. Використання безкоштовної адреси може негативно впливати на сприйняття блогу аудиторією, адже такі домени часто асоціюються з низькою якістю. Застарілий вигляд редактора та невеликий вибір інструментів може створювати відчуття недостатньої сучасності у порівнянні з новішими сервісами [9], [11], [16].

Платформа Blogger найкраще підходить авторам, які створюють записи для власного задоволення та не планують розбудовувати складні комерційні ресурси. Простота початку роботи та повна безоплатність роблять її привабливою для користувачів, що шукають легкий спосіб оприлюднювати власні думки без зайвих технічних кроків [9], [11], [16].

Програмне забезпечення Blogger має такі особливості [9], [11], [16]:

- безоплатна основа: програмне забезпечення Blogger надає можливість публікувати матеріали без фінансових витрат, забезпечуючи доступ до власного блогу без необхідності придбання окремого програмного забезпечення чи хостингу [9], [11], [16];

- швидкий старт роботи: доступ до редактора відкривається відразу після входу в обліковий запис, що усуває потребу у складних налаштуваннях та дозволяє одразу перейти до створення контенту [9], [11], [16];

- інтеграція з екосистемою google: програмне забезпечення Blogger легко поєднується з інструментами на кшталт аналітики чи рекламних сервісів, що дозволяє ефективно моніторити трафік і використовувати монетизаційні можливості [9], [11], [16];

- базові можливості оформлення: програмне забезпечення Blogger підтримує використання стандартних тем та їх просте коригування на рівні шрифтів і структури сторінки, забезпечуючи мінімальні варіанти візуального налаштування [9], [11], [16];

- обмежена гнучкість: інструментарій персоналізації суттєво поступається більш сучасним платформам, що ускладнює створення нестандартних або професійних макетів [9], [11], [16];

- залежність від доменної зони blogspot: використання безкоштовної адреси створює враження менш професійного ресурсу та може негативно впливати на довіру читачів [9], [11], [16];

- мінімальна технічна підтримка: користувачі стикаються з обмеженим набором довідкових матеріалів, що може ускладнювати розв'язання нетипових задач або технічних проблем [9], [11], [16];

- можливість додавання гаджетів: у програмне забезпечення Blogger вбудовано набір додаткових блоків, які розширюють функціональність блогу, хоча створення чи завантаження складніших елементів потребує роботи з кодом [9], [11], [16];

- орієнтація на некомерційне використання: програмне забезпечення Blogger більше підходить авторам, що публікують матеріали для особистих потреб, оскільки інфраструктура платформи обмежена для побудови масштабних або бізнес-орієнтованих проєктів [9], [11], [16].

Серед недоліків програмного забезпечення Blogger можна відзначити обмеженість інструментів для глибокої персоналізації, через що створення унікального або складного зовнішнього вигляду стає майже недосяжним без втручання у код. Спрощена система керування структурою блогу зменшує можливості для розширення та робить платформу менш придатною для масштабних проєктів. Застарілий інтерфейс, що оновлюється нечасто, може відштовхувати користувачів, які звикли до сучасних візуальних рішень і розвинених панелей керування. Інтеграції з іншими сервісами поза екосистемою Google реалізовані дуже обмежено, тому взаємодія з професійними інструментами часто потребує додаткових технічних дій. Залежність від доменної зони blogspot може знижувати авторитетність ресурсу в очах аудиторії, а відсутність розвиненої спільноти та широкої підтримки ускладнює пошук рішень для нестандартних задач [9], [11], [16].

Програмне забезпечення Social Champ (рис. 1.3) є інструментом, покликаним підтримувати роботу блогу через автоматизацію поширення матеріалів та підтримання взаємодії з читацькою аудиторією. Його можливості орієнтовані на те, щоб автори могли не лише готувати контент, а й безперервно підтримувати присутність свого блогу у різних мережових середовищах. Завдяки цьому платформа слугує проміжною ланкою між створенням публікацій та їхнім поширенням, забезпечуючи стабільне охоплення та видимість статей після їх виходу [13], [17].

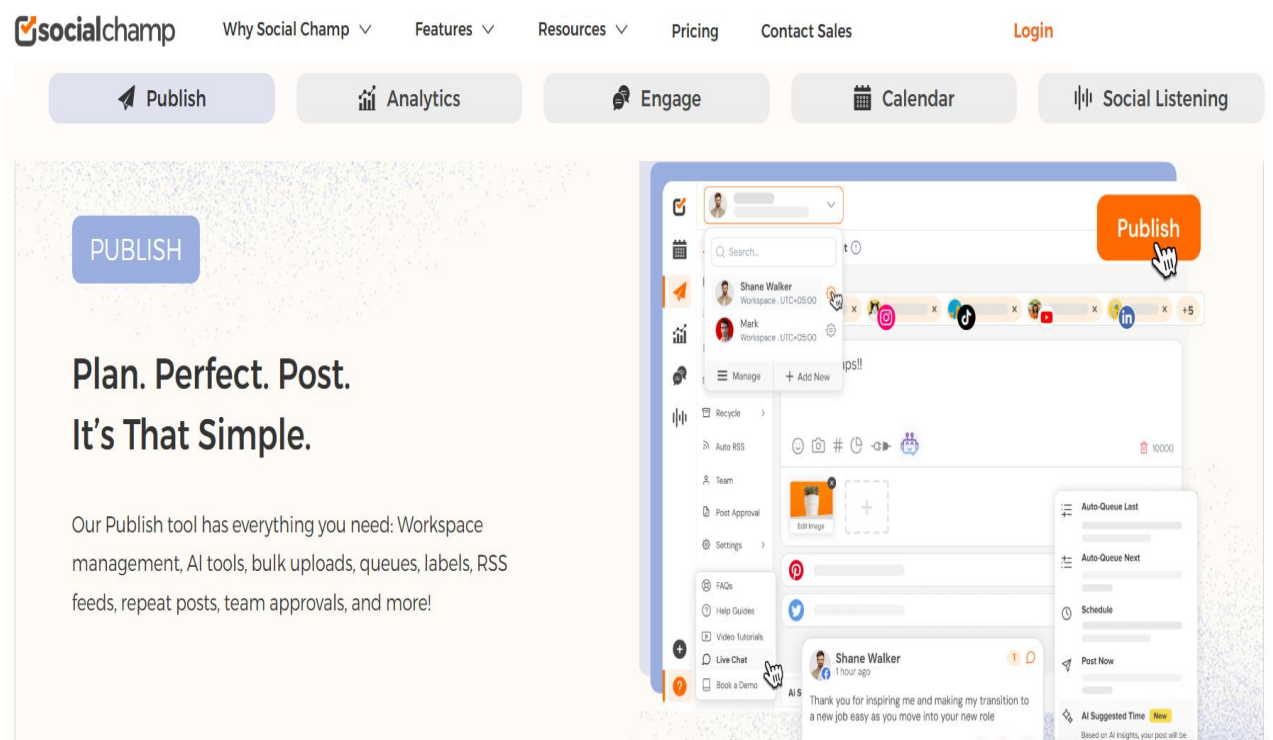


Рисунок 1.3 – Програмне забезпечення Social Champ [17]

У програмі Social Champ передбачено можливість завчасного розміщення матеріалів, що дозволяє блогерам підтримувати регулярність оновлень без необхідності постійної ручної публікації. Механізми оцінювання ефективності дають змогу стежити за реакцією аудиторії та розуміти, які теми найбільш привабливі для читачів. Ця аналітика сприяє оптимізації подальшої роботи та адаптації контентної стратегії [13], [17].

Програмне забезпечення Social Champ також підтримує командний підхід, коли над блогом працює кілька людей. Завдяки цьому матеріали можуть готуватися, узгоджуватися та плануватися спільно, що робить процес створення систематичним та впорядкованим. Програмне забезпечення Social Champ підходить як для невеликих блогів, що хочуть розвиватися, так і для більших ресурсів, де важливо забезпечити узгодженість стилю, структури та публікаційного графіка [13], [17].

Програмне забезпечення Social Champ має такі особливості [13], [17]:

- автоматизоване планування публікацій: програмне забезпечення Social Champ надає можливість створювати та розкладати контент для різних соціальних мереж, оптимізуючи час виходу дописів і підвищуючи регулярність активності [13], [17];

- аналітичний моніторинг: програмне забезпечення Social Champ забезпечує збір статистики щодо взаємодій із контентом, що дозволяє користувачам відстежувати ефективність публікацій і коригувати контентну стратегію [13], [17];

- підтримка декількох платформ: програмне забезпечення Social Champ дозволяє працювати з різними соціальними мережами в одному інтерфейсі, полегшуючи управління кількома сторінками або проектами [13], [17];

- інструменти для командної роботи: програмне забезпечення Social Champ дає змогу розподіляти ролі між учасниками команди та контролювати процес створення й публікації контенту [13], [17];

- інтеграції зі сторонніми сервісами: програмне забезпечення Social Champ підтримує підключення додаткових інструментів для розширення функціональності та підвищення ефективності роботи [13], [17].

Серед недоліків програмного забезпечення Social Champ можна відзначити обмеження деяких функцій у базових тарифах, що знижує доступність повного інструментарію для користувачів без платної передплати. Також можуть виникати затримки у синхронізації з окремими соціальними платформами, що іноді ускладнює своєчасну публікацію контенту [13], [17].

Порівняльну характеристику програмного забезпечення для створення та публікації статей у багатокористувацькому блозі наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Критерій порівняння	Ghost [15]	Blogger [16]	Social Champ [17]
Можливості персоналізації	+	+—	—
Інструменти для роботи з контентом	+	+—	+
Простота використання	+—	+	+—
Наявність інструментів аналітики	+—	—	+
Підтримка мультимедійних матеріалів	+	+	+—
Інтеграції зі сторонніми сервісами	+	+—	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для створення та публікації статей у багатокористувацькому блозі. Проте деякі програмні засоби для створення та публікації статей у багатокористувацькому блозі за наявності високого навантаження можуть не забезпечувати прийнятної продуктивності, що ускладнює роботу авторів та може спричинити затримки у публікації матеріалів. Технічні обмеження інколи позначаються і на можливостях редагування: відсутність гнучких інструментів, незручний інтерфейс або надлишок другорядних функцій створює додаткові труднощі під час підготовки текстів. Суттєвим недоліком є й необхідність постійного оновлення програмного забезпечення, що не завжди гарантує бездоганну роботу. Проблеми безпеки також залишаються одним з найвразливіших

аспектів таких платформ. За умов багатокористувацького доступу зростає ймовірність несанкціонованих дій, випадкового видалення важливих матеріалів або поширення небажаного контенту. Додатковим недоліком є можливість виникнення конфліктних ситуацій між авторами, коли одночасні зміни у тих самих матеріалах призводять до накладення правок або втрати частини тексту. У таких випадках програмне забезпечення не завжди здатне коректно об'єднати різні варіанти однієї статті, що змушує учасників витратити час на узгодження та відновлення змісту вручну. Таким чином, попри високий потенціал багатокористувацьких блогівих платформ, їх використання супроводжується низкою обмежень, які потребують постійної уваги, належного технічного супроводу та зваженого адміністрування. Тому актуальною є розробка програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.

При розробці програмного забезпечення для створення та публікації статей у багатокористувацькому блозі необхідно забезпечити такі функціональні вимоги:

- підтримка можливості відображення переліку публікацій у багатокористувацькому блозі;
- підтримка можливості реєстрації та авторизації;
- можливість створення публікацій користувачем;
- можливість редагування та видалення публікацій;
- підтримка можливості користувача з правами адміністратора редагування та видалення публікацій інших користувачів.

1.3 Висновки за розділом 1

Визначено, що програмне забезпечення для створення та публікації статей у багатокористувацькому блозі використовується для організації простору, у якому кілька авторів можуть одночасно працювати над контентом, публікувати власні матеріали та взаємодіяти з редакційною системою. Подібні

програмні рішення забезпечують єдину платформу, що об'єднує інструменти для написання текстів, керування доступами, узгодження матеріалів і контролю за їхньою якістю.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для створення та публікації статей у багатокористувацькому блозі. Проте деякі програмні засоби для створення та публікації статей у багатокористувацькому блозі за наявності високого навантаження можуть не забезпечувати прийнятної продуктивності, що ускладнює роботу авторів та може спричинити затримки у публікації матеріалів. Технічні обмеження інколи позначаються і на можливостях редагування: відсутність гнучких інструментів, незручний інтерфейс або надлишок другорядних функцій створює додаткові труднощі під час підготовки текстів. Суттєвим недоліком є й необхідність постійного оновлення програмного забезпечення, що не завжди гарантує бездоганну роботу. Таким чином, попри високий потенціал багатокористувацьких блогових платформ, їх використання супроводжується низкою обмежень, які потребують постійної уваги, належного технічного супроводу та зваженого адміністрування. Тому актуальною є розробка програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.

Сформульовано функціональні вимоги до програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для створення та публікації статей у багатокористувацькому блозі було використано мову програмування Python [18], [19].

Мова Python відзначається високою універсальністю, що дозволяє застосовувати її у найрізноманітніших сферах сучасного програмування. Її синтаксис відзначається простотою та логічною структурою, що значно знижує поріг входження для початківців і одночасно робить код зрозумілим та підтримуваним для командних проєктів. Завдяки цьому стає можливим швидко розробляти прототипи, тестувати ідеї та впроваджувати їх у реальних системах без потреби у великих витратах часу на освоєння складних конструкцій мови. Крім того, Python володіє потужними можливостями для обробки даних, що забезпечує ефективне опрацювання великих масивів інформації та інтеграцію з різними джерелами даних, включно з базами даних, вебсервісами та зовнішніми API [18], [19].

Важливою особливістю є наявність розвиненої екосистеми сторонніх бібліотек і фреймворків, що охоплюють практично всі напрямки програмування, від веброзробки та автоматизації до штучного інтелекту та машинного навчання. Це дозволяє зменшувати обсяг власноруч написаного коду, підвищуючи швидкість створення додатків і водночас підвищуючи їхню стабільність. Python підтримує модульність і об'єктно-орієнтований підхід, що робить структуру проєктів логічною і легко масштабованою, дозволяючи без проблем додавати нові функції або змінювати існуючі без ризику порушення загальної роботи системи [18], [19].

Однією з визначальних переваг мови є її кросплатформеність, що дозволяє виконувати програми на різних операційних системах без суттєвих змін у коді. Це відкриває широкі можливості для створення універсальних програмних продуктів і забезпечує їхню сумісність із різними середовищами

користувачів. Крім того, Python активно використовується у наукових дослідженнях і бізнес-аналітиці завдяки потужним інструментам для математичного моделювання, статистичної обробки даних та візуалізації результатів [18], [19].

Важливою перевагою стає і активна спільнота розробників, яка постійно створює нові бібліотеки, надає приклади вирішення складних завдань і підтримує новачків у вивченні мови. Завдяки цьому реалізація нестандартних проєктів значно спрощується, оскільки можна швидко знайти готові рішення та адаптувати їх під власні потреби. Також Python дозволяє легко інтегруватися з іншими мовами та технологіями, забезпечуючи гнучкість у виборі інструментів для різних частин проєкту [18], [19].

У сучасному цифровому середовищі Python демонструє здатність швидко адаптуватися до нових вимог і тенденцій, підтримуючи розвиток веб-технологій, автоматизації, обробки великих даних та штучного інтелекту. Це робить його не лише ефективним інструментом для створення програмного забезпечення, а й універсальною платформою для реалізації складних, масштабованих та гнучких рішень у різних сферах діяльності [18], [19].

Доцільність використання мови Python для створення програмного забезпечення, орієнтованого на роботу багатокористувацького блогу, пов'язана насамперед із поєднанням швидкості розробки, доступності готових інструментів і високої гнучкості, яка дозволяє ефективно реалізувати як базовий функціонал публікації статей, так і складні механізми взаємодії користувачів. Застосування цієї мови забезпечує можливість створення повноцінної вебсистеми значно швидше, ніж у випадку з багатьма іншими технологіями, адже велика кількість бібліотек і фреймворків дає змогу скоротити час на реалізацію серверної логіки, керування даними та інтеграцію додаткових модулів [18], [19].

Python має широке підтримання у сфері веброзробки завдяки таким фреймворкам, як Django або Flask, які надають готові компоненти для роботи з авторизацією, маршрутизацією, управлінням базою даних, шаблонізацією та

відображенням контенту. Це особливо важливо для багатокористувацького блогу, де безпека, швидка обробка запитів і чітка організація даних мають ключове значення. Використання цих фреймворків дозволяє будувати масштабовані системи, здатні працювати зі значним обсягом контенту й одночасним доступом багатьох авторів [18], [19].

Ще однією перевагою Python є його читабельність та простота підтримки, що спрощує подальший розвиток проєкту. Код, написаний цією мовою, легко модифікувати та розширювати, що дає можливість швидко додавати нові функції: системи коментування, засоби модерації, аналітичні модулі або механізми персоналізації публікацій. Завдяки цьому програмне забезпечення може адаптуватися до змін потреб користувачів без значних витрат часу чи ресурсів [18], [19].

Мова програмування Python також відзначається розвиненою екосистемою для роботи з базами даних, що дає змогу впорядковувати статті, користувачів та коментарі таким чином, щоб забезпечити стабільність і швидку взаємодію. Інструменти ORM спрощують управління даними та підвищують надійність системи, зменшуючи ризик помилок під час роботи з великими обсягами інформації [18], [19].

Крім того, Python має активну спільноту та великий обсяг документації, що значно полегшує вирішення технічних питань у процесі розробки. У разі необхідності можна швидко знайти готові рішення або приклади реалізації подібних функцій, тому створення багатокористувацького блогу на мові Python стає не лише ефективним, але й економним з погляду ресурсів [18], [19].

Таким чином, використання Python дає змогу отримати стабільне, масштабоване та зручне програмне забезпечення, яке відповідає вимогам сучасних вебплатформ і забезпечує комфортну роботу великої кількості авторів [18], [19].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для створення та публікації статей у багатокористувацькому блозі наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Критерій порівняння мов програмування	Мова програмування		
	Python	PHP	JavaScript
Масштабованість	+	+—	+—
Гнучкість і універсальність	+	+—	+—
Безпека та стабільність	+	+	+—
Зручність для створення ПЗ для створення та публікації статей у багатокористувацькому блозі	+	—	—
Спільнота та документація	+	+—	+

Отже, для реалізації програмного забезпечення для створення та публікації статей у багатокористувацькому блозі обрано мову програмування Python, яка забезпечує поєднання швидкості розробки, доступності готових інструментів і високої гнучкості, що дозволяє ефективно реалізувати як базовий функціонал публікації статей, так і складні механізми взаємодії користувачів. Застосування цієї мови забезпечує можливість створення повноцінної вебсистеми значно швидше, ніж у випадку з багатьма іншими технологіями, адже велика кількість бібліотек і фреймворків дає змогу скоротити час на реалізацію серверної логіки, керування даними та інтеграцію додаткових модулів.

2.2 Вибір середовища розробки для створення програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

В якості середовища розробки для створення програмного забезпечення для створення та публікації статей у багатокористувацькому блозі було обрано середовище Visual Studio (VS) Code [20], [21].

Середовище розробки Visual Studio Code вирізняється високою адаптивністю до різних типів проєктів і рівнів складності програмного забезпечення, що дозволяє створювати як невеликі модулі, так і великі багатокористувацькі платформи. Його архітектура передбачає модульність та підтримку розширень, завдяки чому можна легко інтегрувати додаткові інструменти для автоматизації процесів, тестування, деплою та налагодження коду, не виходячи з середовища розробки. Такий підхід забезпечує суттєву економію часу на рутинні операції та підвищує продуктивність розробників, дозволяючи зосередитися на логіці проєкту та його функціональних можливостях [20], [21].

Особливою перевагою Visual Studio Code є його здатність до глибокої інтеграції з системами контролю версій, хмарними сервісами та платформами для спільної розробки, що робить його ефективним інструментом для командної роботи над складними проєктами. Платформа забезпечує можливість одночасного відстеження змін у коді, ведення історії правок і швидкого вирішення конфліктів, що значно полегшує керування багатокористувацькими системами, де одночасно працює велика кількість авторів і розробників [20], [21].

Visual Studio Code також надає потужні інструменти для налагодження та аналізу коду, що дозволяє виявляти помилки на ранніх етапах розробки та зменшувати ймовірність збоїв у роботі програмного забезпечення. Середовище підтримує інтерактивне тестування, профілювання продуктивності та підсвітку проблем у реальному часі, що забезпечує більш

високу якість кінцевого продукту та знижує витрати на його подальшу підтримку [20], [21].

Не менш важливим аспектом є кросплатформеність Visual Studio Code, яка дозволяє використовувати його на різних операційних системах без необхідності внесення суттєвих змін у робочі процеси. Це забезпечує універсальність середовища та можливість організувати єдине робоче середовище для команди розробників з різними технічними платформами. Наявність великої і активної спільноти користувачів сприяє постійному оновленню і вдосконаленню функціоналу, надає доступ до численних навчальних ресурсів, прикладів реалізації складних функцій та готових рішень для інтеграції нових модулів у проєкт [20], [21].

Крім того, Visual Studio Code забезпечує інтуїтивно зрозумілий інтерфейс, який поєднує гнучкість професійного середовища та простоту для початківців, дозволяючи швидко опановувати робочий процес і ефективно виконувати завдання будь-якого рівня складності. Можливість персоналізації зовнішнього вигляду, налаштування гарячих клавіш та адаптації під конкретні вимоги проєкту сприяє підвищенню комфорту роботи і оптимізації робочого часу [20], [21].

Таким чином, Visual Studio Code поєднує швидкість розробки, гнучкість налаштувань, потужні інструменти для контролю якості та ефективну командну роботу, що робить його оптимальним вибором для створення сучасного багатокористувацького програмного забезпечення.

Доцільність використання середовища розробки Visual Studio Code для створення програмного забезпечення багатокористувацького блогу зумовлюється його широким функціоналом, який забезпечує ефективну роботу над складними проєктами та значно прискорює процес розробки. VS Code підтримує безліч мов програмування, у тому числі Python, що робить його універсальним інструментом для створення серверної логіки, інтеграції баз даних та розробки вебінтерфейсів. Наявність великої кількості плагінів і розширень дозволяє автоматизувати рутинні операції, наприклад,

автозавершення коду, форматування, підсвітку синтаксису, налагодження та інтеграцію з системами контролю версій, що особливо важливо при роботі над багатокористувацькими платформами [20], [21].

VS Code відзначається гнучкістю конфігурації під потреби конкретного проєкту. Можливість налаштовувати середовище, додавати власні розширення та інтегрувати сторонні інструменти дозволяє створювати ефективний робочий простір для командної розробки. Це полегшує управління великими кодовими базами та одночасну роботу кількох розробників, забезпечуючи узгодженість коду і швидке виявлення помилок. Наявність інтегрованої підтримки систем контролю версій, таких як Git, дозволяє відстежувати зміни, координувати роботу учасників та зменшувати ризик конфліктів у багатокористувацькому проєкті [20], [21].

Особливою перевагою VS Code є легкість у роботі з веб-технологіями, що робить його зручним для розробки фронтенду блогівих платформ. Завдяки вбудованій підтримці HTML, CSS, JavaScript та популярних фреймворків можна ефективно реалізовувати інтерфейс користувача, створювати динамічні сторінки та тестувати інтерактивні елементи без потреби у додаткових середовищах. Це скорочує час на налагодження та підвищує продуктивність команди [20], [21].

Крім того, VS Code активно підтримується спільнотою розробників, що забезпечує швидкий доступ до навчальних матеріалів, прикладів коду та готових рішень для типових задач. Завдяки цьому реалізація багатокористувацьких функцій, таких як система коментарів, механізми модерації або інтеграція з базою даних, стає значно простішою. Легка вага середовища, швидке завантаження та можливість роботи на різних операційних системах роблять VS Code практичним і доступним інструментом для розробки програмного забезпечення будь-якої складності [20], [21].

Таким чином, використання Visual Studio Code забезпечує поєднання зручності, гнучкості та потужних функціональних можливостей, що робить

процес розробки багатокористувацького блогу швидким, ефективним і масштабованим [20], [21].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для створення та публікації статей у багатокористувацькому блозі наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio Code	Atom	Sublime Text
Налагодження та профілювання	+	+—	+—
Інтеграція з системами контролю версій	+	+	+—
Підтримка командної роботи	+	+—	—
Легкість налаштування	+	+—	+
Простота інтерфейсу та користування	+	+—	+

Таким чином, для створення програмного забезпечення для створення та публікації статей у багатокористувацькому блозі обрано середовище розробки Visual Studio Code, що характеризується широким функціоналом, який забезпечує ефективну роботу над складними проєктами та значно прискорює процес розробки, є універсальним інструментом для створення серверної логіки, інтеграції баз даних та розробки вебінтерфейсів. Наявність великої кількості плагінів і розширень дозволяє автоматизувати рутинні

операції, наприклад, автозавершення коду, форматування, підсвітку синтаксису, налагодження та інтеграцію з системами контролю версій, що особливо важливо при роботі над багатокористувацькими платформами.

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення для створення та публікації статей у багатокористувацькому блозі обрано мову програмування Python, яка забезпечує поєднання швидкості розробки, доступності готових інструментів і високої гнучкості, що дозволяє ефективно реалізувати як базовий функціонал публікації статей, так і складні механізми взаємодії користувачів. Застосування цієї мови забезпечує можливість створення повноцінної вебсистеми значно швидше, ніж у випадку з багатьма іншими технологіями, адже велика кількість бібліотек і фреймворків дає змогу скоротити час на реалізацію серверної логіки, керування даними та інтеграцію додаткових модулів.

Для створення програмного забезпечення для створення та публікації статей у багатокористувацькому блозі обрано середовище розробки Visual Studio Code, що характеризується широким функціоналом, який забезпечує ефективну роботу над складними проєктами та значно прискорює процес розробки, є універсальним інструментом для створення серверної логіки, інтеграції баз даних та розробки вебінтерфейсів. Наявність великої кількості плагінів і розширень дозволяє автоматизувати рутинні операції, наприклад, автозавершення коду, форматування, підсвітку синтаксису, налагодження та інтеграцію з системами контролю версій, що особливо важливо при роботі над багатокористувацькими платформами.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Структуру програмного забезпечення для створення та публікації статей у багатокористувацькому блозі наведено на рис. 3.1.



Рисунок 3.1 – Структура програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Як видно з рис. 3.1, структура програмного забезпечення для створення та публікації статей у багатокористувацькому блозі охоплює два взаємопов'язані компоненти: серверну та клієнтську частини, кожна з яких містить по три функціональні модулі.

Серверна частина формується блоком керування користувачами, модулем роботи з контентом та модулем взаємодії з базою даних. Перший модуль відповідає за реалізацію алгоритмів реєстрації та авторизації, забезпечує перевірку облікових даних, управління сесіями та контроль доступу. Другий модуль має призначення підтримувати операції зі створення, редагування та видалення постів, обробляє запити від клієнтів і виконує логіку керування вмістом відповідно до встановлених правил системи. Третій модуль утримує функції збереження, оновлення й пошуку даних у сховищі, забезпечує узгодженість інформації про користувачів і публікації та гарантує цілісність під час виконання операцій над записами.

Клієнтська частина містить модуль інтерфейсу користувача, модуль обробки взаємодії та модуль мережових запитів. Інтерфейсний модуль призначений для відображення засобів роботи з обліковими записами та постами, показує поля введення для реєстрації й авторизації, а також інструменти створення, редагування й видалення статей. Модуль взаємодії формує послідовність дій відповідно до обраних користувачем операцій, перевіряє коректність введених даних і перетворює дії на відповідні запити. Модуль мережової комунікації забезпечує передачу даних на сервер і приймання відповідей, підтримує обмін інформацією під час реєстрації, входу до системи та виконання операцій зі створення, редагування або видалення публікацій.

У такій структурі всі модулі працюють узгоджено, що дозволяє повністю реалізувати процеси, закладені в алгоритмах реєстрації, авторизації та управління постами у багатокористувацькому блозі.

3.2 Функціонування програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Функціонування програмного забезпечення для створення та публікації статей у багатокористувацькому блозі подамо за допомогою діаграми прецедентів, зображеної на рис. 3.2.

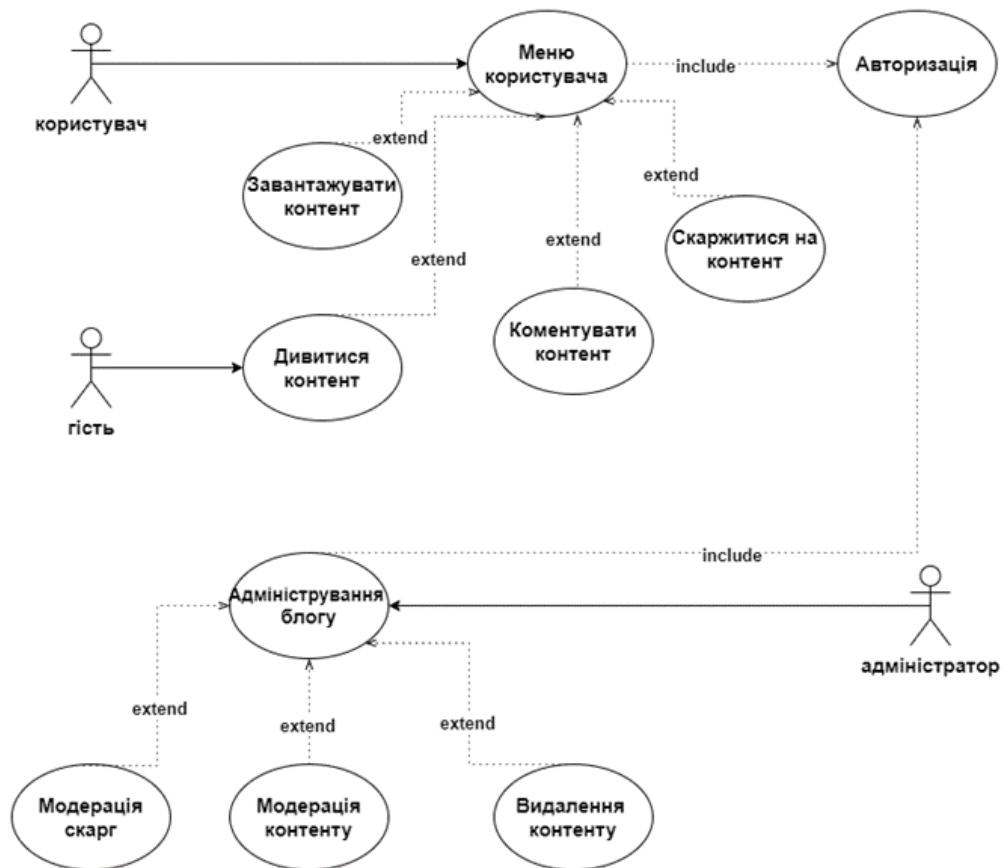


Рисунок 3.2 – Діаграма прецедентів програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Функціонування програмного забезпечення для створення та публікації статей у багатокористувацькому блозі ґрунтується на моделі взаємодії різних типів учасників із системою, кожен з яких отримує доступ до певного кола можливостей. Загальна логіка роботи вибудовується навколо центрального вузла — механізму авторизації, що викликається у всіх випадках, коли передбачається виконання дій, пов'язаних із персоналізованими правами. Усі

потоки взаємодії спрямовуються до відповідних меню та інструментів, які розкриваються залежно від ролі користувача.

Для звичайного користувача взаємодія починається з переходу до особистого меню, яке ініціює механізм перевірки облікових даних. Щойно авторизація успішно завершена, перед ним відкривається можливість працювати з контентом у різних формах. Система розширює базовий набір функцій додатковими можливостями, які увімкнено за потреби. Зокрема, користувачу дозволяється публікувати матеріали, взаємодіяти з уже наявними записами через створення коментарів, надсилати скарги у разі виявлення порушень та переглядати доступні статті. Кожна дія активується як продовження роботи в межах користувацького меню, тому система розглядає їх як розширення одного функціонального простору, у межах якого забезпечується як створення, так і контроль відповідності матеріалів встановленим правилам.

Гість займає найпростіше положення у структурі ролей, оскільки його взаємодія обмежується переглядом матеріалів. У цьому випадку система не ініціює перевірку облікових даних і не активує додаткові механізми керування контентом. Гість взаємодіє з блогом у спостережувальному режимі, що забезпечує відкритість доступу до інформації, але не допускає внесення змін або додавання нових матеріалів.

Адміністратор діє як керівна ланка системи. Його шлях починається з модуля адміністративного управління, який так само звертається до процесу авторизації, оскільки виконання подальших дій потребує найвищого рівня доступу. Після успішного підтвердження повноважень адміністратор отримує можливість контролювати весь блог, стежити за дотриманням правил та забезпечувати коректність взаємодії учасників. Серед доступних розширених можливостей передбачено втручання у вигляді модерації контенту, усунення небажаних або неправомірних матеріалів та обробки скарг, які надходять від користувачів. Постійний зв'язок між адміністративними інструментами та загальною логікою роботи дозволяє підтримувати стабільність системи, а

також оперативно реагувати на ситуації, що можуть впливати на якість блогу та безпеку його учасників.

У такому вигляді вся система функціонує як послідовна взаємодія ролей із відповідними сценаріями поведінки, де кожен елемент діаграми прецедентів відображає логічний перехід між діями, а включення й розширення демонструють залежність між основними процесами й допоміжними операціями. Такий механізм забезпечує цілісність, передбачуваність та керованість роботи програмного забезпечення, створюючи стабільне середовище для публікації, перегляду та модерації статей у багатокористувацькому блозі.

3.3 Розробка бази даних програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

База даних програмного забезпечення для створення та публікації статей у багатокористувацькому побудована таким чином, щоб забезпечити безперервний обмін інформацією між усіма складовими системи та гарантувати узгодженість даних у процесі роботи. Логічна модель організовується навколо кількох ключових сутностей, які відображають учасників, матеріали та взаємодію з контентом. Уся структура спроектована так, щоб забезпечити контроль доступу, можливість створення та редагування матеріалів, фіксацію реакцій користувачів, а також належні інструменти для модерації.

Схему бази даних програмного забезпечення для створення та публікації статей у багатокористувацькому блозі наведено на рис. 3.3.

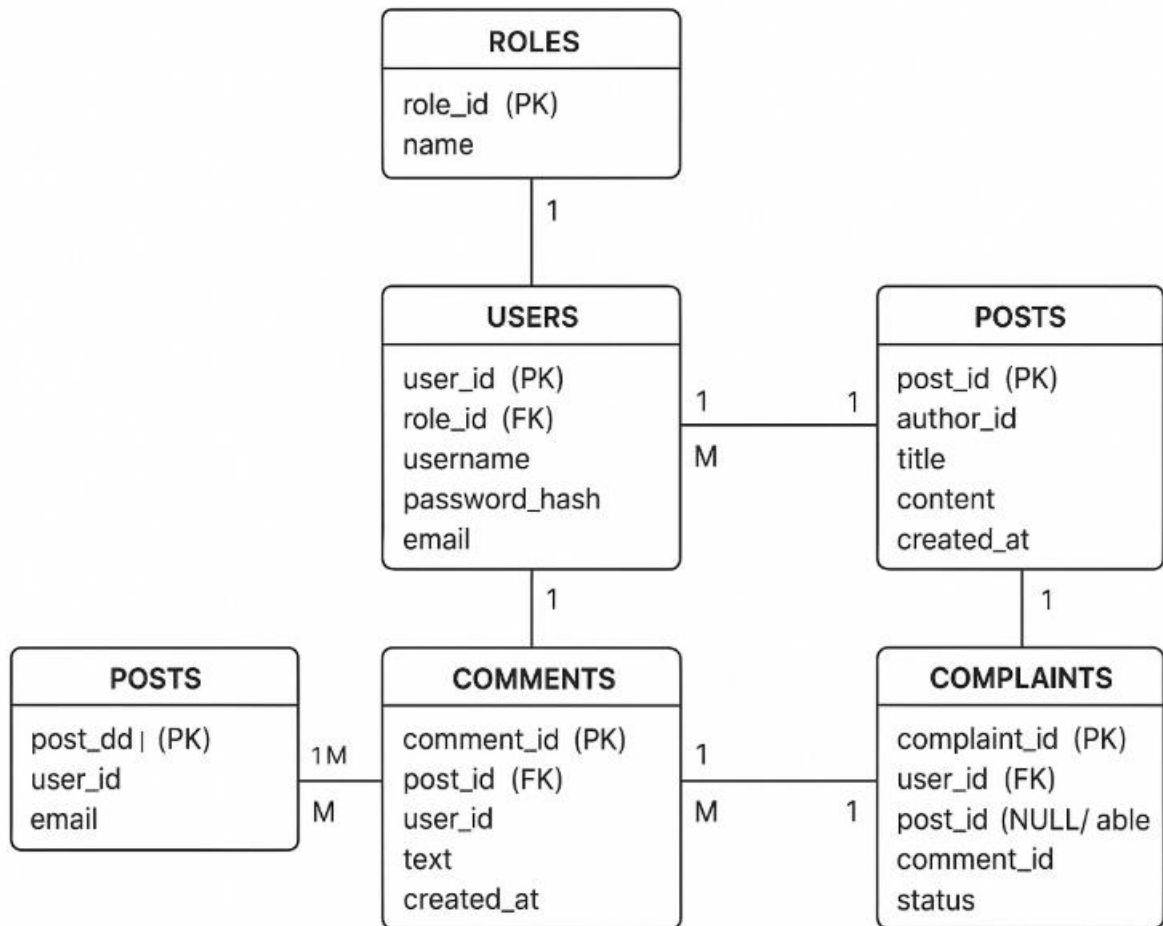


Рисунок 3.3 – Схема бази даних програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

У центрі моделі розташовується таблиця, що відображає інформацію про користувачів. Вона містить унікальний ідентифікатор, який використовується як основа для встановлення зв'язків з іншими сутностями. Поруч з ним зберігаються дані для автентифікації, ролі користувача та параметри, необхідні для розмежування рівнів доступу. Цей ідентифікатор виступає базовою прив'язкою для таблиці постів, де фіксується авторство кожної статті. У таблиці постів розміщується унікальний ключ для кожного матеріалу, а також текстова частина, заголовок, інформація про дату створення та останнє редагування. Зв'язок між постами й користувачами реалізується через зовнішній ключ, що дозволяє однозначно визначити автора кожної публікації.

Поруч із цими сутностями функціонує таблиця коментарів, яка так само містить зовнішні ключі до таблиць користувачів та постів. Це дозволяє пов'язати кожен коментар із конкретним автором і матеріалом. Коментарі зберігаються у вигляді окремих записів, що відображають текст, час створення та можливість подальшого редагування. Вони формують розгалужену структуру, через яку забезпечується активна взаємодія між учасниками системи.

Окреме місце у моделі займає таблиця скарг. Вона прив'язується до постів або коментарів залежно від того, який саме матеріал став об'єктом звернення. Через зовнішні ключі створюється зв'язок із користувачем, який залишив скаргу, а також із матеріалом, що потребує перевірки. У таблиці зберігається текст зауваження, його категорія та поточний статус розгляду. Завдяки цьому забезпечується повний цикл контролю якості та можливість адміністратора своєчасно реагувати на проблемні ситуації.

Останньою важливою сутністю є таблиця ролей, яка використовується для систематизації прав доступу. Вона пов'язана з таблицею користувачів через зовнішній ключ і дозволяє системі відрізнити адміністратора від звичайного користувача або гостя. Це забезпечує виконання різних сценаріїв поведінки згідно з діаграмою прецедентів і формує основу для реалізації алгоритмів авторизації та аутентифікації.

Усі зазначені сутності пов'язані між собою за моделлю «один до багатьох», що дозволяє легко масштабувати систему, додавати нових учасників та збільшувати обсяг матеріалів без порушення внутрішньої логіки. Через такі зв'язки формується цілісна структура даних, у якій кожен елемент має чітко визначене місце і може бути доступний для обробки, модерації та відображення у клієнтській частині програмного забезпечення. Така організація бази даних забезпечує не лише стабільну роботу системи, а й можливість подальшого розширення функцій без потреби у зміні фундаментальних механізмів зберігання інформації.

3.4 Проектування інтерфейсу програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Проектування інтерфейсу взаємодії користувача з програмним забезпеченням для створення та публікації статей у багатокористувацькому блозі виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для створення та публікації статей у багатокористувацькому блозі враховані функціональні вимоги до програми:

- підтримка можливості відображення переліку публікацій у багатокористувацькому блозі;
- підтримка можливості реєстрації та авторизації;
- можливість створення публікацій користувачем;
- можливість редагування та видалення публікацій;
- підтримка можливості користувача з правами адміністратора редагування та видалення публікацій інших користувачів.

Інтерфейс головної сторінки програмного забезпечення для створення та публікації статей у багатокористувацькому блозі наведено на рис. 3.4.

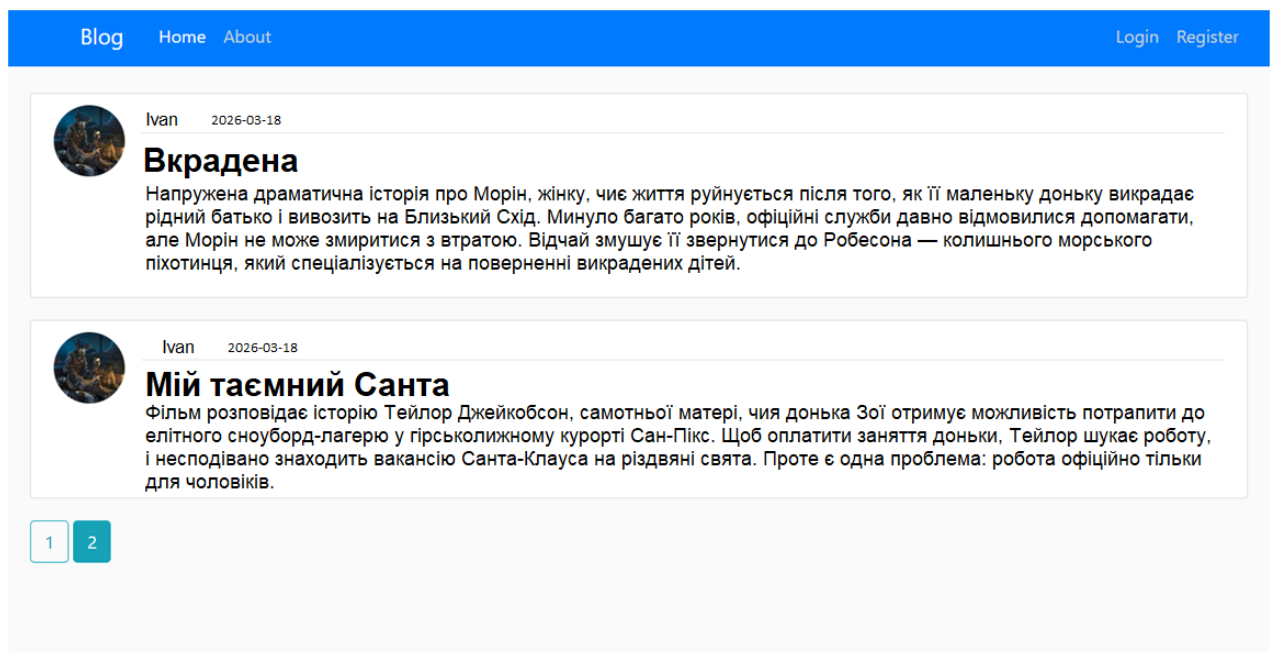


Рисунок 3.4 – Інтерфейс головної сторінки програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Інтерфейс сторінки публікацій користувача наведено на рис. 3.5.

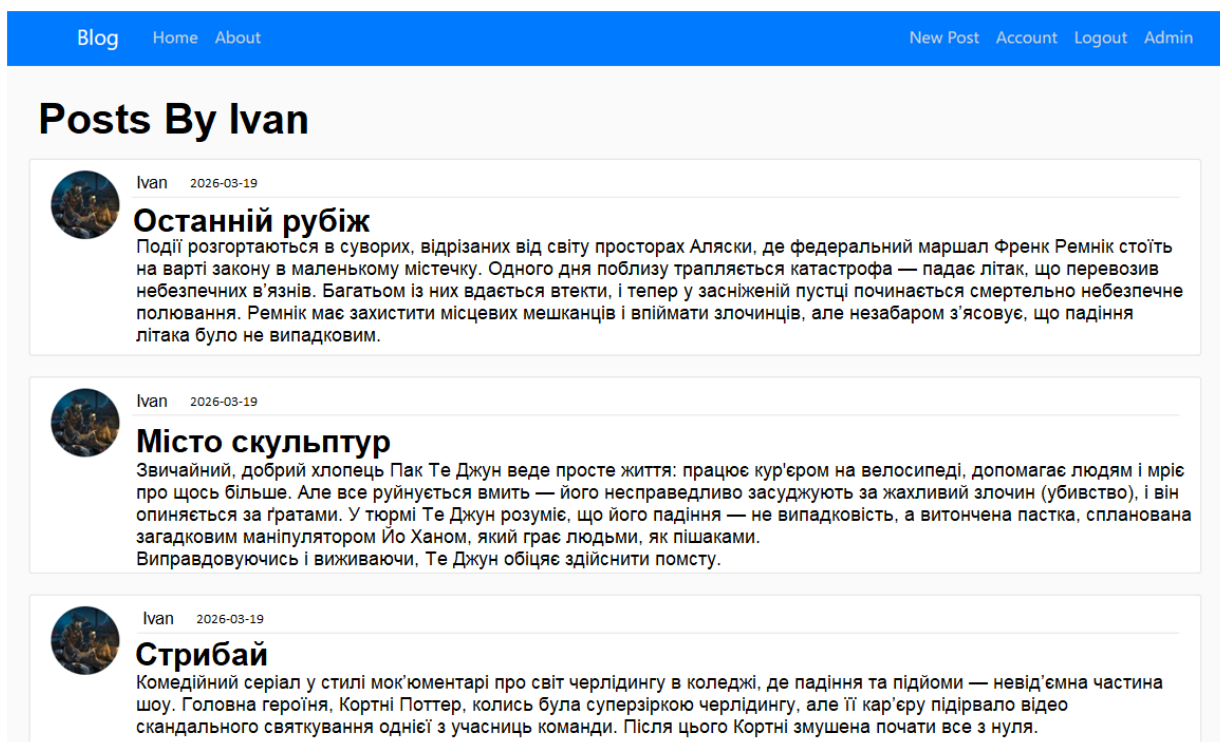


Рисунок 3.5 – Інтерфейс сторінки публікацій користувача

Інтерфейс сторінки створення публікації наведено на рис. 3.6.

The screenshot shows a 'New Post' form. At the top, there is a blue navigation bar with links: 'Blog', 'Home', 'About', 'New Post', 'Account', 'Logout', and 'Admin'. Below the navigation bar, the main heading is 'New Post'. There are two input fields: one for 'Title' and one for 'Content'. Below the 'Content' field, there is a blue button labeled 'Post'.

Рисунок 3.6 – Інтерфейс сторінки створення публікації

Сторінку з прикладом редагування публікації наведено на рис. 3.7.
Сторінку адміністратора наведено на рис. 3.8.

Blog
Home
About
New Post
Account
Logout
Admin

Update Post

Title

Місто скульптур

Content

Пак Те Джун веде просте життя: працює кур'єром на велосипеді, допомагає людям і мріє про щось більше. Але все руйнується вмить — його несправедливо засуджують за жадливий злочин (убивство), і він опиняється за ґратами. У тюрмі Те Джун розуміє, що його падіння — не випадковість, а витончена пастка, спланована

Post

Рисунок 3.7 – Сторінка з прикладом редагування публікації

Blog - Admin						Home Page
Home User Post						
List (7)	Create	With selected ▼				
☐		Title	Date Posted	Content	Author	
☐	 	Останній рубіж	2026-03-19	Події розгортаються в суворих, відрізаних від світу просторах Аляски, де федеральний маршал Френк Ремнік стоїть на варті закону в маленькому містечку. Одного дня поблизу трапляється катастрофа — падає літак, що	Ivan	
☐	 	Місто скульптур	2026-03-19	Звичайний, добрий хлопець Пак Те Джун веде просте життя: працює кур'єром на велосипеді, допомагає людям і мріє про щось більше. Але все руйнується вмить — його несправедливо засуджують за жадливий злочин (убивство), і він опиняється за ґратами. У тюрмі Те Джун розуміє, що його падіння — не випадковість	Ivan	
☐	 	Стрибай	2026-03-19	Комедійний серіал у стилі мок'юментарі про світ черлідінгу в коледжі, де падіння та підйоми — невід'ємна частина шоу. Головна героїня, Кортні Поттер, колись була суперзіркою черлідінгу, але її кар'єру підірвало відео скандального святкування	Ivan	

Рисунок 3.8 – Сторінка адміністратора

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для створення та публікації статей у багатокористувацькому блозі.

Запропоновано структуру програмного забезпечення для створення та публікації статей у багатокористувацькому блозі, що складається з серверної

та клієнтської частин, кожна з яких містить по три функціональні модулі. Серверна частина формується блоком керування користувачами, модулем роботи з контентом та модулем взаємодії з базою даних. Клієнтська частина містить модуль інтерфейсу користувача, модуль обробки взаємодії та модуль мережових запитів. У такій структурі всі модулі працюють узгоджено, що дозволяє повністю реалізувати процеси, закладені в алгоритмах реєстрації, авторизації та управління постами у багатокористувацькому блозі.

Описано функціонування програмного забезпечення для створення та публікації статей у багатокористувацькому блозі, що ґрунтується на моделі взаємодії різних типів учасників із системою, кожен з яких отримує доступ до певного кола можливостей. Програмне забезпечення функціонує як послідовна взаємодія ролей із відповідними сценаріями поведінки, де кожен елемент діаграми прецедентів відображає логічний перехід між діями, а включення й розширення демонструють залежність між основними процесами й допоміжними операціями. Такий механізм забезпечує цілісність, передбачуваність та керованість роботи програмного забезпечення, створюючи стабільне середовище для публікації, перегляду та модерації статей у багатокористувацькому блозі.

Розроблено базу даних програмного забезпечення для створення та публікації статей у багатокористувацькому блозі, що забезпечує безперервний обмін інформацією між усіма складовими системи та гарантувати узгодженість даних у процесі роботи. Логічна модель організована навколо кількох ключових сутностей, які відображають учасників, матеріали та взаємодію з контентом. Уся структура спроектована так, щоб забезпечити контроль доступу, можливість створення та редагування матеріалів блогу, фіксацію реакцій користувачів, а також належні інструменти для модерації.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для створення та публікації статей у багатокористувацькому блозі.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для створення та публікації статей у багатокористувацькому блозі. Програма написана на мові Python, яка забезпечує поєднання швидкості розробки, доступності готових інструментів і високої гнучкості, що дозволяє ефективно реалізувати як базовий функціонал публікації статей, так і складні механізми взаємодії користувачів. В якості середовища розробки для створення програмного забезпечення для створення та публікації статей у багатокористувацькому блозі обрано Visual Studio Code, що характеризується широким функціоналом, який забезпечує ефективну роботу над складними проєктами та значно прискорює процес розробки, є універсальним інструментом для створення серверної логіки, інтеграції баз даних та розробки вебінтерфейсів.

Програмне забезпечення для створення та публікації статей у багатокористувацькому блозі забезпечує виконання таких функцій:

- підтримка можливості відображення переліку публікацій у багатокористувацькому блозі;
- підтримка можливості реєстрації та авторизації;
- можливість створення публікацій користувачем;
- можливість редагування та видалення публікацій;
- підтримка можливості користувача з правами адміністратора редагування та видалення публікацій інших користувачів.

Для роботи програмного забезпечення для створення та публікації статей у багатокористувацькому блозі потрібні такі технічні та програмні ресурси: забезпечення стабільного функціонування системи вимагає наявності обчислювальної потужності, достатньої для обробки запитів користувачів, виконання серверної логіки та підтримання роботи бази даних у реальному часі. Оптимальним вважається використання процесора з тактовою частотою

не нижче середнього сучасного рівня, що дає змогу ефективно опрацьовувати паралельні операції та забезпечує швидкий відгук інтерфейсу. Обсяг оперативної пам'яті має бути достатнім для розміщення серверних служб, кешування запитів та підтримки стабільної роботи вебсервера; зазвичай рекомендується використовувати не менше кількох гігабайт, щоб унеможливити затримки при збільшенні кількості одночасних відвідувачів.

З боку зберігання даних потрібен дисковий простір для розміщення самих застосунків, бази даних і створених користувачами матеріалів. Наявність швидкісного SSD сприяє зменшенню часу доступу до інформації, що позитивно впливає на швидкодію системи. Обов'язковою умовою є стабільне мережеве підключення, оскільки обмін даними між клієнтською та серверною частинами здійснюється через протокол HTTP або HTTPS, і від швидкості з'єднання залежить час відгуку інтерфейсу та доступність функцій у режимі онлайн.

З боку програмного забезпечення потрібні операційна система з підтримкою вебсерверів та запуску програм на мові Python, наявність інструментів для роботи з базами даних, а також можливість встановлення додаткових бібліотек та модулів. На сервері мають працювати платформи, здатні обробляти Python-застосунки (наприклад, середовище виконання Python, сервер WSGI або його аналоги), а також сучасний вебсервер для маршрутизації запитів. На клієнтській частині потрібен сучасний браузер, здатний правильно інтерпретувати HTML, CSS та JavaScript, забезпечувати роботу інтерактивних елементів інтерфейсу та стабільне відображення динамічних даних. Усе програмне середовище повинно бути налаштоване таким чином, щоб підтримувати захищений обмін даними, стійкість до збоїв і можливість подальшого масштабування системи.

Функціональні характеристики розробленого програмного забезпечення для створення та публікації статей у багатокористувацькому блозі наведено у технічному завданні (додаток А). Фрагмент тексту

програмного забезпечення для створення та публікації статей у багатокористувацькому блозі наведено у додатку Б.

4.2 Характеристики програми для створення та публікації статей у багатокористувацькому блозі

Характеристики програмного забезпечення для створення та розповсюдження матеріалів у багатокористувацькому блозі визначаються його здатністю забезпечувати безперервний обмін інформацією між авторами, читачами та адміністраторами. Система повинна працювати таким чином, щоб усі дії користувачів виконувалися без затримок, а структура інтерфейсу залишалася зрозумілою навіть під час активної взаємодії великої кількості учасників. Програмне середовище має підтримувати створення текстових матеріалів, додавання ілюстрацій, роботу з чернетками та можливість подальшого редагування вже опублікованих матеріалів. Важливою властивістю є здатність забезпечувати збереження кожної версії матеріалу, щоб автор міг повертатися до попередніх варіантів або продовжувати роботу над ним у зручний момент.

Система повинна мати механізми контролю доступу, які дозволяють регулювати, хто може створювати нові матеріали, хто має право вносити зміни, а хто відповідає за загальний нагляд за вмістом. Робота з ролями користувачів забезпечує чітку організацію внутрішньої структури, де кожен учасник взаємодіє з платформою у межах своїх повноважень. Важливим елементом є підтримка комунікації між авторами та читачами через коментарі, що сприяє формуванню спільноти й забезпечує двосторонній обмін думками.

Програмне забезпечення повинно мати засоби підтримки стабільності навіть за умов збільшення навантаження, що виникає під час одночасного доступу великої кількості користувачів. Висока надійність роботи забезпечується завдяки можливості автоматично обслуговувати запити, розподіляти ресурси та запобігати конфліктам при оновленні інформації.

Гнучка архітектура дозволяє адаптувати систему до змін у структурі блогу, розширювати функціональність або підключати додаткові модулі без порушення цілісності роботи.

Важливе значення має підтримка безпеки на всіх етапах взаємодії, що включає захист конфіденційних даних, фільтрацію потенційно небезпечного вмісту та запобігання несанкціонованим діям. Також система повинна забезпечувати коректне зберігання матеріалів у незалежному сховищі, що гарантує їх доступність навіть за потреби перенесення програми на іншу інфраструктуру. Завдяки таким характеристикам програмне забезпечення здатне працювати стабільно, підтримувати розвиток контенту й забезпечувати комфортну взаємодію всіх учасників блогу.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для створення та публікації статей у багатокористувацькому блозі на серверній частини необхідно запустити відповідний ру-файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для створення та публікації статей у багатокористувацькому блозі є інформація, що надходить від користувача під час роботи з матеріалом. До неї належать текстові фрагменти статті. Додатково система отримує службові дані, що передаються під час авторизації, редагування матеріалу або взаємодії з іншими елементами блогу.

Вихідними даними програмного забезпечення для створення та публікації статей у багатокористувацькому блозі є сформований контент, який користувач бачить після опублікування або збереження змін. Це можуть бути

відображені матеріали, оновлені версії текстів, а також системні відповіді, що підтверджують успішне виконання певної дії.

4.3.3 Повідомлення

Користувач програмного забезпечення для створення та публікації статей у багатокористувацькому блозі може отримати такі повідомлення: повідомлення про успішне створення матеріалу, повідомлення з інформацією про виявлені помилки, зокрема некоректне введення даних, порушення правил доступу або відсутність підключення до сховища.

4.4 Виконання програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Після запуску програмного забезпечення для створення та публікації статей у багатокористувацькому блозі користувачу відображається головна сторінка (рис. 4.1) зі списком останніх публікацій.

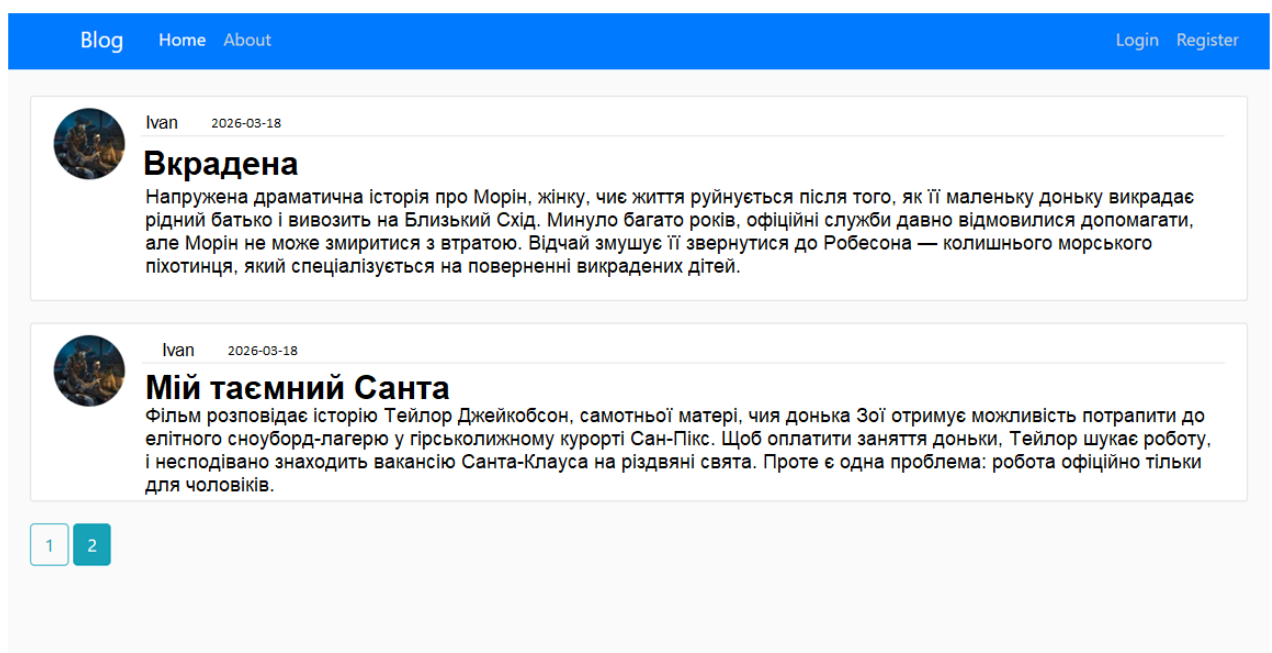
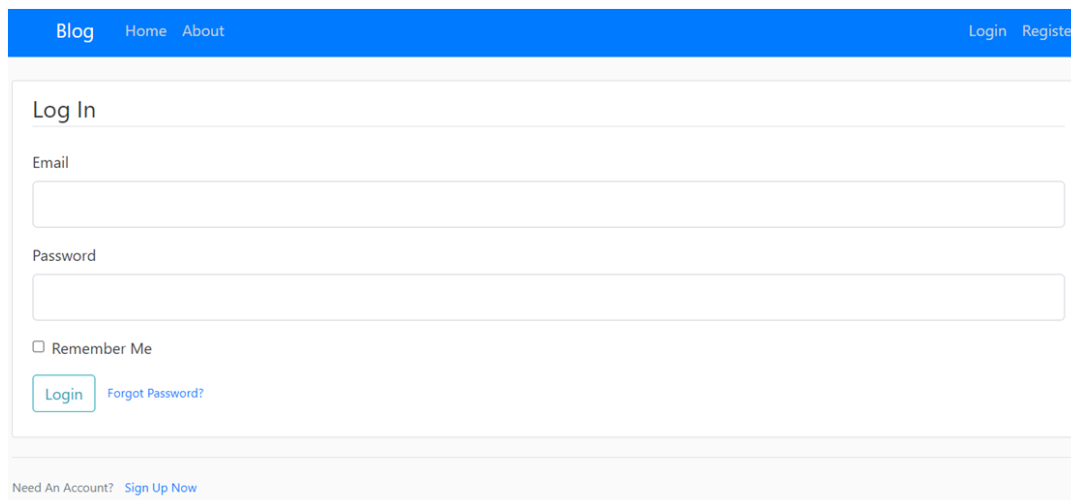


Рисунок 4.1 – Інтерфейс головної сторінки програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

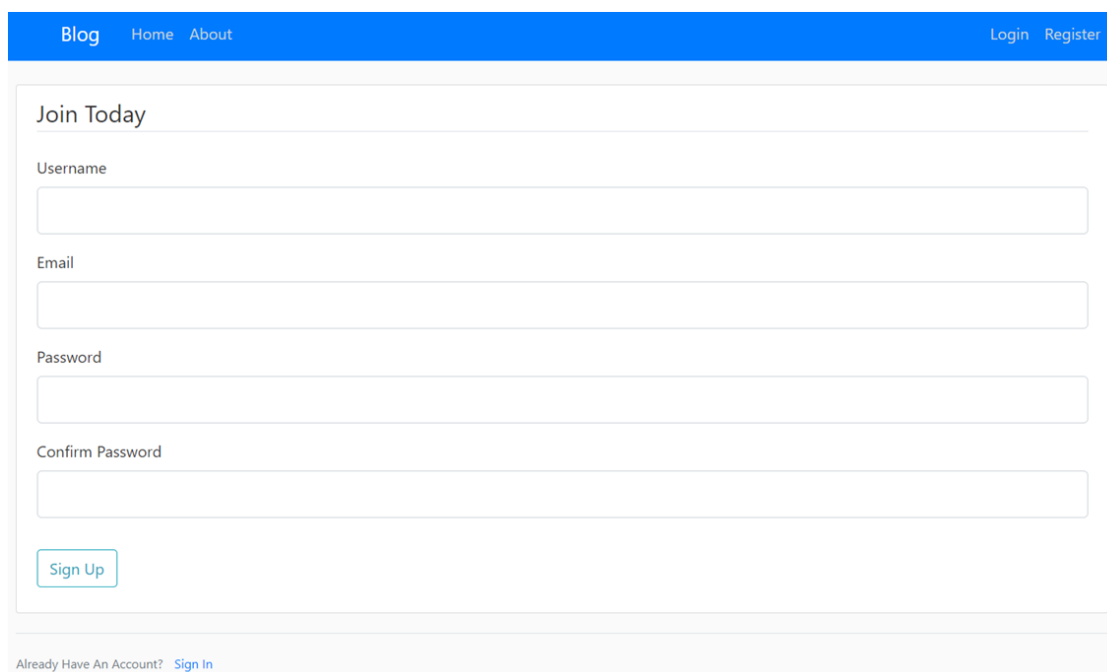
Для входу необхідно перейти за посиланням Login у правому верхньому куті сторінки, після чого у формі (рис. 4.2) ввести електронну адресу та пароль користувача.



The screenshot shows a web application interface with a blue header bar. On the left of the header are links 'Blog', 'Home', and 'About'. On the right are links 'Login' and 'Register'. Below the header is a 'Log In' form. It contains an 'Email' input field, a 'Password' input field, a checkbox labeled 'Remember Me', a 'Login' button, and a 'Forgot Password?' link. At the bottom of the form area, there is a link 'Need An Account? Sign Up Now'.

Рисунок 4.2 – Вхід користувача до системи

Реєстрація (рис. 4.3) відбувається за посиланням Register. У реєстраційній формі необхідно ввести ім'я користувача, електронну пошту, пароль, після чого натиснути кнопку Sign Up.



The screenshot shows a web application interface with a blue header bar. On the left of the header are links 'Blog', 'Home', and 'About'. On the right are links 'Login' and 'Register'. Below the header is a 'Join Today' form. It contains a 'Username' input field, an 'Email' input field, a 'Password' input field, and a 'Confirm Password' input field. At the bottom of the form area, there is a 'Sign Up' button. At the bottom of the page, there is a link 'Already Have An Account? Sign In'.

Рисунок 4.3 – Реєстрація користувача

Після входу до систем користувач бачить сторінку зі своїми публікаціями (рис. 4.4).

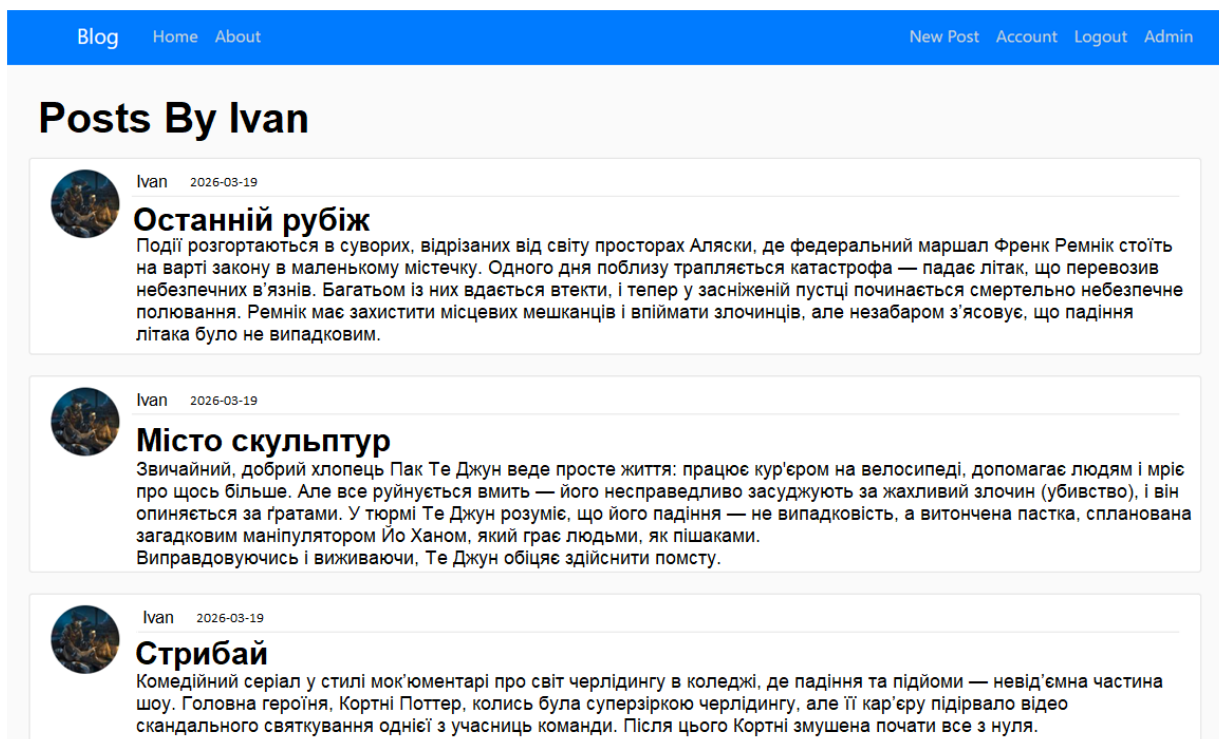


Рисунок 4.4 – Сторінка публікацій користувача

При створення нової публікації необхідно перейти за посиланням New Post та у формі (рис. 4.5) ввести заголовок публікації, а також її зміст, після чого натиснути кнопку Post.

The screenshot shows the 'New Post' form in the same web application. It has a blue header with the same navigation links as the previous page. The form itself is white and contains the following elements:

- A heading 'New Post'.
- A label 'Title' above a text input field.
- A label 'Content' above a larger text area for the post body.
- A blue button labeled 'Post' at the bottom left of the form.

Рисунок 4.5 – Сторінка створення публікації

Форму редагування публікації наведено на рис. 4.6.

BlogHomeAboutNew PostAccountLogoutAdmin

Update Post

Title

Місто скульптур

Content

Пак Те Джун веде просте життя: працює кур'єром на велосипеді, допомагає людям і мріє про щось більше. Але все руйнується вмить — його несправедливо засуджують за жакливий злочин (убивство), і він опиняється за ґратами. У тюрмі Те Джун розуміє, що його падіння — не випадковість, а витончена пастка, спланована

Post

Рисунок 4.6 – Редагування публікації

При вході адміністратора програмного забезпечення для створення та публікації статей у багатокористувацькому блозі відображається Адміністративна панель (рис. 4.7), де адміністратор може відредагувати публікацію або видалити її.

Blog - AdminHomeUserPostHome Page

List (7)CreateWith selected

		Title	Date Posted	Content	Author
☐	 	Останній рубіж	2026-03-19	Події розгортаються в суворих, відрізаних від світу просторах Аляски, де федеральний маршал Френк Ремнік стоїть на варті закону в маленькому містечку. Одного дня поблизу трапляється катастрофа — падає літак, що	Ivan
☐	 	Місто скульптур	2026-03-19	Звичайний, добрий хлопець Пак Те Джун веде просте життя: працює кур'єром на велосипеді, допомагає людям і мріє про щось більше. Але все руйнується вмить — його несправедливо засуджують за жакливий злочин (убивство), і він опиняється за ґратами. У тюрмі Те Джун розуміє, що його падіння — не випадковість	Ivan
☐	 	Стрибай	2026-03-19	Комедійний серіал у стилі мок'юментарі про світ черлідінгу в коледжі, де падіння та підйоми — невід'ємна частина шоу. Головна героїня, Кортні Поттер, колись була суперзіркою черлідінгу, але її кар'єру підірвало відео скандального святкування	Ivan

Рисунок 4.7 – Панель адміністратора

4.5 Тестування програмного забезпечення для створення та публікації статей у багатокористувацькому блозі

Виконано тестування програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.

Виявлено, що програма для створення та публікації статей у багатокористувацькому блозі функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримку діяльності багатокористувацького блогу.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою діяльності багатокористувацького блогу.

4.6 Висновки за розділом 4

Описано програмне забезпечення для створення та публікації статей у багатокористувацькому блозі. Виконано тестування розробленого програмного забезпечення для створення та публікації статей у багатокористувацькому блозі. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою діяльності багатокористувацького блогу.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для багатокористувацького блогу.

Визначено, що програмне забезпечення для створення та публікації статей у багатокористувацькому блозі використовується для організації простору, у якому кілька авторів можуть одночасно працювати над контентом, публікувати власні матеріали та взаємодіяти з редакційною системою. Подібні програмні рішення забезпечують єдину платформу, що об'єднує інструменти для написання текстів, керування доступами, узгодження матеріалів і контролю за їхньою якістю.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для створення та публікації статей у багатокористувацькому блозі. Проте деякі програмні засоби для створення та публікації статей у багатокористувацькому блозі за наявності високого навантаження можуть не забезпечувати прийнятної продуктивності, що ускладнює роботу авторів та може спричинити затримки у публікації матеріалів. Технічні обмеження інколи позначаються і на можливостях редагування: відсутність гнучких інструментів, незручний інтерфейс або надлишок другорядних функцій створює додаткові труднощі під час підготовки текстів. Суттєвим недоліком є й необхідність постійного оновлення програмного забезпечення, що не завжди гарантує бездоганну роботу. Таким чином, попри високий потенціал багатокористувацьких блогових платформ, їх використання супроводжується низкою обмежень, які потребують постійної уваги, належного технічного супроводу та зваженого адміністрування. Тому актуальною є розробка програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.

Сформульовано функціональні вимоги до програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.

Для реалізації програмного забезпечення для створення та публікації статей у багатокористувацькому блозі обрано мову програмування Python, яка забезпечує поєднання швидкості розробки, доступності готових інструментів і високої гнучкості, що дозволяє ефективно реалізувати як базовий функціонал публікації статей, так і складні механізми взаємодії користувачів. Застосування цієї мови забезпечує можливість створення повноцінної вебсистеми значно швидше, ніж у випадку з багатьма іншими технологіями, адже велика кількість бібліотек і фреймворків дає змогу скоротити час на реалізацію серверної логіки, керування даними та інтеграцію додаткових модулів.

Для створення програмного забезпечення для створення та публікації статей у багатокористувацькому блозі обрано середовище розробки Visual Studio Code, що характеризується широким функціоналом, який забезпечує ефективну роботу над складними проєктами та значно прискорює процес розробки, є універсальним інструментом для створення серверної логіки, інтеграції баз даних та розробки вебінтерфейсів. Наявність великої кількості плагінів і розширень дозволяє автоматизувати рутинні операції, наприклад, автозавершення коду, форматування, підсвітку синтаксису, налагодження та інтеграцію з системами контролю версій, що особливо важливо при роботі над багатокористувацькими платформами.

Розроблено програмне забезпечення для створення та публікації статей у багатокористувацькому блозі.

Запропоновано структуру програмного забезпечення для створення та публікації статей у багатокористувацькому блозі, що складається з серверної та клієнтської частин, кожна з яких містить по три функціональні модулі. Серверна частина формується блоком керування користувачами, модулем роботи з контентом та модулем взаємодії з базою даних. Клієнтська частина містить модуль інтерфейсу користувача, модуль обробки взаємодії та модуль мережових запитів. У такій структурі всі модулі працюють узгоджено, що

дозволяє повністю реалізувати процеси, закладені в алгоритмах реєстрації, авторизації та управління постами у багатокористувацькому блозі.

Описано функціонування програмного забезпечення для створення та публікації статей у багатокористувацькому блозі, що ґрунтується на моделі взаємодії різних типів учасників із системою, кожен з яких отримує доступ до певного кола можливостей. Програмне забезпечення функціонує як послідовна взаємодія ролей із відповідними сценаріями поведінки, де кожен елемент діаграми прецедентів відображає логічний перехід між діями, а включення й розширення демонструють залежність між основними процесами й допоміжними операціями. Такий механізм забезпечує цілісність, передбачуваність та керованість роботи програмного забезпечення, створюючи стабільне середовище для публікації, перегляду та модерації статей у багатокористувацькому блозі.

Розроблено базу даних програмного забезпечення для створення та публікації статей у багатокористувацькому блозі, що забезпечує безперервний обмін інформацією між усіма складовими системи та гарантувати узгодженість даних у процесі роботи. Логічна модель організована навколо кількох ключових сутностей, які відображають учасників, матеріали та взаємодію з контентом. Уся структура спроектована так, щоб забезпечити контроль доступу, можливість створення та редагування матеріалів блогу, фіксацію реакцій користувачів, а також належні інструменти для модерації.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для створення та публікації статей у багатокористувацькому блозі.

Виконано тестування розробленого програмного забезпечення для створення та публікації статей у багатокористувацькому блозі. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою діяльності багатокористувацького блогу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. How to Start A Blog. URL: <https://www.theblogstarter.com/> (date of access: 25.04.2026).
2. What Is Blogging In 2026? Definition, Pros & Cons Explained. URL: <https://bloggerspassion.com/what-is-blogging/> (date of access: 25.04.2026).
3. How to Start a Blog. A Step-By-Step Guide. URL: <https://www.elegantthemes.com/blog/business/how-to-start-a-blog> (date of access: 25.04.2026).
4. What is a Blog and How Does it Work. URL: <https://samanthanorth.com/what-is-a-blog> (date of access: 25.04.2026).
5. How to start a blog in 10 steps: a beginner's guide. URL: <https://www.wix.com/blog/how-to-start-a-blog> (date of access: 25.04.2026).
6. What Is A Blog? The Essential Guide For Beginners. URL: <https://bloggingwizard.com/what-is-a-blog/> (date of access: 25.04.2026).
7. How to Start a Blog for Beginner: The Complete 2025 Guide. URL: <https://firstsiteguide.com/start-blog/> (date of access: 25.04.2026).
8. What is a blog? Definition, types, benefits and why you need one. URL: <https://www.wix.com/blog/what-is-a-blog> (date of access: 25.04.2026).
9. Best Blogging Platforms (In-Depth Review). URL: <https://feather.so/blog/best-blogging-platforms> (date of access: 25.04.2026).
10. Best Blogging Tools for Growth. URL: <https://www.semrush.com/blog/blogging-tools/> (date of access: 25.04.2026).
11. Best Blogging Software Options to Consider. URL: <https://www.ryrob.com/best-blogging-software/> (date of access: 25.04.2026).
12. Best Blogging Platforms. URL: <https://www.softwaresuggest.com/blog-platform> (date of access: 25.04.2026).
13. Best Blogging Tools and Software to Use. URL: <https://clickup.com/blog/blogging-tools/> (date of access: 25.04.2026).

14. Best Blogging Tools to Use. URL: <https://www.ryrob.com/blogging-tools/> (date of access: 25.04.2026).
15. Ghost. URL: <https://ghost.org/> (date of access: 25.04.2026).
16. Blogger. URL: <https://www.blogger.com/about/?hl=en> (date of access: 25.04.2026).
17. Social Champ. URL: <https://www.socialchamp.com/> (date of access: 25.04.2026).
18. Python documentation. URL: <https://docs.python.org/3/index.html> (date of access: 25.04.2026).
19. Python Basics: Introduction to Python. URL: <https://realpython.com/learning-paths/python-basics/> (date of access: 25.04.2026).
20. Visual Studio Code tips and tricks. URL: <https://code.visualstudio.com/docs/getstarted/tips-and-tricks> (date of access: 25.04.2026).
21. Dream big. Achieve more. Visual Studio. URL: <https://visualstudio.microsoft.com/> (date of access: 25.04.2026).

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для створення та публікації статей у багатокористувацькому блозі.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення для створення та публікації статей у багатокористувацькому блозі», затверджене наказом Національного університету «Запорізька політехніка» № 139 від 7 квітня 2026 р.

A.2 Призначення розробки

Програмний продукт призначений для підтримки багатокористувацького блогу.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для створення та публікації статей у багатокористувацькому блозі забезпечує виконання таких функцій:

- підтримка можливості відображення переліку публікацій у багатокористувацькому блозі;
- підтримка можливості реєстрації та авторизації;
- можливість створення публікацій користувачем;
- можливість редагування та видалення публікацій;
- підтримка можливості користувача з правами адміністратора редагування та видалення публікацій інших користувачів.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для створення та публікації статей у багатокористувацькому блозі повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

A.3.3 Вимоги до надійності

Програмне забезпечення для створення та публікації статей у багатокористувацькому блозі повинно забезпечити надійне функціонування.

A.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для створення та публікації статей у багатокористувацькому блозі необхідна наявність персонального комп'ютера.

A.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи програмного забезпечення для створення та публікації статей у багатокористувацькому блозі потрібні такі технічні та програмні ресурси: забезпечення стабільного функціонування системи вимагає наявності обчислювальної потужності, достатньої для обробки запитів користувачів, виконання серверної логіки та підтримання роботи бази даних у реальному часі. Оптимальним вважається використання процесора з тактовою частотою не нижче середнього сучасного рівня, що дає змогу ефективно опрацьовувати паралельні операції та забезпечує швидкий відгук інтерфейсу. Обсяг

оперативної пам'яті має бути достатнім для розміщення серверних служб, кешування запитів та підтримки стабільної роботи вебсервера; зазвичай рекомендується використовувати не менше кількох гігабайт, щоб унеможливити затримки при збільшенні кількості одночасних відвідувачів.

А.3.6 Вимоги до маркування і пакування

Програма для створення та публікації статей у багатокористувацькому блозі може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для створення та публікації статей у багатокористувацькому блозі».

А.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для створення та публікації статей у багатокористувацькому блозі є інформація, що надходить від користувача під час роботи з матеріалом. До неї належать текстові фрагменти статті. Додатково система отримує службові дані, що передаються під час авторизації, редагування матеріалу або взаємодії з іншими елементами блогу.

Вихідними даними програмного забезпечення для створення та публікації статей у багатокористувацькому блозі є сформований контент, який користувач бачить після опублікування або збереження змін. Це можуть бути відображені матеріали, оновлені версії текстів, а також системні відповіді, що підтверджують успішне виконання певної дії.

ДОДАТОК Б
Фрагмент тексту програми

```

def init_db(db):
    class RLS(db.Model):
        __tablename__ = 'roles'
        pk = db.Column(db.Integer, primary_key=True)
        nm = db.Column(db.String(50), nullable=False,
unique=True)
    class SRS(db.Model):
        __tablename__ = 'users'
        pk = db.Column(db.Integer, primary_key=True)
        rl_pk = db.Column(db.Integer, db.ForeignKey('roles.pk'),
nullable=False)
        lg = db.Column(db.String(150), nullable=False,
unique=True)
        pss_hsh = db.Column(db.String(256), nullable=False)
        nm = db.Column(db.String(150), nullable=True)
        ml = db.Column(db.String(150), nullable=True)
        crt = db.Column(db.DateTime, default=db.func.now())
        rl = db.relationship('RLS', backref=db.backref('srs',
lazy='dynamic'))
    class PST(db.Model):
        __tablename__ = 'posts'
        pk = db.Column(db.Integer, primary_key=True)
        ath_pk = db.Column(db.Integer, db.ForeignKey('users.pk'))
        tt = db.Column(db.String(300), nullable=False)
        cnt = db.Column(db.Text, nullable=False)
        stts = db.Column(db.String(50), nullable=False,
default='drf')
        crt = db.Column(db.DateTime, default=db.func.now())
        upd = db.Column(db.DateTime, default=db.func.now(),
onupdate=db.func.now())
        ath = db.relationship('SRS', backref=db.backref('psts',
lazy='dynamic'))
    class CMM(db.Model):
        __tablename__ = 'comments'
        pk = db.Column(db.Integer, primary_key=True)
        pst_pk = db.Column(db.Integer, db.ForeignKey('posts.pk'),
nullable=False)
        srs_pk = db.Column(db.Integer, db.ForeignKey('users.pk'),
nullable=False)
        txt = db.Column(db.Text, nullable=False)
        crt = db.Column(db.DateTime, default=db.func.now())
        pst = db.relationship('PST', backref=db.backref('cmms',
lazy='dynamic'))
        srs = db.relationship('SRS', backref=db.backref('cmms',
lazy='dynamic'))
    class CMPLNT(db.Model):
        __tablename__ = 'complaints'

```

```

        pk = db.Column(db.Integer, primary_key=True)
        srs_pk = db.Column(db.Integer, db.ForeignKey('users.pk'),
nullable=False)
        pst_pk = db.Column(db.Integer, db.ForeignKey('posts.pk'),
nullable=True)
        cmm_pk = db.Column(db.Integer,
db.ForeignKey('comments.pk'), nullable=True)
        txt = db.Column(db.Text, nullable=False)
        stts = db.Column(db.String(50), nullable=False,
default='new')
        crt = db.Column(db.DateTime, default=db.func.now())
        srs = db.relationship('SRS',
backref=db.backref('cmplnts', lazy='dynamic'))
        pst = db.relationship('PST',
backref=db.backref('cmplnts', lazy='dynamic'))
        cmm = db.relationship('CMM',
backref=db.backref('cmplnts', lazy='dynamic'))
        db.create_all()
        try:
            admin_rl = RLS.query.filter_by(nm='admin').first()
            if not admin_rl:
                admin_rl = RLS(nm='admin')
                db.session.add(admin_rl)
                db.session.commit()
        except Exception:
            pass
        globals()['RLS'] = RLS
        globals()['SRS'] = SRS
        globals()['PST'] = PST
        globals()['CMM'] = CMM
        globals()['CMPLNT'] = CMPLNT

def chck_pss(pss, pss_hsh):
    return ckh(pss, pss_hsh)
def mk_pss(pss):
    return gnh(pss)
def sgn_in_srs(srs_lg):
    ssn['srs_lg'] = srs_lg
def sgn_ot_srs():
    ssn.pop('srs_lg', None)
def gtr_srs_by_lg(lg):
    return SRS.query.filter_by(lg=lg).first()
def gtr_srs_by_pk(pk):
    return SRS.query.get(pk)
def gtr_pst_by_pk(pk):
    return PST.query.get(pk)
def gtr_cmm_by_pk(pk):
    return CMM.query.get(pk)

```

```

def gtr_cmplnt_by_pk(pk):
    return CMPLNT.query.get(pk)
def crt_srs(lg, pss, nm=None, ml=None, rl_nm='user'):
    rl = RLS.query.filter_by(nm=rl_nm).first()
    if not rl:
        rl = RLS(nm=rl_nm)
        DBmod.session.add(rl)
        DBmod.session.commit()
    pss_hsh = mk_pss(pss)
    srs = SRS(lg=lg, pss_hsh=pss_hsh, nm=nm, ml=ml, rl_pk=rl.pk)
    DBmod.session.add(srs)
    DBmod.session.commit()
    return srs
def lg_srs(lg, pss):
    srs = gtr_srs_by_lg(lg)
    if not srs:
        return None
    if not chck_pss(pss, srs.pss_hsh):
        return None
    sgn_in_srs(srs.lg)
    return srs
def crt_pst(ath_pk, tt, cnt, stts='pub'):
    pst = PST(ath_pk=ath_pk, tt=tt, cnt=cnt, stts=stts)
    DBmod.session.add(pst)
    DBmod.session.commit()
    return pst
def edt_pst(pk, tt=None, cnt=None, stts=None):
    pst = gtr_pst_by_pk(pk)
    if not pst:
        return None
    if tt is not None:
        pst.tt = tt
    if cnt is not None:
        pst.cnt = cnt
    if stts is not None:
        pst.stts = stts
    DBmod.session.add(pst)
    DBmod.session.commit()
    return pst
def dlt_pst(pk):
    pst = gtr_pst_by_pk(pk)
    if not pst:
        return False
    for c in pst.cmms.all():
        DBmod.session.delete(c)
    for cm in pst.cmplnts:
        DBmod.session.delete(cm)
    DBmod.session.delete(pst)
    DBmod.session.commit()

```

```

    return True
def crt_cmm(pst_pk, srs_pk, txt):
    cmm = CMM(pst_pk=pst_pk, srs_pk=srs_pk, txt=txt)
    DBmod.session.add(cmm)
    DBmod.session.commit()
    return cmm
def crt_cmplnt(srs_pk, pst_pk=None, cmm_pk=None, txt=''):
    cmpl = CMPLNT(srs_pk=srs_pk, pst_pk=pst_pk, cmm_pk=cmm_pk,
txt=txt)
    DBmod.session.add(cmpl)
    DBmod.session.commit()
    return cmpl
def gtr_srs_cur():
    lg = ssu.get('srs_lg')
    if not lg:
        return None
    return gtr_srs_by_lg(lg)
def chck_srs_ath(pk):
    srs = gtr_srs_cur()
    if not srs:
        return False
    return srs.pk == pk or srs.rl.nm == 'admin'

def hme():
    psts
    PST.query.filter_by(stts='pub').order_by(PST.crt.desc()).all()
    tpl = """
    <html>
    <head><title>Blog hm</title></head>
    <body>
    <h1>Blog hm</h1>
    <div>
    {% for p in psts %}
    <div>
    <h2><a href="{{ url_for('pst_vw', pk=p.pk) }}">{{ p.tt
    }}</a></h2>
    <div>by {{ p.ath.nm or p.ath.lg }}</div>
    <div>{{ p.cnt[:200] }}...</div>
    </div>
    {% endfor %}
    </div>
    <div><a href="{{ url_for('rg') }}">Reg</a> | <a href="{{
url_for('lg') }}">Log</a></div>
    </body>
    </html>
    """
    return rts(tpl, psts=psts)
def rg_hnd():
    if rq.method == 'GET':

```

```

    tpl = """
    <html><body>
    <h2>Reg</h2>
    <form method="post">
    <label>Login</label><input name="lg" />
    <label>Pss</label><input name="pss" type="password" />
    <label>Nm</label><input name="nm" />
    <label>Ml</label><input name="ml" />
    <button type="submit">Rg</button>
    </form>
    </body></html>
    """
    return rts(tpl)
lg = rq.form.get('lg', '').strip()
pss = rq.form.get('pss', '').strip()
nm = rq.form.get('nm', '').strip()
ml = rq.form.get('ml', '').strip()
if not lg or not pss:
    return rts("<html><body>Err: miss fld</body></html>")
existing = gtr_srs_by_lg(lg)
if existing:
    return rts("<html><body>Err: lg tk</body></html>")
srs = crt_srs(lg, pss, nm=nm or None, ml=ml or None)
return rd(ur('lg'))
def lg_hnd():
    if rq.method == 'GET':
        tpl = """
        <html><body>
        <h2>Log</h2>
        <form method="post">
        <label>Login</label><input name="lg" />
        <label>Pss</label><input name="pss" type="password" />
        <button type="submit">Lg</button>
        </form>
        </body></html>
        """
        return rts(tpl)
lg = rq.form.get('lg', '').strip()
pss = rq.form.get('pss', '').strip()
if not lg or not pss:
    return rts("<html><body>Err: miss fld</body></html>")
srs = lg_srs(lg, pss)
if not srs:
    return rts("<html><body>Err: inv crd</body></html>")
return rd(ur('hm'))
def lg_ot_hnd():
    from tls import sgn_ot_srs as sgn_ot_srs
    sgn_ot_srs()
    return rd(ur('hm'))

```

```

def pst_crt_hnd():
    if rq.method == 'GET':
        tpl = """
        <html><body>
        <h2>CrPst</h2>
        <form method="post">
        <label>Title</label><input name="tt" />
        <label>Cnt</label><textarea name="cnt"></textarea>
        <label>Stts</label><select                                name="stts"><option
value="pub">Public</option><option
value="drf">Draft</option></select>
        <button type="submit">Cr</button>
        </form>
        </body></html>
        """
        return rts(tpl)
    srs = gtr_srs_cur()
    if not srs:
        return rts("<html><body>Err: no srs</body></html>")
    tt = rq.form.get('tt', '').strip()
    cnt = rq.form.get('cnt', '').strip()
    stts = rq.form.get('stts', 'drf').strip()
    if not tt or not cnt:
        return rts("<html><body>Err: miss fld</body></html>")
    pst = crt_pst(srs.pk, tt, cnt, stts=stts)
    return rd(ur('pst_vw', pk=pst.pk))

def pst_dlt_hnd(pk):
    pst = gtr_pst_by_pk(pk)
    if not pst:
        return rts("<html><body>Not found</body></html>")
    if not chck_srs_ath(pst.ath_pk):
        return rts("<html><body>Err: no auth</body></html>")
    ok = dlt_pst(pk)
    if not ok:
        return rts("<html><body>Err del</body></html>")
    return rd(ur('hm'))

def cmm_crt_hnd():
    srs = gtr_srs_cur()
    if not srs:
        return rts("<html><body>Err: no srs</body></html>")
    pst_pk = rq.form.get('pst_pk', '').strip()
    txt = rq.form.get('txt', '').strip()
    try:
        pst_pk_i = int(pst_pk)
    except Exception:
        return rts("<html><body>Err: inv</body></html>")
    if not txt:
        return rts("<html><body>Err: miss txt</body></html>")

```

```

    cmm = crt_cmm(pst_pk_i, srs.pk, txt)
    return rd(ur('pst_vw', pk=pst_pk_i))
def cmpltnt_crt_hnd():
    srs = gtr_srs_cur()
    if not srs:
        return rts("<html><body>Err: no srs</body></html>")
    pst_pk = rq.form.get('pst_pk', '').strip()
    txt = rq.form.get('txt', '').strip()
    try:
        pst_pk_i = int(pst_pk)
    except Exception:
        return rts("<html><body>Err: inv</body></html>")
    if not txt:
        return rts("<html><body>Err: miss txt</body></html>")
    cmplt = crt_cmpltnt(srs.pk, pst_pk=pst_pk_i, txt=txt)
    return rd(ur('pst_vw', pk=pst_pk_i))

def adm_dlt_hnd():
    srs = gtr_srs_cur()
    if not srs or srs.rl.nm != 'admin':
        return rts("<html><body>Err: no adm</body></html>")
    pk = rq.form.get('pk', '').strip()
    try:
        pk_i = int(pk)
    except Exception:
        return rts("<html><body>Err: inv</body></html>")
    ok = dlt_pst(pk_i)
    if not ok:
        return rts("<html><body>Err: del</body></html>")
    return rd(ur('adm_hm'))

def crt_db_tbls(cnn):
    c = cnn.cursor()
    c.execute("CREATE TABLE IF NOT EXISTS usrs (id INTEGER
PRIMARY KEY, nm TEXT, ps TEXT, tk TEXT)")
    c.execute("CREATE TABLE IF NOT EXISTS psts (id INTEGER
PRIMARY KEY, ttl TEXT, cnt TEXT, uid INTEGER)")
    c.execute("CREATE TABLE IF NOT EXISTS cmmnts (id INTEGER
PRIMARY KEY, pid INTEGER, uid INTEGER, txt TEXT)")
    cnn.commit()

def sv_usr(cnn, nm, ps):
    c = cnn.cursor()
    hp = bcrpt.hashpw(ps.encode(), bcrpt.gensalt()).decode()
    c.execute("INSERT INTO usrs (nm, ps, tk) VALUES (?, ?, '')",
(nm, hp))
    cnn.commit()

def ld_usr_by_nm(cnn, nm):

```

```

    c = cnn.cursor()
    c.execute("SELECT id, nm, ps, tk FROM usrs WHERE nm=?",
(nm,))
    return c.fetchone()

def upd_usr_tk(cnn, uid, tk):
    c = cnn.cursor()
    c.execute("UPDATE usrs SET tk=? WHERE id=?", (tk, uid))
    cnn.commit()

def gnr_tk(nm):
    pl = {"usr": nm, "tm": tm.time()}
    sk = "s crt"
    return jws.encode(pl, sk)

def chck_tk(tk):
    sk = "s crt"
    try:
        d = jws.decode(tk, sk)
        return d.get("usr")
    except:
        return None

def crt_ck(rsp, nm, tk):
    rsp.set_cookie("usr", nm)
    rsp.set_cookie("tkn", tk)
    return rsp

def rm_ck(rsp):
    rsp.delete_cookie("usr")
    rsp.delete_cookie("tkn")
    return rsp

def sv_pst(cnn, ttl, cnt, uid):
    c = cnn.cursor()
    c.execute("INSERT INTO psts (ttl, cnt, uid) VALUES (?, ?,
?)", (ttl, cnt, uid))
    cnn.commit()

def ld_psts(cnn):
    c = cnn.cursor()
    c.execute("SELECT id, ttl, cnt, uid FROM psts")
    return c.fetchall()

def ld_pst(cnn, pid):
    c = cnn.cursor()
    c.execute("SELECT id, ttl, cnt, uid FROM psts WHERE id=?",
(pid,))
    return c.fetchone()

```

```

def del_pst(cnn, pid, uid):
    c = cnn.cursor()
    c.execute("DELETE FROM psts WHERE id=? AND uid=?", (pid,
uid))
    cnn.commit()

def edt_pst(cnn, pid, ttl, cnt, uid):
    c = cnn.cursor()
    c.execute("UPDATE psts SET ttl=?, cnt=? WHERE id=? AND
uid=?", (ttl, cnt, pid, uid))
    cnn.commit()

def sv_cmmnt(cnn, pid, uid, txt):
    c = cnn.cursor()
    c.execute("INSERT INTO cmmnts (pid, uid, txt) VALUES (?, ?,
?)", (pid, uid, txt))
    cnn.commit()

def ld_cmmnts(cnn, pid):
    c = cnn.cursor()
    c.execute("SELECT id, pid, uid, txt FROM cmmnts WHERE pid=?",
(pid,))
    return c.fetchall()

def snd_ml(addr, sbj, body):
    srv = smtplib.SMTP("localhost")
    msg = f"Sbjt: {sbj}\n\n{body}"
    srv.sendmail("n-nply@blg", addr, msg)
    srv.quit()

def rst_ps(cnn, nm):
    usr = ld_usr_by_nm(cnn, nm)
    if not usr:
        return None
    nw = rndm.randint(1111, 9999)
    hp = bcrpt.hashpw(str(nw).encode(), bcrpt.gensalt()).decode()
    c = cnn.cursor()
    c.execute("UPDATE usrs SET ps=? WHERE id=?", (hp, usr[0]))
    cnn.commit()
    snd_ml(nm, "Pwd Rst", f"Nw ps: {nw}")
    return True

def db_rds_cnfg_ld():
    cnfg = {}
    pth = 'rds_cfg.env'
    if os.path.exists(pth):
        with open(pth, 'r') as f:
            for ln in f.readlines():

```

```

        prts = ln.strip().split('=')
        if len(prts)==2:
            cnfg[prts[0]] = prts[1]
    return cnfg

def db_pool_init():
    gbl_pool = []
    for _ in range(10):
        gbl_pool.append({"cn":None,"st":0})
    return gbl_pool

def db_pool_gt(pl):
    for itm in pl:
        if itm['st']==0:
            itm['st']=1
            itm['cn']={"id":id(itm)}
            return itm
    return None

def db_pool_rls(pl, cn):
    for itm in pl:
        if itm['cn']==cn:
            itm['st']=0
            itm['cn']=None
            return True
    return False

def tkn_rfrsh(old_tkn):
    if not old_tkn:
        return None
    nw = hsh_tkn(old_tkn + str(time.time()))
    return nw

def usr_rl_asgn(usr_id, rl):
    fl = os.path.join('usr_rls', str(usr_id) + '.txt')
    d = rl
    os.makedirs('usr_rls', exist_ok=True)
    with open(fl,'w') as f:
        f.write(d)
    return True

def usr_rl_rd(usr_id):
    fl = os.path.join('usr_rls', str(usr_id) + '.txt')
    if not os.path.exists(fl):
        return None
    with open(fl,'r') as f:
        return f.read().strip()

def chk_usr_prm(usr_id, act):

```

```
rl = usr_rl_rd(usr_id)
if rl is None:
    return False
if rl=='adm':
    return True
if rl=='wrt' and act in ['crt','edt','plsh']:
    return True
if rl=='rd' and act=='rd':
    return True
return False
```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»

Програмне забезпечення для
створення та публікації статей у
багатокористувацькому блозі

Виконав: Іван ІВАНЮК

ст. групи КНТ-113сп

Керівник: Андрій ОЛІЙНИК

д.т.н., професор НУ «Запорізька політехніка»

Рисунок В.1 – Слайд 1

2

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки програмного забезпечення для багатокористувацького блогу.

Предмет дослідження – програмні засоби для створення та публікації статей у багатокористувацькому блозі.

Метою роботи є розробка програмного забезпечення для створення та публікації статей у багатокористувацькому блозі, що забезпечує зменшення часу публікації статей та підвищення швидкості взаємодії авторів у багатокористувацькому блозі.

Рисунок В.2 – Слайд 2

Завдання дослідження

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

- виконати аналіз предметної області та програмних засобів для створення та публікації статей у багатокористувацькому блозі;
- здійснити проєктування програмного забезпечення для створення та публікації статей у багатокористувацькому блозі;
- створити програмне забезпечення для створення та публікації статей у багатокористувацькому блозі;
- виконати тестування розробленого програмного забезпечення для створення та публікації статей у багатокористувацькому блозі.

Рисунок В.3 – Слайд 3

Порівняння ПЗ для підтримки блогів

Критерій порівняння	Ghost	Blogger	Social Champ
Можливості персоналізації	+	+–	–
Інструменти для роботи з контентом	+	+–	+
Простота використання	+–	+	+–
Наявність інструментів аналітики	+–	–	+
Підтримка мультимедійних матеріалів	+	+	+–
Інтеграції зі сторонніми сервісами	+	+–	+

Рисунок В.4 – Слайд 4

Постановка завдання

При розробці програмного забезпечення для створення та публікації статей у багатокористувацькому блозі необхідно забезпечити такі функціональні вимоги:

- підтримка можливості відображення переліку публікацій у багатокористувацькому блозі;
- підтримка можливості реєстрації та авторизації;
- можливість створення публікацій користувачем;
- можливість редагування та видалення публікацій;
- підтримка можливості користувача з правами адміністратора редагування та видалення публікацій інших користувачів.

Рисунок В.5 – Слайд 5

Порівняння мов програмування

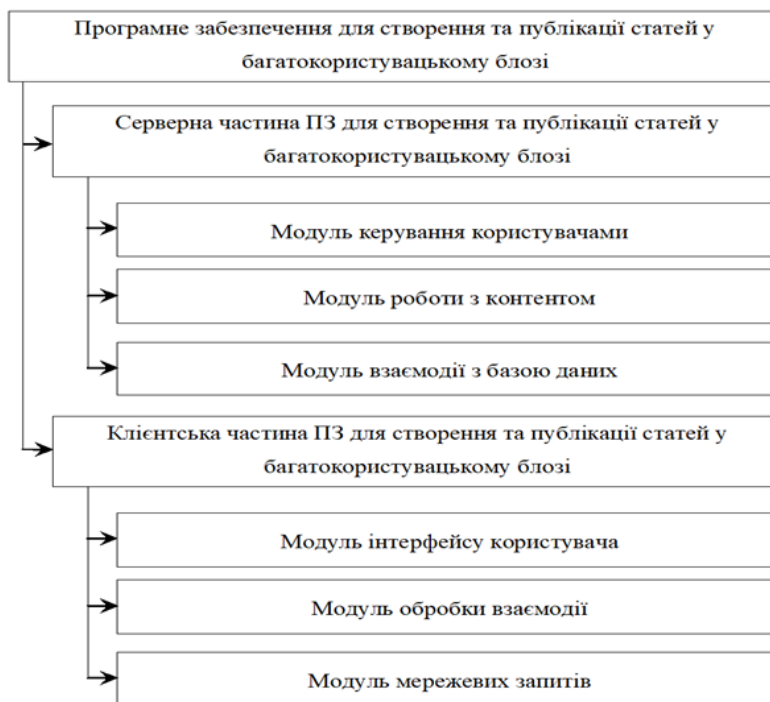
Критерій порівняння мов програмування	Мова програмування		
	Python	PHP	JavaScript
Масштабованість	+	+-	+-
Гнучкість і універсальність	+	+-	+-
Безпека та стабільність	+	+	+-
Зручність для створення ПЗ для створення та публікації статей у багатокористувацькому блозі	+	-	-
Спільнота та документація	+	+-	+

Рисунок В.6 – Слайд 6

Порівняння середовищ розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio Code	Atom	Sublime Text
Налагодження та профілювання	+	+—	+—
Інтеграція з системами контролю версій	+	+	+—
Підтримка командної роботи	+	+—	—
Легкість налаштування	+	+—	+
Простота інтерфейсу та користування	+	+—	+

Рисунок В.7 – Слайд 7



Структура ПЗ для створення та публікації статей у багатокористувацькому блозі

Рисунок В.8 – Слайд 8

Діаграма прецедентів програмного забезпечення

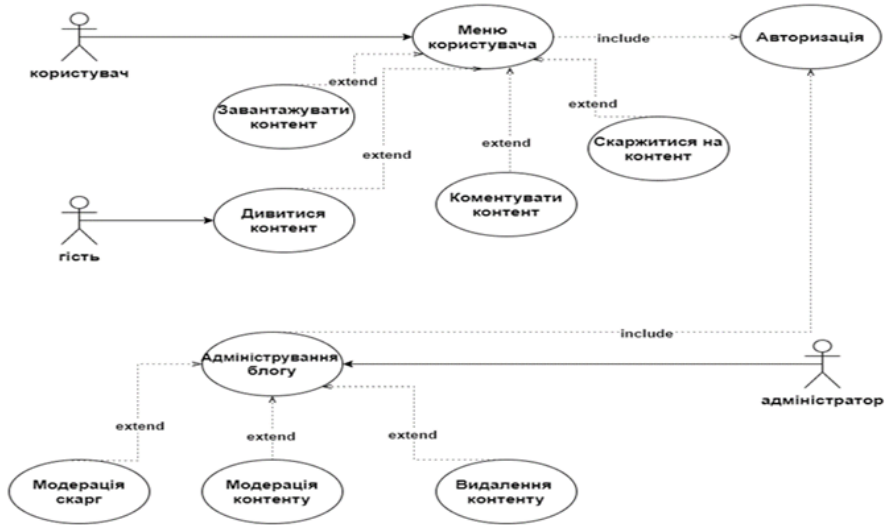


Рисунок В.9 – Слайд 9

Схема бази даних програмного забезпечення

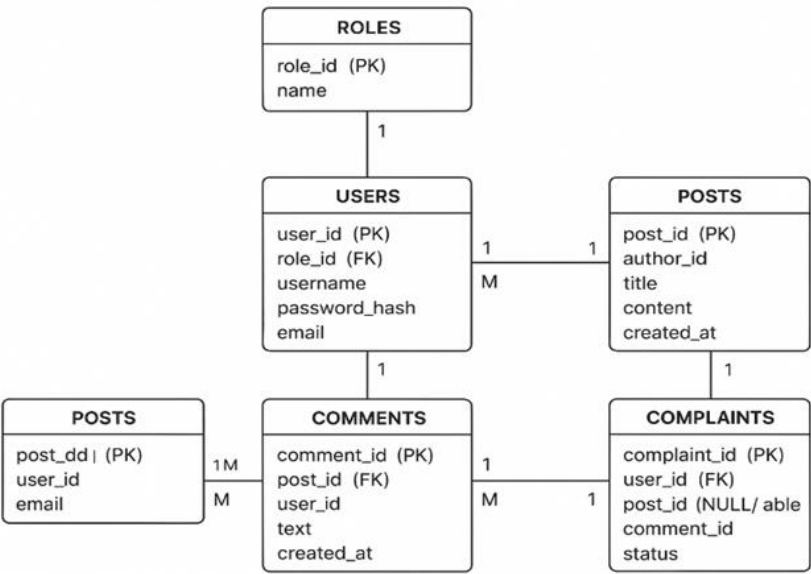


Рисунок В.10 – Слайд 10

Робота з програмою. Головна сторінка. Вхід в систему

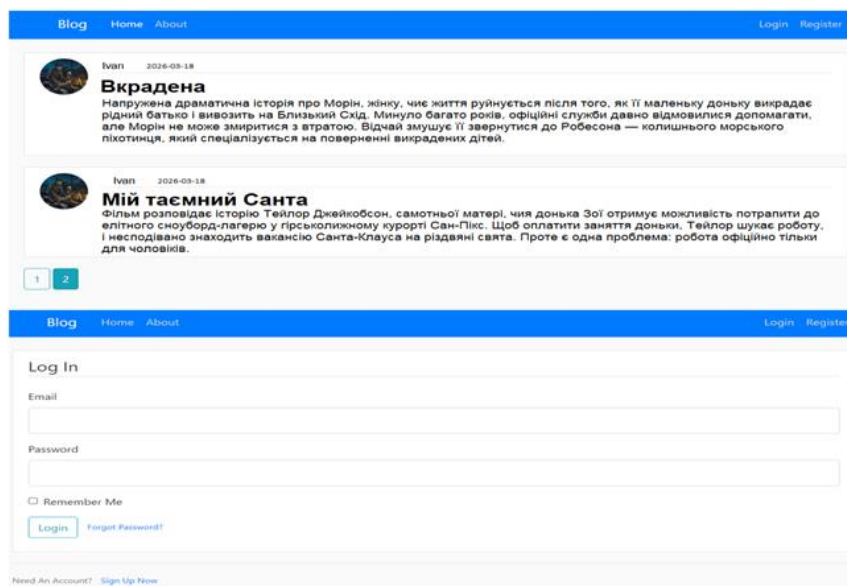


Рисунок В.11 – Слайд 11

Робота з програмою. Додавання та редагування публікацій

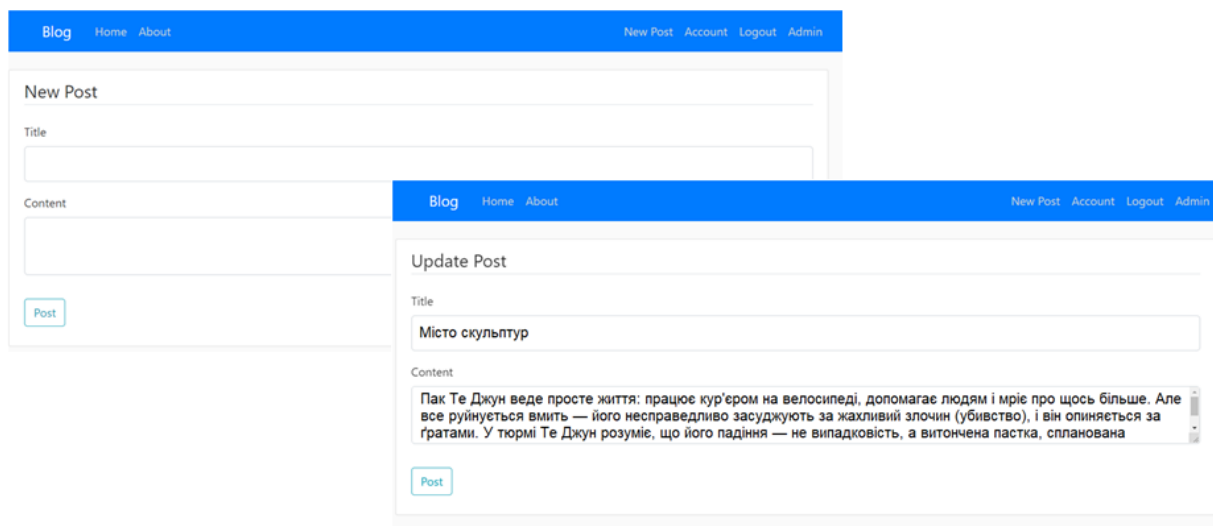


Рисунок В.12 – Слайд 12

Робота з програмою. Адміністративна панель

Blog - Admin

Home

User

Post

Home Page

List (7)

Create

With selected ▼

☐		Title	Date Posted	Content	Author
☐	 	Останній рубіж	2026-03-19	Події розгортаються в суворих, відрізаних від світу просторах Аляски, де федеральний маршал Френк Ремнік стоїть на варті закону в маленькому містечку. Одного дня поблизу трапляється катастрофа — падає літак, що	Ivan
☐	 	Місто скульптур	2026-03-19	Звичайний, добрий хлопець Пак Те Джун веде просте життя: працює кур'єром на велосипеді, допомагає людям і мріє про щось більше. Але все руйнується вмить — його несправедливо засуджують за жорсткий злочин (убивство), і він опиняється за ґратами. У тюрмі Те Джун розуміє, що його падіння — не випадковість	Ivan
☐	 	Стрибай	2026-03-19	Комедійний серіал у стилі мук'юментарі про світ черлідінгу в коледжі, де падіння та підйоми — невід'ємна частина шоу. Головна героїня, Кортні Поттер, колись була суперзіркою черлідінгу, але її кар'єру підірвало відео скандального святкування	Ivan

Рисунок В.13 – Слайд 13

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для багатокористувацького блогу.

Розроблено програмне забезпечення для створення та публікації статей у багатокористувацькому блозі. Запропоновано структуру програмного забезпечення для створення та публікації статей у багатокористувацькому блозі. В результаті виконання завдання було отримано систему, що відповідає всім вимогам представленим в технічному завданні, зручна для використання та виконує всі необхідні функції.

Усі завдання випускної дипломної роботи бакалавра повністю виконано.

Рисунок В.14 – Слайд 14