Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Кібербезпеки_____
(повна назва)

Кафедра _____Інформаційно-мережної інженерії_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____перший (бакалаврський)_____
_____Проектування та розробка веб-платформи інтернет-газети_____
_____
(тема)

Виконав:
здобувач ___4_____ року навчання,
групи ___ТРІМІ-22-1_____
_____Нурметов Андрій_____
(власне ім'я, прізвище)

Спеціальність ____172 Телекомунікації та____
_____радіотехніка_____
(код і повна назва спеціальності)
Тип програми ____освітньо-професійна_____
Освітня програма ___Інформаційно-мережна_
_____ інженерія _____
(повна назва освітньої програми)

Керівник _____ст. викл. Малінін Олександр_
(посада, власне ім'я, прізвище)

Допускається до захисту

в.о. Завідувач кафедри _____ _____ Чеботарьова Дарія_
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Кібербезпеки

Кафедра Інформаційно-мережної інженерії

Рівень вищої освіти перший (бакалаврський)

Спеціальність 172 Телекомунікації та радіотехніка
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформаційно-мережна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:

в.о. Зав. кафедри _____
(підпис)

«____»_____ 20 26 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Нурметову Андрію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Проектування та розробка веб-платформи інтернет-газети

затверджена наказом університету від 28 квітня 2026 р. № 342 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 липня 2026 р.

3. Вихідні дані до роботи
Веб-сервер Apache, HTML, CSS, CSS-фреймворк, Серверна мова, Реляційна або NoSQL бази даних, Інструменти адміністрування бази даних

4. Перелік питань, що потрібно опрацювати в роботі
Вступ
1. Аналіз предметної області
2. Аналіз програмного забезпечення та технологій
3. Проектування веб-платформи інтернет-газети
4. Розробка веб-платформи інтернет-газети
Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Слайди у форматі Power Point (назва, мета роботи, постановка задачі, актуальність роботи, завдання роботи, технології проекту, проектування та розробка веб-платформи, висновки)

_____
_____
_____
_____
_____
_____
_____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім’я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ	ст. викл. Малінін О.П,		05.04.26

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Ознайомлення із завданням. Уточнення ТЗ	*11.02-19.02.26*	виконано
2	Підбір літератури за темою роботи	*20.02-03.03.26*	виконано
3	Виконання розділу 1	*04.03-07.04.26*	виконано
4	Виконання розділу 2	*08.04-06.05.26*	виконано
5	Виконання розділу 3	*07.05-04.06.26*	виконано
6	Виконання розділу 4	*05.06-03.07.26*	виконано
7	Оформлення пояснювальної записки	*04.07-05.07.26*	виконано
8	Оформлення презентаційного матеріалу	*06.07-07.07.26*	виконано
9	Підготовка до захисту у ЕК	*08.07-09.07.26*	виконано

Дата видачі завдання _11__ _лютого_____2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ __ст. викл. Малінін О.П._____
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 71 с., 32 рис., 23 джерел, 1 додаток.

Об'єкт дослідження - процес проектування, програмної реалізації та випробування веб-платформи інтернет-газети «Flash News» для розповсюдження інформації.

Мета роботи - проектування, програмна реалізація та комплексне випробування адаптивної веб-платформи інтернет-газети «Flash News» для забезпечення ефективного розповсюдження інформації, зручної навігації та оптимізації взаємодії з користувачами. Відповідно до поставленої мети визначено такі задачі: дослідити предметну область, проаналізувати схожі роботи, проаналізувати та зробити вибір технології розробки; розробити структуру веб-платформи, базу даних, клієнтський інтерфейс, серверну логіку та тести для перевірки функціональності; розробити клієнтський інтерфейс, реалізувати базу даних і серверну логіку, а також провести випробування веб-платформи.

Результати роботи - проведено аналіз предметної області та існуючих аналогів веб-платформ інтернет-газет, виконано порівняльний аналіз і обґрунтування вибору технологій розробки. Розроблено структуру веб-платформи, базу даних, клієнтський інтерфейс, серверну логіку та випробуванні сценарії.

ABSTRACT

Explanatory note: 71 p., 32 fig., 23 sources, 1 app.

Object of research - the process of designing, developing, implementing, and testing the «Flash News» online newspaper web platform for information dissemination.

Purpose of the work - to design, develop, implement, and comprehensively test the adaptive«Flash News» online newspaper web platform to ensure effective information dissemination, convenient navigation, and optimized user interaction. To achieve this goal, the following tasks were defined: to study the subject area; to analyze existing solutions and select appropriate development technologies; to design the web platform architecture, database, user interface, server-side logic, and functional tests; to implement the user interface, database, and server-side logic; and to conduct comprehensive testing of the web platform.

Results of the work - the subject area and existing online newspaper web platforms were analyzed, and a comparative analysis and justification of the selected development technologies were carried out. The web platform architecture, database, user interface, server-side logic, and test scenarios were developed.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ 7
ВСТУП 8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ 11
1.1 Опис предметної області 11
1.2 Порівняльний аналіз існуючих аналогів та веб-платформ 12
1.3 Обґрунтування доцільності розробки власної веб-платформи 13
1.4 Формулювання вимог до веб-платформи 13
1.5 Постановка завдання дипломного проекту 14
Висновки до розділу 1 14
2 АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ТЕХНОЛОГІЙ 16
2.1 Веб-сервер Apache 16
2.2 Мова розмітки HTML 17
2.3 Каскадні таблиці стилів CSS 17
2.4 Аналіз та вибір CSS-фреймворку 18
2.5 Мова програмування JavaScript 20
2.6 Аналіз та вибір мови програмування для серверної логіки 21
2.7 Аналіз та вибір системи керування базами даних 24
2.8 Мова структурованих запитів SQL 26
2.9 Аналіз та вибір інструментів адміністрування базами даних 26
Висновки до розділу 2 29
3 ПРОЕКТУВАННЯ ВЕБ-ПЛАТФОРМИ ІНТЕРНЕТ-ГАЗЕТИ 30
3.1 Проектування структури веб-платформи 30
3.2 Проектування клієнтського інтерфейсу веб-платформи 31
3.3 Проектування бази даних веб-платформи 37
3.4 Проектування серверної логіки та алгоритмів взаємодії веб-платформи 37
3.5 Розробка випробувальних сценаріїв для перевірки функціональності веб-платформи 39
Висновки до розділу 3 40
4 РОЗРОБКА ВЕБ-ПЛАТФОРМИ ІНТЕРНЕТ-ГАЗЕТИ 41
4.1 Опис апаратного та програмного середовища розробки 41
4.2 Розробка клієнтського інтерфейсу веб-платформи 41
4.3 Реалізація бази даних веб-платформи 45
4.4 Розробка серверної логіки веб-платформи 50

4.5 Тестування веб-платформи 53
Висновки до розділу 4 58
ВИСНОВКИ 59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 60

ПЕРЕЛІК СКОРОЧЕНЬ

БД - база даних;

ЗМІ - засоби масової інформації;

ООП - об'єктно-орієнтоване програмування;

СКБД - система керування базами даних;

CSS (Cascading Style Sheets) - каскадні таблиці стилів;

CMS (Content Management System) - система керування вмістом

DOM (Document Object Model) - об’єктна модель документа;

HTML (HyperText Markup Language) - стандартизована мова розмітки гіпертексту;

PHP (PHP: Hypertext Preprocessor) - рекурсивний акронім «PHP: препроцесор гіпертексту»;

SQL (Structured Query Language) - мова структурованих запитів.

ВСТУП

У сучасному інформаційному суспільстві традиційні друковані засоби масової інформації поступово втрачають провідну роль, поступаючись цифровим аналогам. Ефективність функціонування таких ресурсів безпосередньо залежить від набору технологічних засобів, які були використані під час розробки, адже сучасний читач вимагає не лише якісного контенту, а й швидкого завантаження сторінок, зручної навігації та коректного відображення сайту на будь-яких пристроях. Динамічний розвиток медіасфери спонукає до створення гнучких веб-ресурсів, здатних оперативно доносити інформацію до аудиторії та забезпечувати високий рівень інтерактивності.

Актуальність теми дослідження зумовлена необхідністю розробки та впровадження надійної, швидкої та адаптивної веб-платформи для інтернет-газети. Використання збалансованого набору сучасних технологій дозволяє оптимізувати процес публікації новин, автоматизувати взаємодію з базою даних та гарантувати стабільну роботу ресурсу під час великих навантажень, що є критично важливим для сучасних інформаційних порталів.

Мета роботи - проектування, програмна реалізація та комплексне випробування адаптивної веб-платформи інтернет-газети «Flash News» для забезпечення ефективного розповсюдження інформації, зручної навігації та оптимізації взаємодії з користувачами. Відповідно до поставленої мети визначено такі завдання роботи: аналіз науково-практичних джерел, предметної області та існуючих аналогів веб-платформ інтернет-газет; формування вимог до інформаційної системи; проектування логічної та фізичної архітектури веб-платформи, а також структури бази даних для збереження інформації; реалізація адаптивного клієнтського інтерфейсу платформи на основі фреймворку Bootstrap; розробка серверної логіки для динамічного формування контенту, автоматичного створення сторінок та обробки запитів користувачів; використання інструменту адміністрування бази даних phpMyAdmin та

керування таблицями бази даних; розробка тестових сценаріїв, проведення комплексного випробування функціональності.

Об’єкт роботи - процес проектування, програмної реалізації та випробування веб-платформи інтернет-газети «Flash News» для розповсюдження інформації.

Предмет роботи - технологічні інструменти, архітектурні рішення, методи побудови інтерфейсів та реалізації серверної логіки для забезпечення стабільної й адаптивної роботи веб-платформи інтернет-газети.

Методи дослідження: системний та порівняльний аналіз було використано для дослідження аналогів та вибору технологій; метод логічного моделювання було використано для побудови бази даних; методи клієнтського та серверного веб-програмування було використано для реалізації функціональних можливостей платформи, а також метод тестування за сценаріями для перевірки працездатності готового програмного продукту.

Практичне значення роботи полягає у створенні повністю функціональної, готової до використання веб-платформи інтернет-газети «Flash News». Завдяки інтеграції сучасних веб-технологій (Apache, HTML, CSS, Bootstrap, JavaScript, PHP, MySQL, phpMyAdmin) розроблена система забезпечує автоматичне формування веб-сторінок на основі даних із бази даних, коректне адаптивне відображення на різних пристроях користувачів. Результати роботи можуть бути використані як архітектурна та програмна база для розгортання реальних новинних веб-платформ.

Бакалаврська робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел та додатків:

– У вступі обґрунтовано актуальність обраної теми, визначено мету, завдання, об'єкт, предмет та методи дослідження, а також висвітлено практичне значення отриманих результатів.

– У першому розділі здійснено аналіз предметної області та сучасного стану цифрових медіа, проведено порівняльний аналіз існуючих інтернет-газет,

обґрунтовано доцільність розробки власної системи та сформульовано технічні вимоги до її функціоналу та адаптивності.

– У другому розділі проведено аналіз програмного забезпечення та обґрунтовано вибір технологій для реалізації проекту, зокрема: веб-сервера Apache, фреймворку Bootstrap, мови програмування PHP, СКБД MySQL та інструменту адміністрування бази даних.

– У третьому розділі було виконано безпосереднє проектування веб-платформи інтернет-газети «Flash News»: розроблено логічну структуру системи, сформовано вимоги до дизайну, побудовано структуру реляційної бази даних, визначено алгоритми серверної логіки та сценарії тестування.

– У четвертому розділі було виконано програмну реалізацію веб-платформи інтернет-газети «Flash News»: налаштовано апаратне й програмне середовище розробки, зверстано адаптивний клієнтський інтерфейс, створено та інтегровано реляційну базу даних, запрограмовано серверну логіку й проведено комплексне тестування системи.

– У висновках підбито підсумки виконаної роботи та оцінено ступінь виконання поставлених завдань.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

Головною метою цього розділу є дослідження сучасного ринку цифрових медіа, аналіз тенденцій розвитку онлайн-видань, потреб цільової аудиторії та існуючих аналогів, а також формування технічних вимог до розробки веб-платформи інтернет-газети «Flash News». Отримані результати дослідження слугуватимуть основою для визначення функціональних можливостей, структури, дизайну та технологічних рішень, необхідних для створення сучасного, зручного й ефективного інформаційного веб-ресурсу.

1.1 Опис предметної області

Предметна область розробки веб-платформи інтернет-газети охоплює процеси створення, збереження та поширення цифрового новинного контенту. Сучасні інформаційні ресурси являють собою інтерактивні веб-системи, які поєднують текстові матеріали, мультимедійний контент і засоби взаємодії з користувачами. На відміну від статичних веб-сторінок, такі платформи підтримують динамічне оновлення інформації та забезпечують зручний доступ до новинних матеріалів.

За даними моніторингу веб-трафіку, понад 70% користувачів отримують доступ до новинних ресурсів саме через смартфони. Особливу роль у роботі таких систем відіграють адаптивність інтерфейсу, оптимізація швидкості завантаження та логічна організація навігації. Важливими є також персоналізація контенту, інтеграція з соціальними мережами та рекомендаційні механізми [1].

Важливою складовою роботи онлайн-видань є взаємодія з аудиторією через коментарі, форми зворотного зв'язку, соціальні мережі та інструменти веб-аналітики. Аналіз дій користувачів допомагає оцінювати ефективність контенту та вдосконалювати роботу інформаційного ресурсу.

Отже, сучасні новинні веб-платформи є багатофункціональними інформаційними платформами, ефективність яких визначається використанням сучасних технологій, що забезпечують швидкість роботи, зручність користування, надійність і можливість подальшого розвитку.

1.2 Порівняльний аналіз існуючих аналогів та веб-платформ

Для створення концепції інтернет-газети «Flash News» було проаналізовано архітектурні та інтерфейсні рішення провідних світових медіаплатформ, зокрема The New York Times, BBC News, Reuters і CNN. Кожна з них має власний підхід до організації контенту, навігації та взаємодії з користувачами, що дозволяє визначити найбільш ефективні рішення для створення сучасного новинного ресурсу.

The New York Times активно використовує інтерактивні лонгріди, мультимедійні матеріали та персоналізацію контенту відповідно до інтересів читачів. Разом із тим значна частина публікацій доступна лише за платною підпискою, що обмежує доступ для широкої аудиторії. На відміну від цього, BBC News робить акцент на швидкому завантаженні сторінок, простій структурі розділів і загальній доступності контенту. Завдяки цьому користувачі можуть комфортно переглядати новини навіть за низької швидкості інтернет-з'єднання.

Веб-сайт агентства Reuters вирізняється лаконічним оформленням і мінімальною кількістю графічних елементів. Такий підхід позитивно впливає на швидкість рендерингу сторінок, хоча інтерфейс може здаватися менш привабливим для користувачів, які віддають перевагу мультимедійному контенту. Натомість CNN активно використовує відеоматеріали, великі зображення та інтерактивні елементи. Через значну кількість рекламних і аналітичних скриптів окремі сторінки можуть завантажуватися повільніше, а також іноді спостерігається зміщення елементів макета.

Результати проведеного порівняльного аналізу визначили, що для сучасного новинного ресурсу, яким прагне бути «Flash News», важливими є адаптивний інтерфейс, швидкий пошук, чітка структура матеріалів, оптимізоване завантаження сторінок і зручна система внутрішніх посилань. Саме ці рішення були враховані під час формування технічних вимог до проектування веб-платформи інтернет-газети «Flash News».

1.3 Обґрунтування доцільності розробки власної веб-платформи

Проведений аналіз предметної області та наявних веб-платформ показав, що для реалізації інтернет-газети «Flash News» доцільно створити власну веб-платформу без використання готових систем керування контентом. Такий підхід дозволяє реалізувати необхідний функціонал, уникнути надлишкового програмного коду, оптимізувати архітектуру під конкретні завдання, підвищити швидкість завантаження сторінок і спростити подальший супровід проекту.

Також власна розробка дає змогу самостійно визначати структуру системи, організацію бази даних, принципи роботи інтерфейсу та серверної частини. Це забезпечує більший контроль над архітектурою веб-платформи та дозволяє адаптувати її під конкретні вимоги проекту. Завдяки цьому можна реалізувати адаптивний інтерфейс, зручну навігацію, швидкий пошук інформації та оптимізувати завантаження новин без використання зайвих компонентів.

Додатковою перевагою є використання доступних веб-технологій, що дозволяє зменшити витрати на розробку, спростити супровід системи та забезпечити можливість її подальшого розвитку та відповідність технічним вимогам. Таким чином, розробка «Flash News» з нуля є доцільним рішенням, оскільки дозволяє створити систему, адаптовану до визначених вимог, із можливістю подальшого розширення.

1.4 Формулювання вимог до веб-платформи

На основі аналізу предметної області та існуючих аналогів визначено основні вимоги до веб-платформи «Flash News». Система повинна забезпечувати швидке подання новинного контенту, зручний доступ до інформації та стабільну роботу.

Основними функціональними вимогами є:

– збереження та отримання даних із бази даних;

– автоматичне формування сторінок на основі отриманої інформації;

– відображення списку та повного тексту новин;

– забезпечення навігації між розділами веб-платформи.

Додатковими вимогами є адаптивне відображення на різних пристроях та можливість подальшого розширення функціоналу.

1.5 Постановка завдання дипломного проекту

Метою дипломного проекту є розробка веб-платформи інтернет-газети «Flash News», призначеної для публікації, збереження та перегляду новинного контенту. Основним завданням системи є забезпечення доступу користувачів до інформаційних матеріалів та організація ефективної роботи з новинним контентом.

У межах проекту необхідно реалізувати основні компоненти веб-платформи, що забезпечують роботу користувацького інтерфейсу, серверної логіки та збереження інформації.

Веб-платформа повинна забезпечувати відображення списку новин, перегляд окремих публікацій, зручну навігацію між розділами та коректне відображення на різних типах пристроїв. Архітектура системи має передбачати можливість подальшого розширення функціоналу, зокрема додавання нових можливостей для керування та обробки контенту.

Під час розробки необхідно врахувати вимоги до продуктивності, зручності використання, структури зберігання даних та взаємодії між окремими компонентами веб-платформи. Реалізація системи повинна забезпечити стабільну роботу інтернет-газети «Flash News» та створити основу для її подальшого розвитку.

Висновки до розділу 1

У першому розділі було проведено аналіз предметної області розробки веб-платформи інтернет-газети «Flash News». Розглянуто особливості сучасних цифрових медіа, принципи споживання новинного контенту та значення веб-технологій у роботі онлайн-видань.

Встановлено, що сучасні новинні ресурси є складними інформаційними системами, які повинні забезпечувати швидке поширення інформації, зручну навігацію, адаптивність, структуроване представлення контенту та можливість взаємодії з користувачами.

Порівняльний аналіз існуючих аналогів дозволив визначити основні рішення для створення «Flash News»: простий і зрозумілий інтерфейс, оптимізовану структуру сторінок, швидке завантаження та зручний доступ до новин.

Було обґрунтовано розробку власної веб-платформи без використання готових систем керування контентом, що дозволяє краще контролювати архітектуру, продуктивність і можливість подальшого розширення системи.

На основі проведеного дослідження сформовано основні вимоги до веб-платформи, визначено необхідність створення клієнтської та серверної частин, бази даних MySQL і модулів для роботи з новинним контентом. Отримані результати стали основою для подальшого проектування та розробки інтернет-газети «Flash News».

2 АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ТЕХНОЛОГІЙ

Розробка веб-платформи інтернет-газети «Flash News» потребує правильного вибору програмного забезпечення та технологій, які забезпечують стабільну роботу системи, зручність розробки й можливість подальшого розвитку. У цьому розділі проведено аналіз основних технологій і програмних засобів, що використовуються для створення клієнтської та серверної частин, роботи з базою даних і адміністрування системи.

Для реалізації поставленої мети необхідно визначити оптимальний набір технологій, обрати мови програмування, систему керування базами даних та допоміжні інструменти розробки. Основними критеріями вибору є продуктивність, безпека, простота реалізації, сумісність компонентів і можливість подальшого супроводу.

2.1 Веб-сервер Apache

Веб-сервер є важливим компонентом системи, оскільки забезпечує обробку HTTP-запитів користувачів та передачу необхідних ресурсів. Для реалізації інтернет-газети було обрано Apache, який є одним із найпоширеніших рішень у сфері веб-розробки [2].

Основним завданням Apache є отримання запитів від браузера користувача, їх обробка та формування відповіді у вигляді веб-сторінки або результату виконання серверного коду. Сервер забезпечує взаємодію між клієнтською частиною, PHP-скриптами та базою даних MySQL [2].

До основних переваг Apache належать стабільність роботи, підтримка різних операційних систем, можливість розширення функціональності за допомогою модулів, гнучке налаштування та сумісність із сучасними веб-технологіями [2].

Вибір Apache для розробки «Flash News» є доцільним, оскільки він забезпечує надійну роботу серверної частини, виконання PHP-скриптів та легко інтегрується з пакетом XAMPP для локального тестування системи [3].

2.2 Мова розмітки HTML

HTML є основною мовою розмітки для створення структури веб-сторінок. За допомогою HTML формуються основні елементи інтерфейсу: заголовки, текстові блоки, зображення, посилання, форми та інші компоненти, які відображаються у браузері [4].

Після завантаження HTML-документа браузер аналізує його структуру та створює внутрішнє представлення сторінки, що дозволяє взаємодіяти з елементами через JavaScript [4].

Однією з головних переваг HTML є сумісність із CSS і JavaScript. HTML відповідає за структуру сторінки, CSS використовується для оформлення, а JavaScript забезпечує інтерактивність і динамічну зміну вмісту [4].

Для веб-платформи інтернет-газети «Flash News» використання HTML є необхідним, оскільки ця технологія забезпечує створення зрозумілої структури сторінок, підтримується сучасними браузерами та є основою клієнтської частини системи.

2.3 Каскадні таблиці стилів CSS

CSS використовується для оформлення веб-сторінок та визначення зовнішнього вигляду елементів інтерфейсу. За допомогою CSS задаються кольори, шрифти, розміри, відступи, розташування блоків та інші параметри відображення [5].

Основним принципом CSS є розділення структури документа та його оформлення. HTML відповідає за створення елементів сторінки, а CSS дозволяє керувати їхнім виглядом без зміни основної структури [5].

Важливою можливістю CSS є створення адаптивного дизайну, завдяки якому веб-платформа коректно відображається на пристроях із різними розмірами екранів [5].

Використання CSS у розробці «Flash News» забезпечує створення сучасного користувацького інтерфейсу, зручність перегляду матеріалів та адаптацію веб-платформи під комп’ютери, планшети й смартфони.

2.4 Аналіз та вибір CSS-фреймворку

Для прискорення розробки інтерфейсу та забезпечення адаптивності веб-платформи було проведено аналіз CSS-фреймворків. Використання готових компонентів дозволяє скоротити час створення інтерфейсу та забезпечити єдиний стиль оформлення.

Для порівняння було обрано три популярні CSS-фреймворки: Bootstrap, Foundation та Tailwind CSS. Аналіз цих технологій дозволяє визначити найбільш придатне рішення для створення веб-платформи інтернет-газети [6].

Статистика наведена на (рис 2.1.).

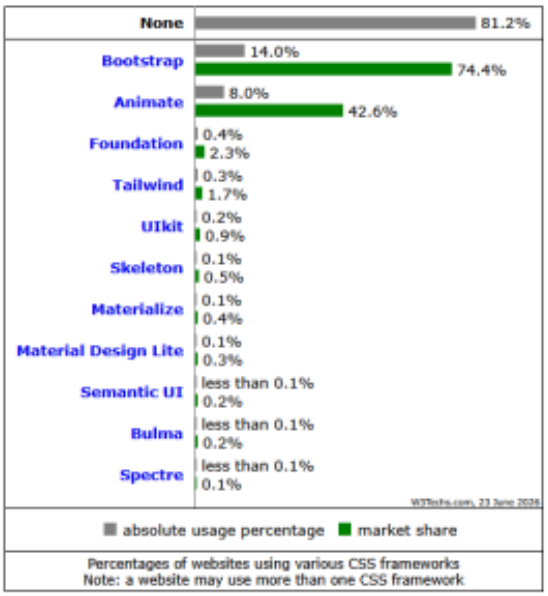


Рисунок 2.1 - Статистика використання CSS-фреймворків

2.4.1 Технічна характеристика фреймворку Bootstrap

Bootstrap є одним із найпоширеніших CSS-фреймворків для створення адаптивних веб-інтерфейсів. Він містить набір готових компонентів, таких як кнопки, форми, навігаційні панелі, картки та інші елементи, що спрощують розробку [7].

Основою Bootstrap є адаптивна сіткова система, яка дозволяє налаштовувати відображення сторінок відповідно до розміру екрана. Також фреймворк містить готові CSS-класи та JavaScript-компоненти для створення інтерактивних елементів [7].

Основними перевагами Bootstrap є простота використання, швидкість розробки, підтримка адаптивного дизайну та широка документація [7].

2.4.2 Технічна характеристика фреймворку Foundation

Foundation є CSS-фреймворком для створення адаптивних веб-інтерфейсів, який орієнтований на гнучке налаштування компонентів та побудову складних веб-проектів [8].

Фреймворк містить адаптивну сіткову систему, готові елементи інтерфейсу, засоби роботи з формами, кнопками, меню та іншими компонентами. Він підтримує сучасні можливості CSS і забезпечує коректне відображення сторінок на різних пристроях [8].

Основною перевагою Foundation є висока гнучкість налаштування, проте для ефективного використання він потребує більшого рівня підготовки розробника порівняно з Bootstrap [8].

2.4.3 Технічна характеристика фреймворку Tailwind CSS

Tailwind CSS є сучасним інструментом стилізації веб-інтерфейсів із підходом Utility-First. Його особливістю є застосування готових службових класів для формування стилів елементів сторінки. Такий підхід забезпечує високий рівень контролю над дизайном і дозволяє створювати унікальні інтерфейси без великої кількості власних CSS-правил. Крім того, Tailwind CSS підтримує адаптивність та оптимізацію кінцевих CSS-стилів. Недоліком цього підходу є збільшення кількості класів у HTML-коді, що може ускладнювати структуру документа під час створення великих проектів [9].

2.4.4 Порівняльний аналіз обраних CSS-фреймворків

Розглянуті CSS-фреймворки мають різні підходи до створення веб-інтерфейсів. Bootstrap орієнтований на швидку розробку завдяки великій кількості готових компонентів. Foundation забезпечує більшу гнучкість налаштування, але потребує складнішого освоєння. Tailwind CSS надає широкі можливості для створення індивідуального дизайну, проте вимагає детальнішої роботи зі стилями.

Проведений аналіз показав, що всі розглянуті фреймворки можуть використовуватися для створення адаптивних веб-інтерфейсів. Однак для веб-платформи інтернет-газети найбільш доцільним є Bootstrap завдяки простоті використання, наявності готових компонентів та швидкості розробки.

2.4.5 Обґрунтування вибору фреймворку Bootstrap для реалізації інтерфейсу

Для створення користувацького інтерфейсу веб-платформи «Flash News» було обрано Bootstrap. Такий вибір зумовлений підтримкою адаптивної верстки,

наявністю готових елементів інтерфейсу та можливістю швидкого створення сучасного дизайну.

Bootstrap забезпечує коректне відображення веб-платформи на різних пристроях, сумісність із сучасними браузерами та спрощує процес розробки завдяки великій кількості готових стилів і компонентів [7].

2.5 Мова програмування JavaScript

JavaScript є однією з основних мов програмування для створення сучасних веб-застосунків і використовується для реалізації динамічної та інтерактивної поведінки веб-сторінок [10]. Основним її призначенням є забезпечення взаємодії користувача з інтерфейсом, обробка подій у браузері та динамічна зміна вмісту HTML-документа без його перезавантаження [10, 11]. Статистика використання JavaScript наведена на (рис. 2.2).

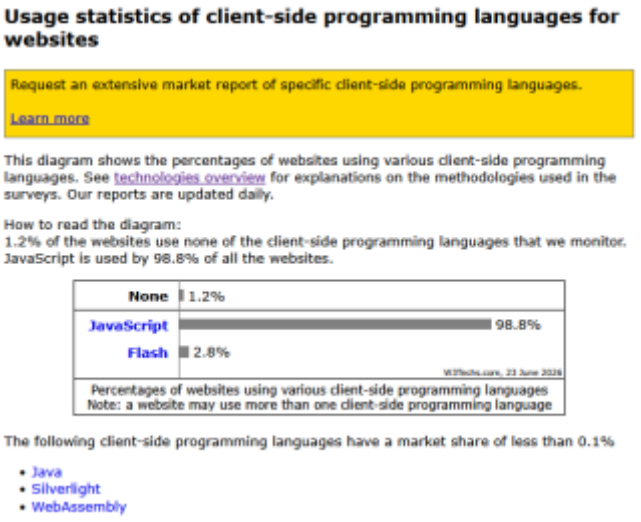


Рисунок 2.2 - Статистика використання мови JavaScript

Однією з головних переваг JavaScript є універсальність і підтримка всіма сучасними браузерами без встановлення додаткового програмного забезпечення. Крім того, мова має розвинену екосистему бібліотек і

фреймворків, що прискорюють розробку та дають змогу повторно використовувати готові модулі [11].

Використання JavaScript є доцільним для розробки веб-платформи інтернет-газети, оскільки ця мова забезпечує інтерактивність інтерфейсу, динамічне оновлення контенту й ефективну взаємодію користувача із системою.

2.6 Аналіз та вибір мови програмування для серверної логіки

Для реалізації серверної логіки веб-платформи інтернет-газети було проведено попередній огляд мов програмування для серверної розробки, серед яких: PHP, Ruby, Java, JavaScript, Scala, C#, Python, ColdFusion, Perl, Haskell, Erlang, Miva Script, C, Smalltalk, C++, Lasso, Go, Tcl, Lisp, BASIC, Lua, Ada [12].

За даними W3Techs, список серверних мов програмування наведено на (рис. 2.3).



Рисунок 2.3 - Список серверних мов програмування за використанням

За результатами попереднього аналізу більшість розглянутих мов програмування було виключено з подальшого розгляду через невідповідність

критеріям оцінювання, таким як продуктивність у веб-середовищі, зручність розробки серверної логіки, підтримка сучасних веб-технологій та наявність активної спільноти розробників.

Для проведення ґрунтовного техніко-економічного порівняння було відібрано три мови програмування, що є найбільш перспективними для подальшого аналізу: PHP, Ruby та Java.

2.6.1 Технічна характеристика мови програмування PHP

PHP є однією з найпоширеніших мов програмування для створення серверної частини веб-застосунків. [13].

Мова виконується на сервері та забезпечує обробку HTTP-запитів, формування динамічних веб-сторінок і взаємодію з базами даних. PHP має простий синтаксис, підтримує об’єктно-орієнтоване програмування та добре інтегрується з MySQL.

Основними перевагами PHP є простота розробки, велика кількість документації, широка підтримка веб-серверів та активна спільнота розробників [13].

2.6.2 Технічна характеристика мови програмування Ruby

Ruby - інтерпретована мова програмування загального призначення з динамічною типізацією, орієнтована на швидку розробку завдяки лаконічному синтаксису. Вона повністю підтримує об'єктно-орієнтоване програмування, автоматичне керування пам’яттю та інтеграцію з сучасними фреймворками, що скорочує час створення серверної частини застосунку [14].

2.6.3 Технічна характеристика мови програмування Java

Java є об'єктно-орієнтованою мовою, призначеною для створення надійних і високонавантажених систем. Використання віртуальної машини Java Virtual Machine (JVM) забезпечує кросплатформеність, а статична типізація, розвинені механізми безпеки та завдяки високій продуктивності вона є оптимальним вибором для великомасштабних проектів [15].

2.6.4 Порівняльний аналіз обраних мов програмування

PHP, Ruby та Java можуть використовуватися для створення серверної логіки веб-застосунків, однак мають різні особливості.

PHP добре підходить для веб-розробки завдяки простій інтеграції з HTML, підтримці баз даних та широкому використанню у веб-середовищі. Ruby забезпечує високу швидкість розробки завдяки простому синтаксису, проте використовується рідше у традиційних веб-проектах. Java характеризується високою продуктивністю та масштабованістю, але потребує складнішої структури розробки.

Для створення веб-платформи інтернет-газети важливими є простота реалізації, швидкість розробки та зручна робота з базою даних, тому PHP є найбільш відповідним рішенням.

2.6.5 Обґрунтування вибору мови програмування PHP для серверної частини

За результатами аналізу для реалізації серверної частини веб-платформи було обрано PHP. Такий вибір зумовлений простою інтеграцією з HTML, високою продуктивністю у веб-середовищі та широкими можливостями роботи з базами даних і HTTP-запитами. Крім того, PHP має велику спільноту

розробників, якісну документацію та значну кількість готових рішень, що спрощує розробку й подальший розвиток системи [13].

2.7 Аналіз та вибір системи керування базами даних

База даних є одним із основних компонентів веб-платформи інтернет-газети, оскільки забезпечує збереження та обробку інформації про новини. Для вибору оптимального рішення було розглянуто три системи керування базами даних: MySQL, PostgreSQL та MongoDB.

Основними критеріями оцінювання були продуктивність, зручність інтеграції з PHP, надійність збереження даних та можливість подальшого розширення системи.

2.7.1 Технічна характеристика системи керування базою даних MySQL

MySQL є реляційною системою керування базами даних, яка широко використовується у веб-розробці для зберігання структурованих даних. Вона підтримує мову SQL і забезпечує швидку обробку запитів у клієнт-серверній архітектурі [16]. До її основних переваг належать проста інтеграція з PHP, невисокі вимоги до системних ресурсів, зручне адміністрування, підтримка транзакцій, індексування та ефективної роботи з великими обсягами структурованих даних, що дає змогу ефективно масштабувати систему [16].

2.7.2 Технічна характеристика системи керування базою даних PostgreSQL

PostgreSQL - потужна об'єктно-реляційна система керування базами даних, орієнтована на роботу зі складними структурами даних. Вона підтримує розширений SQL, користувацькі типи даних і складні запити, тому широко застосовується у великих інформаційних системах [17].

Основними перевагами PostgreSQL є висока надійність, розширюваність, підтримка транзакцій ACID, багатоверсійності (MVCC) та механізмів контролю цілісності даних, що робить її придатною для масштабних проектів [17].

2.7.3 Технічна характеристика системи керування базою даних MongoDB

MongoDB належить до NoSQL-систем і використовує документно-орієнтовану модель даних. Замість таблиць застосовуються JSON-подібні документи (BSON), що забезпечує гнучкість структури даних [18].

Такий підхід є ефективним для проектів із динамічним контентом або горизонтальним масштабуванням. Водночас MongoDB менш придатна для систем зі складними зв'язками між сутностями, оскільки не використовує класичну реляційну модель [18].

2.7.4 Порівняльний аналіз обраних систем керування базами даних

MySQL, PostgreSQL та MongoDB мають різні підходи до зберігання та обробки інформації.

MySQL є простим і ефективним рішенням для більшості веб-застосунків, PostgreSQL надає розширені можливості для складних систем, а MongoDB забезпечує гнучку структуру даних [16, 17].

Для веб-платформи інтернет-газети важливими є структуроване зберігання новин, швидке виконання запитів та проста інтеграція із серверною частиною, тому MySQL є найбільш оптимальним вибором.

2.7.5 Обґрунтування вибору системи керування базою даних MySQL

Для реалізації бази даних веб-платформи «Flash News» було обрано MySQL.

Вибір зумовлений високою продуктивністю, підтримкою реляційної структури даних, зручною інтеграцією з PHP та простотою адміністрування. MySQL забезпечує необхідну функціональність для зберігання новин, виконання запитів та підтримки стабільної роботи веб-платформи [17].

2.8 Мова структурованих запитів SQL

Мова структурованих запитів SQL є стандартом для роботи з реляційними базами даних. SQL забезпечує роботу з реляційною базою даних MySQL, яка використовується у веб-платформі [19].

Перевагою SQL є її декларативний підхід: розробник описує бажаний результат, а система визначає спосіб його отримання. Підтримуються складні операції з’єднання таблиць, фільтрації, групування та сортування даних [19].

SQL забезпечує контроль цілісності даних, підтримку транзакцій та захист інформації від змін. Це гарантує ефективну взаємодію з базою даних та стабільну обробку інформації веб-платформи інтернет-газети [19].

2.9 Аналіз та вибір інструментів адміністрування базами даних

Ефективне адміністрування бази даних є важливою складовою процесу розробки веб-платформи, оскільки забезпечує створення, редагування та контроль структури бази даних, а також виконання резервного копіювання й моніторингу її роботи. Для вибору оптимального інструменту було розглянуто три популярні засоби адміністрування: phpMyAdmin, HeidiSQL та MySQL Workbench.

2.9.1 Технічна характеристика інструменту phpMyAdmin

phpMyAdmin є веб-орієнтованим інструментом адміністрування системи керування базами даних MySQL та MariaDB. Він надає графічний інтерфейс

для виконання більшості адміністративних операцій без необхідності використання командного рядка [20].

За допомогою phpMyAdmin можна створювати та редагувати бази даних і таблиці, виконувати SQL-запити, імпортувати й експортувати дані, керувати користувачами та правами доступу, а також здійснювати резервне копіювання. Основними перевагами інструменту є простота використання, доступ через веб-браузер, інтеграція з локальними веб-серверами та широкий набір функцій для адміністрування баз даних [20].

2.9.2 Технічна характеристика інструменту HeidiSQL

HeidiSQL є безкоштовним настільним інструментом для адміністрування MySQL, MariaDB, PostgreSQL та Microsoft SQL Server [21]. Програма забезпечує зручний графічний інтерфейс для роботи зі структурою баз даних і їхнім вмістом. Серед основних можливостей HeidiSQL - створення й редагування таблиць, виконання SQL-запитів, імпорт та експорт даних, керування користувачами й перегляд серверної інформації. До переваг належать висока швидкодія, підтримка кількох одночасних підключень і невисокі вимоги до системних ресурсів [21].

2.9.3 Технічна характеристика інструменту MySQL Workbench

MySQL Workbench є офіційним інструментом компанії Oracle Corporation для адміністрування та проектування баз даних MySQL [22]. Він поєднує засоби моделювання структури баз даних, адміністрування серверів і розробки SQL-запитів.

Інструмент підтримує створення ER-діаграм, керування користувачами, моніторинг продуктивності, резервне копіювання, відновлення даних та виконання SQL-запитів. Основними перевагами MySQL Workbench є широкі

функціональні можливості, інтеграція із сервером MySQL та підтримка проектування структури баз даних [22].

2.9.4 Порівняльний аналіз обраних засобів адміністрування

Усі три інструменти забезпечують повноцінне адміністрування баз даних MySQL, однак відрізняються способом використання та функціональними можливостями. phpMyAdmin працює через веб-браузер і не потребує встановлення окремого програмного забезпечення, що робить його зручним для локальної розробки та швидкого адміністрування. HeidiSQL є настільною програмою, яка забезпечує високу швидкодію та підтримує кілька систем керування базами даних. MySQL Workbench має найширші можливості, зокрема засоби проектування структури баз даних і моніторингу сервера, однак потребує більше системних ресурсів.

Проведений аналіз показує, що кожен із розглянутих інструментів має власні переваги. phpMyAdmin характеризується простотою використання, HeidiSQL - високою швидкодією, MySQL Workbench - найбільшою функціональністю [23].

2.9.5 Обґрунтування вибору інструменту phpMyAdmin для адміністрування БД

За результатами проведеного аналізу для адміністрування бази даних веб-платформи інтернет-газети було обрано phpMyAdmin. Такий вибір зумовлений простотою використання, доступністю через веб-браузер та повною сумісністю із системою керування базами даних MySQL.

phpMyAdmin забезпечує виконання всіх необхідних операцій із базою даних, зокрема створення таблиць, виконання SQL-запитів, імпорт і експорт даних, резервне копіювання та керування користувачами [20].

Висновки до розділу 2

У даному розділі проведено підбір та обґрунтовано вибір програмного забезпечення та технологій, які були обрані для розробки веб-платформи інтернет-газети.

Для реалізації клієнтської частини було використано мову розмітки HTML5, каскадні таблиці стилів CSS3, CSS-фреймворк Bootstrap для забезпечення адаптивного дизайну та кросбраузерної сумісності. Для створення інтерактивних елементів інтерфейсу було обрано мову програмування JavaScript.

Для реалізації серверної логіки та взаємодії з базою даних було обрано мову програмування PHP та систему керування базою даних MySQL, адміністрування якої здійснюється за допомогою інструменту phpMyAdmin.

Обрані засоби реалізації базуються на результатах порівняльного аналізу та є доцільними для реалізації поставлених завдань, оскільки забезпечують швидку розробку, простоту розгортання, широку доступність та достатній рівень функціональності для веб-платформи.

3 ПРОЕКТУВАННЯ ВЕБ-ПЛАТФОРМИ ІНТЕРНЕТ-ГАЗЕТИ

У цьому розділі наведено проектування веб-платформи інтернет-газети «Flash News». Виконано розробку структури веб-платформи, спроектовано клієнтський інтерфейс, визначено структуру бази даних та особливості серверної логіки. Також розроблено випробувальні сценарії для перевірки працездатності основних функцій веб-платформи. Результати проектування є основою для подальшої програмної реалізації системи та забезпечують відповідність веб-платформи визначеним функціональним і технічним вимогам.

3.1 Проектування структури веб-платформи

Проектування структури веб-платформи інтернет-газети «Flash News» є важливим етапом розробки, оскільки визначає організацію сторінок, навігацію та взаємодію користувача із системою. Структура розроблена з урахуванням вимог сучасних веб-систем: логічного розташування інформації, адаптивності інтерфейсу та швидкого доступу до новинних матеріалів.

Веб-платформа містить такі основні розділи:

– головна сторінка - відображає останні новини, навігаційне меню та посилання для переходу до сторінки новин;

– сторінка «Про нас» - містить інформацію про веб-платформу та контактні дані;

– сторінка новин - відображає перелік новин із бази даних із зазначенням заголовка, тексту, автора та дати публікації. Також реалізовано пошук новин за датою та посторінковий перегляд результатів;

– сторінка окремої новини - призначена для перегляду повного тексту матеріалу. Дані отримуються з бази даних за ідентифікатором новини.

Ієрархічну структуру веб-платформи наведено на (рис. 3.1).



Рисунок 3.1 - Структура веб-платформи

Взаємодія компонентів системи реалізована за клієнт-серверною архітектурою: клієнтська частина відповідає за відображення інтерфейсу, а серверна - за обробку запитів, взаємодію з базою даних MySQL та формування динамічного контенту за допомогою мови програмування PHP.

Розроблена структура забезпечує логічну організацію веб-платформи та створює основу для реалізації користувацького інтерфейсу, серверної логіки й роботи з базою даних.

3.2 Проектування клієнтського інтерфейсу веб-платформи

Проектування клієнтського інтерфейсу є важливим етапом розробки веб-платформи, оскільки саме він забезпечує взаємодію користувача із системою. Під час проектування основну увагу було приділено простоті використання, зручній навігації, адаптивності та логічному розташуванню елементів інтерфейсу. Це дає змогу користувачам швидко знаходити необхідну інформацію та комфортно працювати з веб-платформою незалежно від типу пристрою.

Інтерфейс веб-платформи побудований у єдиному стилі оформлення із використанням HTML, CSS та фреймворку Bootstrap. Для всіх сторінок використовується спільний шаблон, який містить шапку сайту, навігаційне меню, основну область для відображення інформації та нижній колонтитул. У шапці сайту розміщено логотип веб-платформи та меню навігації, що забезпечує швидкий перехід до головної сторінки, сторінки новин і сторінки «Про нас». Нижній колонтитул містить назву веб-платформи та інформацію про авторські права.

Головна сторінка містить інформаційний блок із коротким описом веб-платформи та кнопкою переходу до сторінки новин. Нижче розміщено блок популярних новин, які автоматично завантажуються з бази даних і відображаються у вигляді карток. Кожна картка містить зображення, дату публікації, заголовок, короткий фрагмент тексту та кнопку переходу до повної статті.

Сторінка новин призначена для перегляду всіх публікацій, що зберігаються у базі даних. Новини відображаються у вигляді списку із зазначенням заголовка, тексту та дати публікації. Для зручності користувачів реалізовано пошук новин за датою та посторінковий перегляд результатів, що дає змогу швидко знаходити потрібні матеріали.

Сторінка окремої новини відображає повний текст обраної публікації та дату її розміщення. Дані завантажуються із бази даних відповідно до ідентифікатора новини, переданого в адресному рядку браузера.

Сторінка «Про нас» містить інформацію про веб-платформу, склад редакції та контактні дані. Її призначенням є ознайомлення користувачів із діяльністю інтернет-газети та надання основної контактної інформації.

Під час проектування особливу увагу приділено адаптивності інтерфейсу. Для цього використано можливості CSS-фреймворку Bootstrap, що забезпечує автоматичне підлаштування елементів сторінок під різні розміри екранів. Додатково за допомогою CSS реалізовано єдиний стиль оформлення, а засобами

JavaScript - анімацію взаємодії з елементами списку новин під час наведення курсора.

На рисунку 3.2 наведено макет головної сторінки веб-платформи у десктопній версії. У верхній частині сторінки розташовано шапку сайту з логотипом та навігаційним меню. Нижче знаходиться інформаційний блок із кнопкою переходу до сторінки новин та блок останніх новин у вигляді карток.

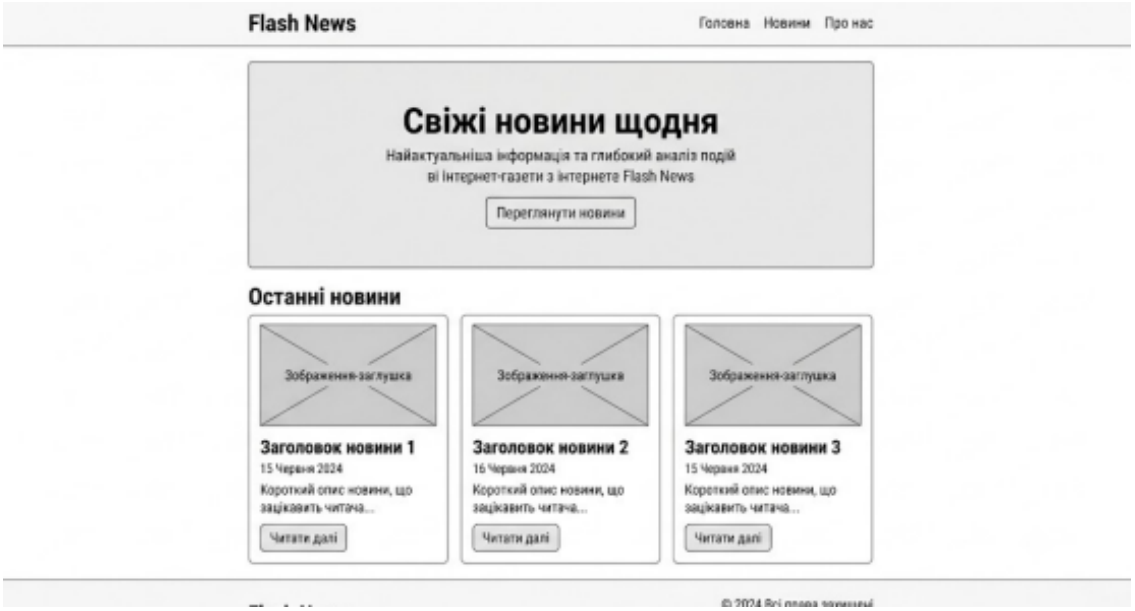


Рисунок 3.2 - Макет головної сторінки (десктопна версія)

На рисунку 3.3 наведено макет головної сторінки для мобільних пристроїв. Елементи інтерфейсу автоматично розташовуються вертикально, а навігаційне меню доступне через кнопку мобільного меню Bootstrap.



Рисунок 3.3 - Макет головної сторінки (мобільна версія)

На рисунку 3.4 показано сторінку перегляду списку новин у десктопній версії. Сторінка містить форму пошуку за датою, список новин та кнопки посторінкового перегляду результатів.



Рисунок 3.4 - Макет сторінки «Новини» (десктопна версія)

На рисунку 3.5 показано адаптивний вигляд сторінки списку новин. Список новин та форма пошуку автоматично підлаштовуються під ширину екрана, забезпечуючи зручність перегляду на смартфонах.



Рисунок 3.5 - Макет сторінки «Новини» (мобільна версія)

На рисунку 3.6 наведено макет сторінки окремої новини. Вона містить заголовок статті, повний текст та дату публікації, отримані з бази даних.

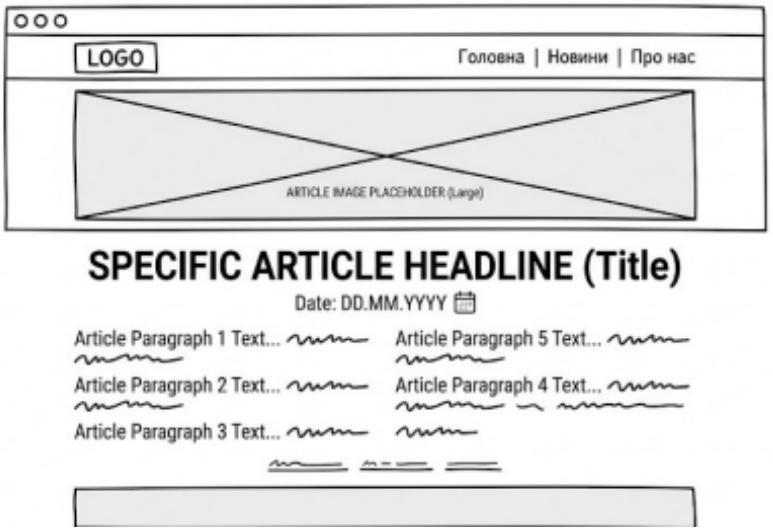


Рисунок 3.6 - Макет сторінки новини (десктопна версія)

На рисунку 3.7 показано мобільну версію сторінки новини. Текст автоматично адаптується до розміру екрана, що забезпечує комфортне читання.

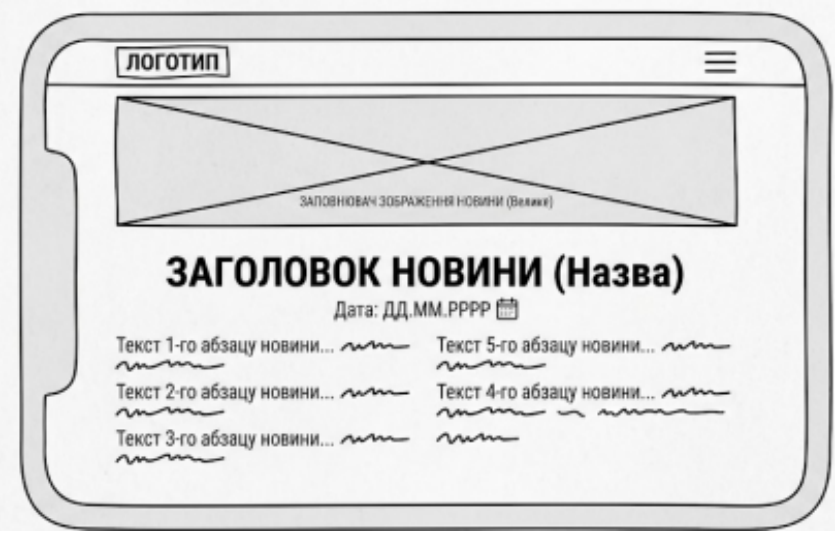


Рисунок 3.7 - Макет сторінки новини (мобільна версія)

На рисунку 3.8 наведено макет сторінки «Про нас». На сторінці розміщено інформацію про веб-платформу, склад редакції та контактні дані.



Рисунок 3.8 - Макет сторінки «Про нас» (десктопна версія)

На рисунку 3.9 показано адаптивне відображення сторінки «Про нас». Усі блоки розташовуються послідовно один під одним, що забезпечує зручний перегляд на мобільних пристроях.



Рисунок 3.9 - Макет сторінки «Про нас» (мобільна версія)

Розроблений клієнтський інтерфейс забезпечує зручну навігацію, логічне розташування інформації, адаптивне відображення сторінок і комфортну взаємодію користувачів із веб-платформою, що створює основу для ефективного використання її функціональних можливостей.

3.3 Проектування бази даних веб-платформи

База даних веб-платформи «Flash News» розроблена для збереження, структурування та отримання новинного контенту. Для реалізації використано реляційну модель даних та СКБД MySQL, що забезпечує ефективну взаємодію з серверною частиною, написаною мовою PHP.

Структура бази даних складається з таблиці news, яка містить основну інформацію про новинні публікації. Така структура є достатньою для реалізації

основного функціоналу веб-платформи та може бути розширена у майбутньому додаванням нових таблиць. Такий підхід дозволяє спростити структуру бази даних, забезпечити швидке отримання інформації та оптимізувати виконання SQL-запитів.

Таблиця news містить такі поля:

– id - ідентифікатор новини;

– title - заголовок новини;

– text - повний текст публікації;

– author - автор новини;

– date - дата публікації.

Під час відкриття сторінки новин сервер виконує SQL-запит до таблиці news, отримує необхідні записи та відображає їх у вигляді списку, відсортованого за датою публікації. Для перегляду повного тексту статті використовується поле id, за допомогою якого виконується пошук відповідного запису в базі даних та формується сторінка обраної новини.

Розроблена структура забезпечує зручне збереження новин, швидке отримання інформації та можливість подальшого розширення системи, зокрема додавання адміністративної панелі для керування контентом.

3.4 Проектування серверної логіки та алгоритмів взаємодії веб-платформи

Серверна логіка веб-платформи «Flash News» відповідає за обробку запитів користувачів, взаємодію з базою даних MySQL та формування динамічного контенту. Робота системи реалізована за принципом клієнт-сервер: користувач надсилає запит через браузер, веб-сервер Apache передає його PHP-скриптам, які виконують необхідні операції з базою даних та формують відповідну HTML-сторінку.

Основним завданням серверної частини є отримання та відображення новин із бази даних, реалізація перегляду окремих публікацій, пошуку новин за датою публікації та посторінкового виведення результатів. Для взаємодії з

базою даних використовується бібліотека PDO у складі PHP та підготовлені SQL-запити, що забезпечує безпечну обробку вхідних даних.

Алгоритм взаємодії компонентів системи:

1 Користувач відкриває головну сторінку, сторінку новин або обирає окрему публікацію.

2 Браузер надсилає HTTP-запит до веб-сервера Apache.

3 Сервер передає запит відповідному PHP-скрипту для обробки.

4 PHP встановлює з'єднання з базою даних MySQL та виконує необхідний SQL-запит.

5 Отримані дані використовуються для формування HTML-сторінки.

6 Сформована сторінка повертається користувачу та відображається у веб-браузері.

На головній сторінці веб-платформи виконується запит до бази даних для отримання трьох останніх новин, які відображаються у блоці «Популярні новини». На сторінці «Новини» реалізовано посторінковий перегляд матеріалів із можливістю фільтрації за датою публікації. Для цього використовуються параметри GET, що дозволяють визначити номер сторінки та вибрану користувачем дату.

Під час відкриття окремої статті система отримує її ідентифікатор (id) із параметрів запиту, виконує пошук відповідного запису в таблиці news та формує сторінку з повним текстом новини. Якщо запис із зазначеним ідентифікатором відсутній, система виводить повідомлення про помилку.

3.5 Розробка випробувальних сценаріїв для перевірки функціональності веб-платформи

Після завершення етапу проектування було розроблено випробувальні сценарії для перевірки функціонування веб-платформи інтернет-газети «Flash News». Їхньою метою є підтвердження відповідності системи функціональним

вимогам, перевірка взаємодії клієнтської та серверної частин, а також оцінка стабільності роботи веб-платформи.

Під час підготовки сценаріїв враховано ключові можливості системи: відображення головної сторінки, перегляд списку новин, відкриття окремих публікацій, отримання даних із бази даних, роботу навігації та адаптивність інтерфейсу.

Перелік випробувальних сценаріїв перевірки функціональності веб-платформи:

1. Завантаження головної сторінки. Послідовність дій користувача: відкрити головну адресу веб-платформи. Очікуваний результат: відображається головна сторінка з навігаційним меню та списком актуальних новин.

2. Перехід до розділу «Новини». Послідовність дій користувача: натиснути пункт меню «Новини». Очікуваний результат: відображається сторінка зі списком новин.

3. Перегляд окремої новини. Послідовність дій користувача: обрати будь-яку новину зі списку. Очікуваний результат: відкривається сторінка з повним текстом новини, автором та датою публікації.

4. Отримання даних із бази даних. Послідовність дій користувача: перейти до сторінки новин. Очікуваний результат: інформація успішно зчитується з бази даних і відображається без помилок.

5. Перевірка навігації. Послідовність дій користувача: виконати переходи між усіма сторінками. Очікуваний результат: усі посилання працюють коректно, сторінки відкриваються без помилок.

6. Перевірка адаптивності. Послідовність дій користувача: відкрити веб-платформу на різних розмірах екрана. Очікуваний результат: інтерфейс автоматично адаптується до екранів персонального комп'ютера, планшета та смартфона.

Результати виконання наведених сценаріїв нададуть змогу оцінити працездатність веб-платформи, перевірити коректність реалізації її основних

функцій та підтвердити відповідність визначеним функціональним і технічним вимогам.

Висновки до розділу 3

У третьому розділі виконано проектування веб-платформи інтернет-газети «Flash News». Розроблено структуру веб-платформи, що забезпечує логічну організацію сторінок та зручну навігацію користувачів. Спроектовано клієнтський інтерфейс із використанням адаптивного дизайну, який забезпечує коректне відображення веб-платформи на персональних комп'ютерах і мобільних пристроях.

Також визначено структуру бази даних MySQL для зберігання новинного контенту та спроектовано серверну логіку, яка забезпечує отримання даних із бази, перегляд окремих новин, пошук за датою публікації та посторінкове відображення результатів. Крім цього, розроблено випробувальні сценарії для перевірки працездатності основних функцій веб-платформи.

Результати, отримані в цьому розділі, створюють основу для реалізації веб-платформи, її тестування та подальшого впровадження.

4 РОЗРОБКА ВЕБ-ПЛАТФОРМИ ІНТЕРНЕТ-ГАЗЕТИ

У цьому розділі наведено процес програмної реалізації веб-платформи інтернет-газети «Flash News». Виконано налаштування апаратного та програмного середовища розробки, реалізовано клієнтський інтерфейс, створено базу даних та інтегровано її з серверною частиною веб-платформи. Також розроблено серверну логіку для обробки запитів користувачів і проведено тестування функціональності створеної системи. Результати виконаної роботи підтверджують працездатність веб-платформи та її відповідність визначеним функціональним і технічним вимогам.

4.1 Опис апаратного та програмного середовища розробки

Для розробки веб-платформи «Flash News» використовувався персональний комп'ютер під керуванням операційної системи Linux Mint та редактор програмного коду Lite XL. Локальний запуск і тестування веб-платформи здійснювалися за допомогою пакета XAMPP, до складу якого входять веб-сервер Apache, СКБД MySQL і веб-інтерфейс phpMyAdmin.

Під час розробки клієнтської частини було використано мови HTML, CSS, JavaScript та CSS-фреймворк Bootstrap, а серверну частину реалізовано мовою програмування PHP. Для взаємодії з базою даних застосовано розширення PDO, яке забезпечує безпечне виконання SQL-запитів.

Обране апаратне та програмне середовище забезпечило стабільну розробку, тестування та подальший розвиток веб-платформи «Flash News».

4.2 Розробка клієнтського інтерфейсу веб-платформи

Після завершення етапу проектування було виконано програмну реалізацію клієнтського інтерфейсу веб-платформи інтернет-газети «Flash

News». Основною метою цього етапу стало створення адаптивного та зручного інтерфейсу, який забезпечує швидкий доступ користувачів до новинного контенту та комфортну навігацію між сторінками веб-платформи.

Верстка сторінок виконувалася за допомогою мови HTML із використанням CSS та фреймворку Bootstrap. Це дозволило створити єдиний стиль оформлення всіх сторінок та забезпечити їх коректне відображення на персональних комп'ютерах, планшетах і смартфонах.

Спочатку було реалізовано спільний шаблон веб-платформи, який використовується на всіх сторінках. До його складу входять шапка сайту, навігаційне меню, основний інформаційний блок та нижній колонтитул. Результат реалізації спільного шаблону веб-платформи наведено на (рис. 4.1).

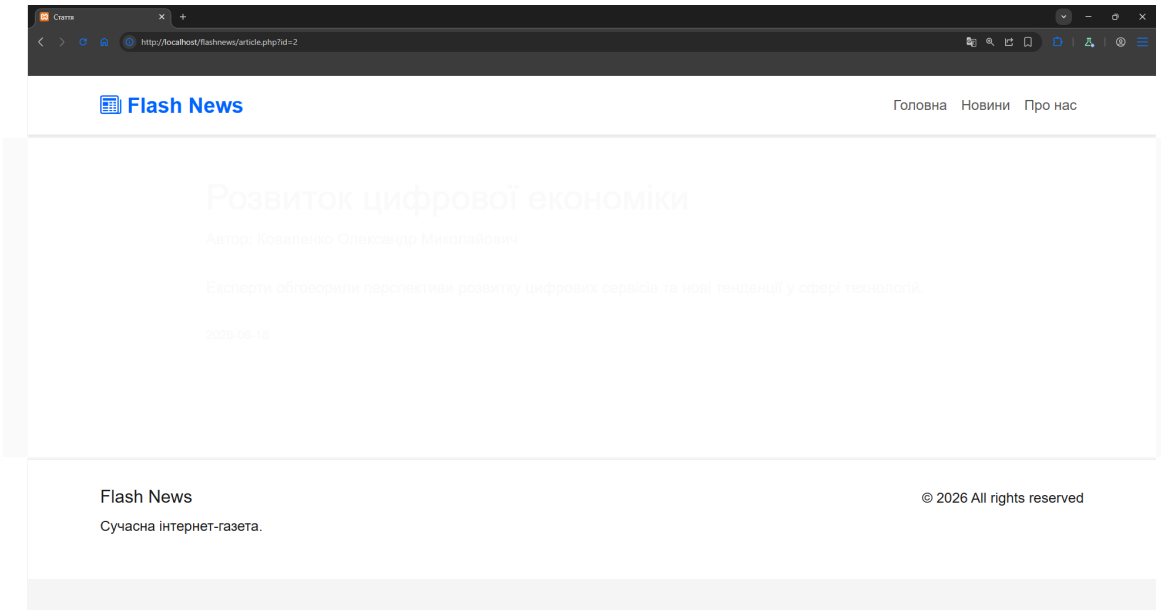


Рисунок 4.1 - Реалізація спільного шаблону веб-платформи

У верхній частині сторінки розташовано логотип веб-платформи та навігаційне меню, яке забезпечує швидкий перехід до головної сторінки, сторінки новин та сторінки «Про нас». Для мобільних пристроїв використовується стандартне меню Bootstrap, яке автоматично перетворюється на кнопку навігації.

Після реалізації спільного шаблону було створено головну сторінку веб-платформи. Її зовнішній вигляд та основні елементи інтерфейсу наведено на (рис. 4.2).

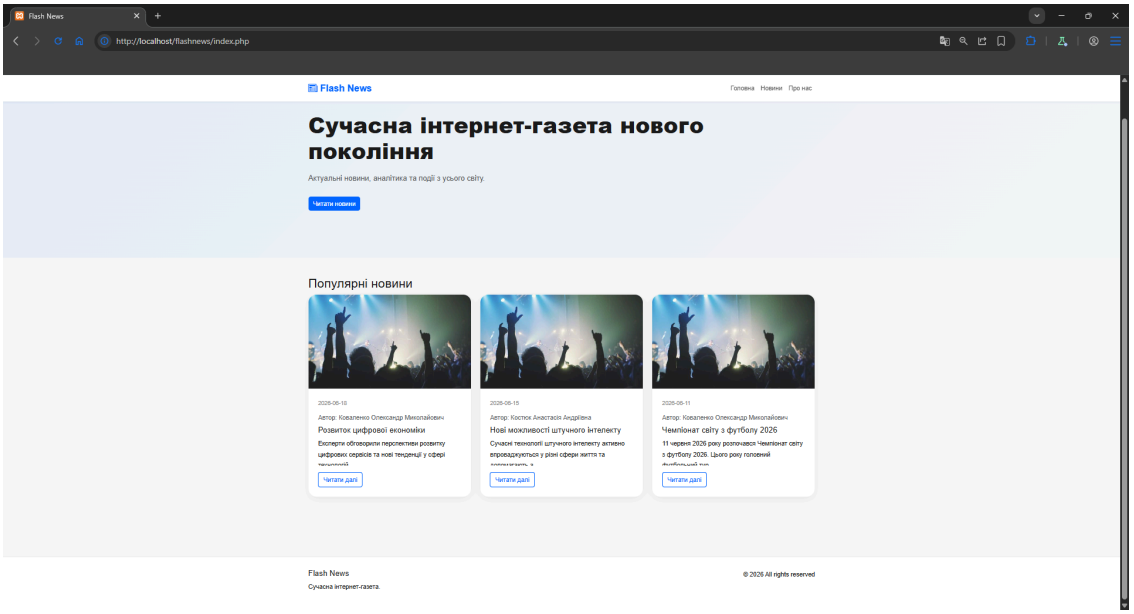


Рисунок 4.2 - Реалізація головної сторінки веб-платформи

Наступним етапом стала реалізація сторінки новин. На сторінці виводиться список усіх новин, які зберігаються у базі даних. Для кожної публікації відображаються заголовок, короткий текст, автор та дата публікації. Також реалізовано форму пошуку новин за датою та посторінковий перегляд результатів. Результат реалізації сторінки новин із відображенням списку публікацій наведено на (рис. 4.3).

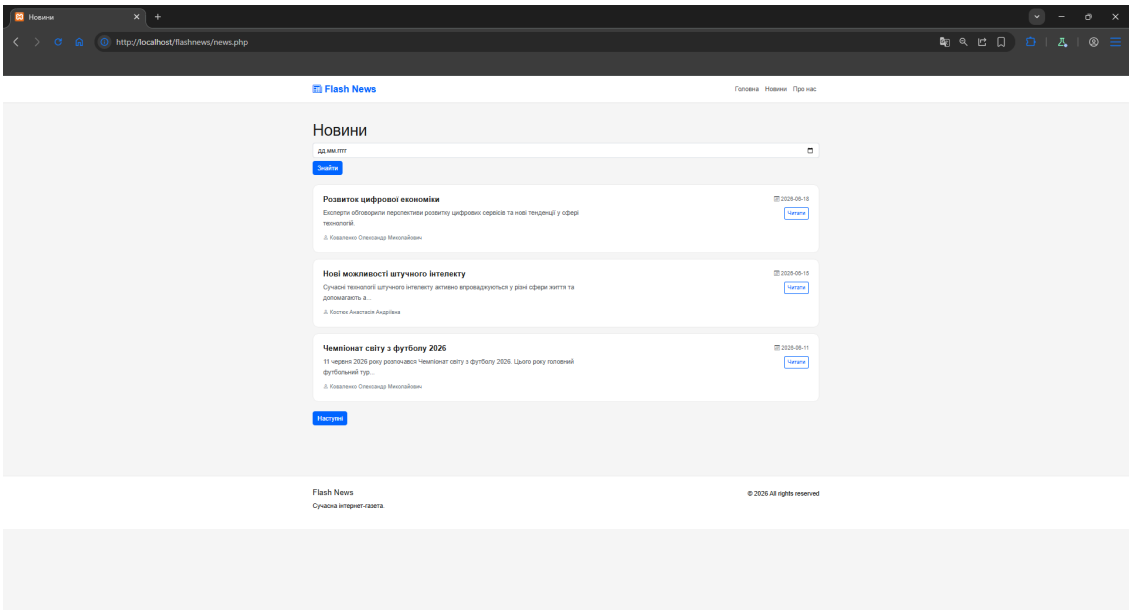


Рисунок 4.3 - Реалізація сторінки новин

Для перегляду повного тексту статті було створено окрему сторінку новини. Інформація автоматично отримується із бази даних відповідно до ідентифікатора, який передається через параметри URL. Реалізацію сторінки окремої новини з повним текстом публікації представлено на (рис. 4.4).

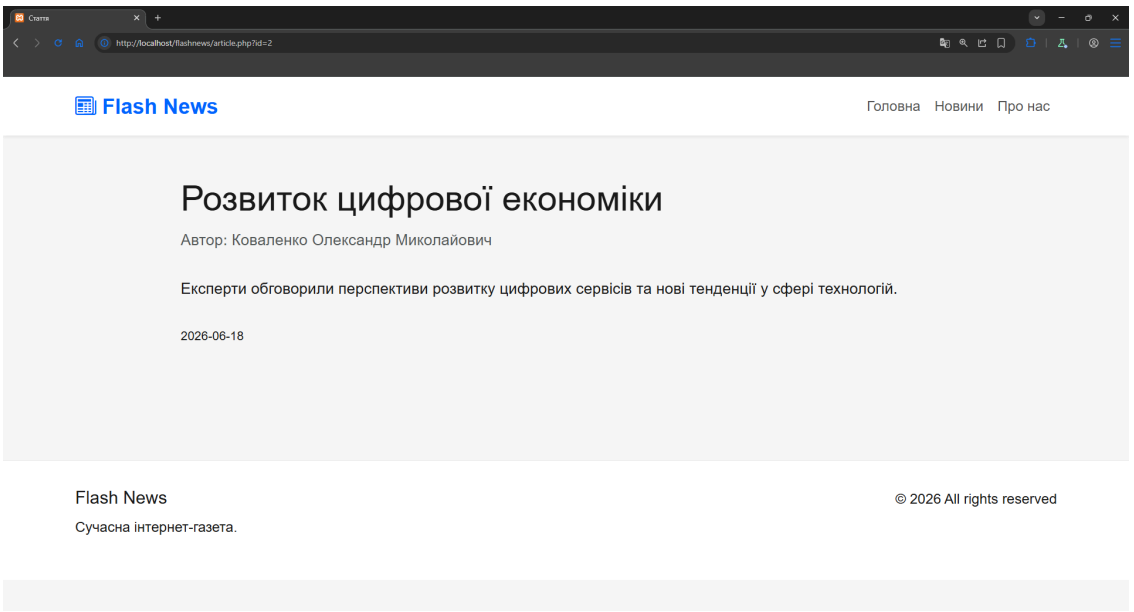


Рисунок 4.4 - Реалізація сторінки окремої новини

Також було реалізовано сторінку «Про нас», яка містить інформацію про веб-платформу, її призначення, склад редакції та контактні дані.

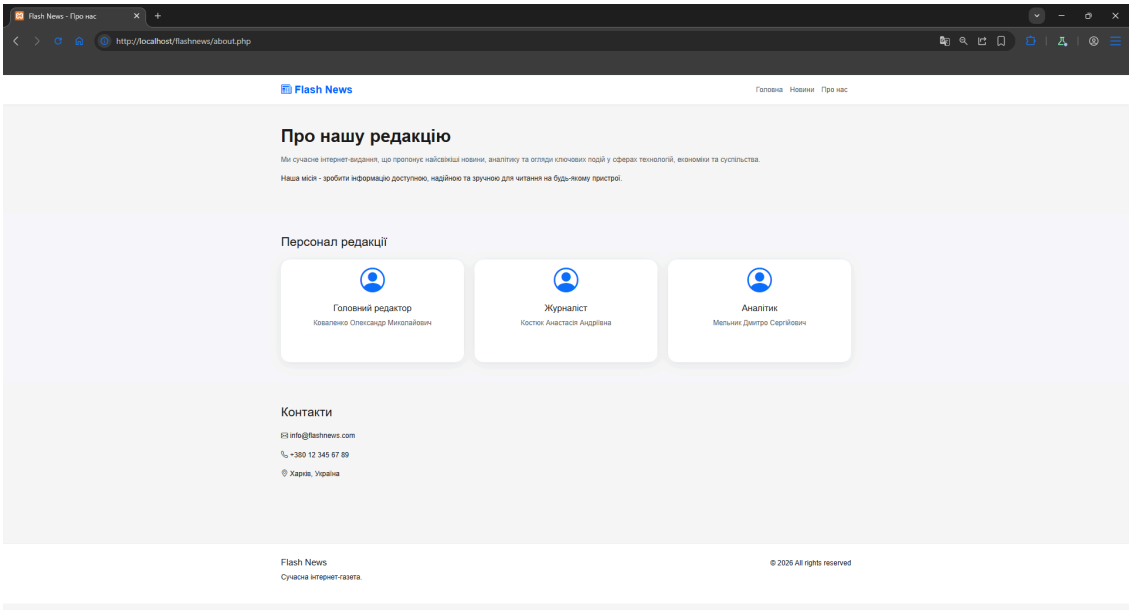


Рисунок 4.5 - Реалізація сторінки «Про нас»

Для оформлення інтерфейсу використано власні стилі CSS. За їх допомогою реалізовано кольорову гаму веб-платформи, оформлення карток новин, кнопок, меню навігації та інших елементів інтерфейсу. Для підвищення зручності використання за допомогою JavaScript реалізовано анімацію взаємодії користувача з елементами списку новин під час наведення курсора.

Завдяки використанню Bootstrap усі сторінки автоматично адаптуються до різних розмірів екранів. Це забезпечує комфортне використання веб-платформи як на персональних комп'ютерах, так і на мобільних пристроях без необхідності створення окремої мобільної версії.

У результаті виконання цього етапу було реалізовано повноцінний клієнтський інтерфейс веб-платформи інтернет-газети «Flash News», який забезпечує зручну навігацію, сучасний зовнішній вигляд та адаптивне відображення інформації.

4.3 Реалізація бази даних веб-платформи

Для зберігання інформації про новини у веб-платформі «Flash News» використовується система керування базами даних MySQL. Створення бази даних було виконано за допомогою графічного веб-інтерфейсу phpMyAdmin, який входить до складу програмного пакета XAMPP та забезпечує зручне адміністрування баз даних без необхідності використання командного рядка.

Спочатку було створено нову базу даних. Для цього після відкриття phpMyAdmin у лівій частині вікна було обрано пункт New, після чого створено базу даних із назвою flash_news.

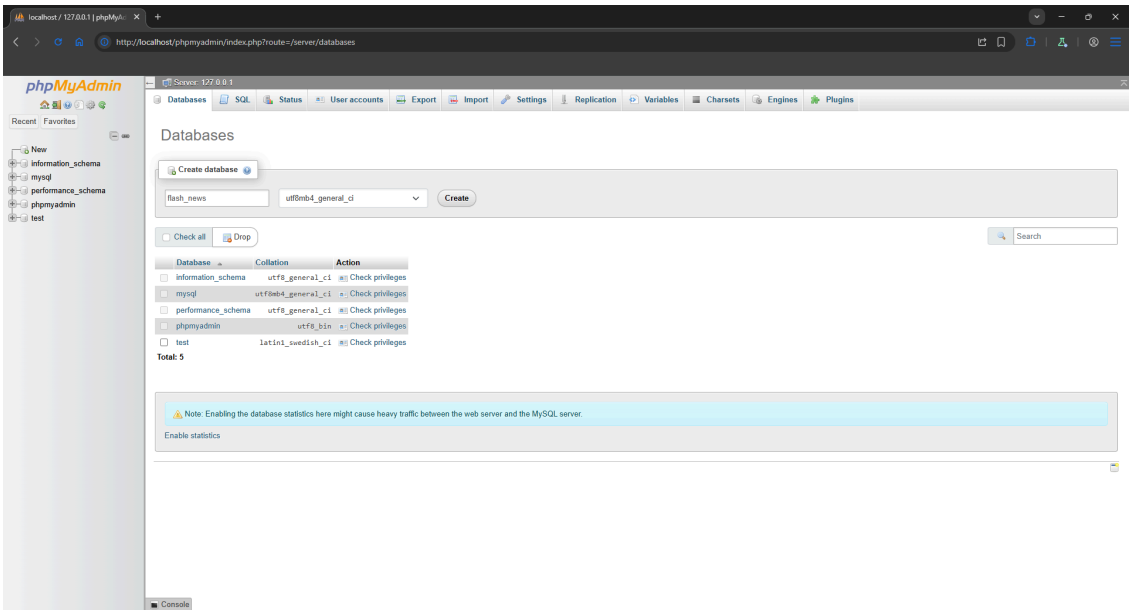


Рисунок 4.6 - Створення бази даних flash_news у phpMyAdmin

Після створення бази даних було виконано створення таблиці news, у якій зберігається інформація про новинні публікації.

Послідовність створення таблиці у середовищі phpMyAdmin наведено на (рис. 4.7).

1. У списку баз даних було обрано базу даних flash_news.

2. У полі Створити таблицю введено назву news.

3. Визначено кількість стовпців таблиці.

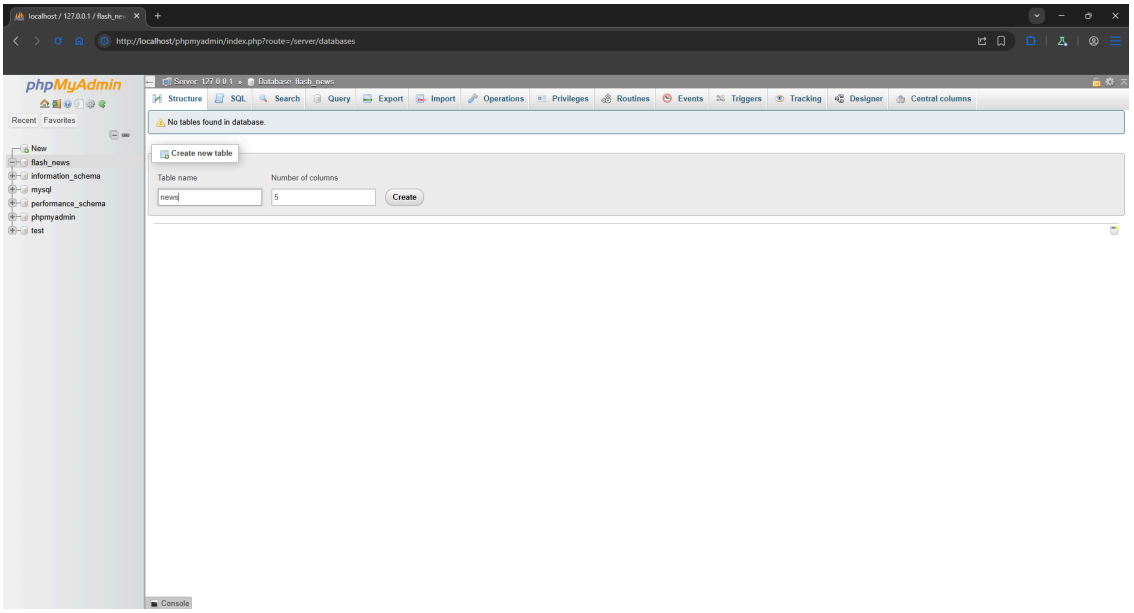


Рисунок 4.7 - Створення таблиці news

Після цього було сформовано структуру таблиці, у якій визначено назви полів, типи даних та основний ключ таблиці.

Таблиця містить такі поля:

– id - унікальний ідентифікатор новини;

– title - заголовок новини;

– text - текст новинної публікації;

– author - автор матеріалу;

– date - дата публікації новини.

Поле id визначено як первинний ключ таблиці з параметром AUTO_INCREMENT, що забезпечує автоматичне присвоєння ідентифікатора кожному новому запису. Структуру створеної таблиці news у середовищі phpMyAdmin наведено на (рис. 4.8).

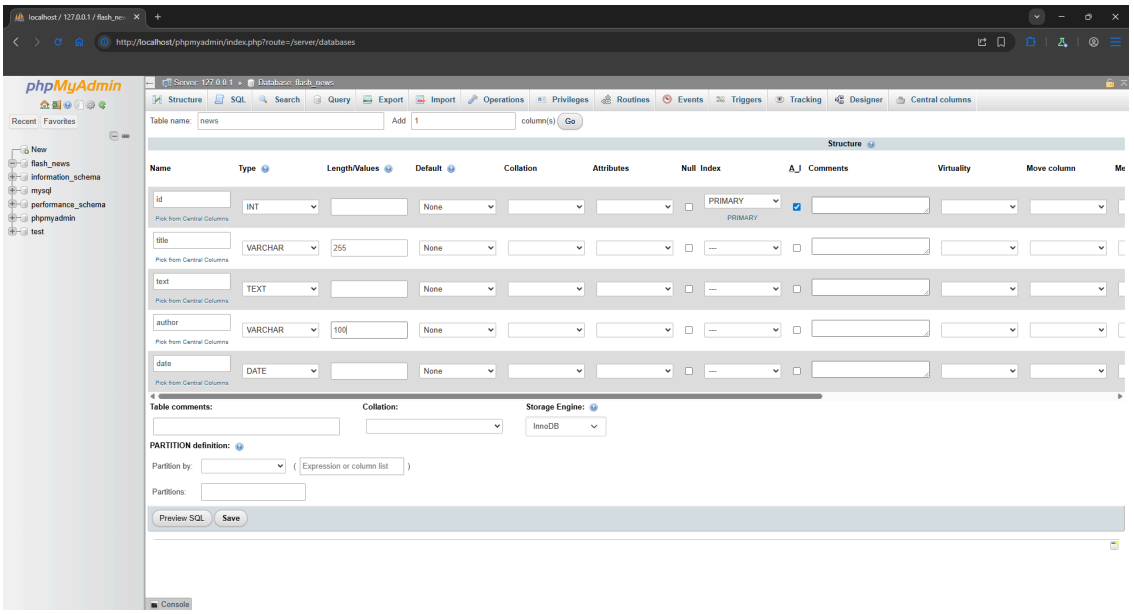


Рисунок 4.8 - Структура таблиці news

Після завершення створення таблиці було додано тестові записи для перевірки роботи веб-платформи.

Послідовність додавання тестових записів:

1. Відкрити таблицю news.

2. Перейти до вкладки Insert.

3. Заповнити поля title, text, author та date.

4. Поле id залишити порожнім, оскільки його значення формується автоматично.

Процес додавання тестового запису у таблицю news наведено на (рис. 4.9).

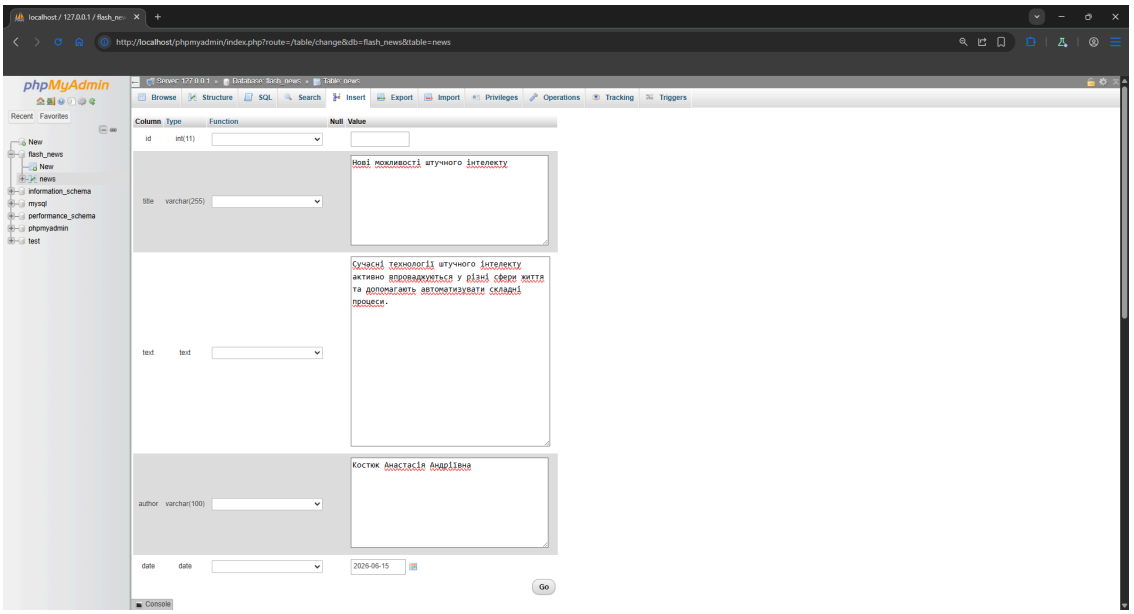


Рисунок 4.9 - Додавання тестового запису

Після додавання тестових записів таблиця news містить заповнені поля з інформацією про новини. Результат заповнення таблиці наведено на (рис. 4.10).

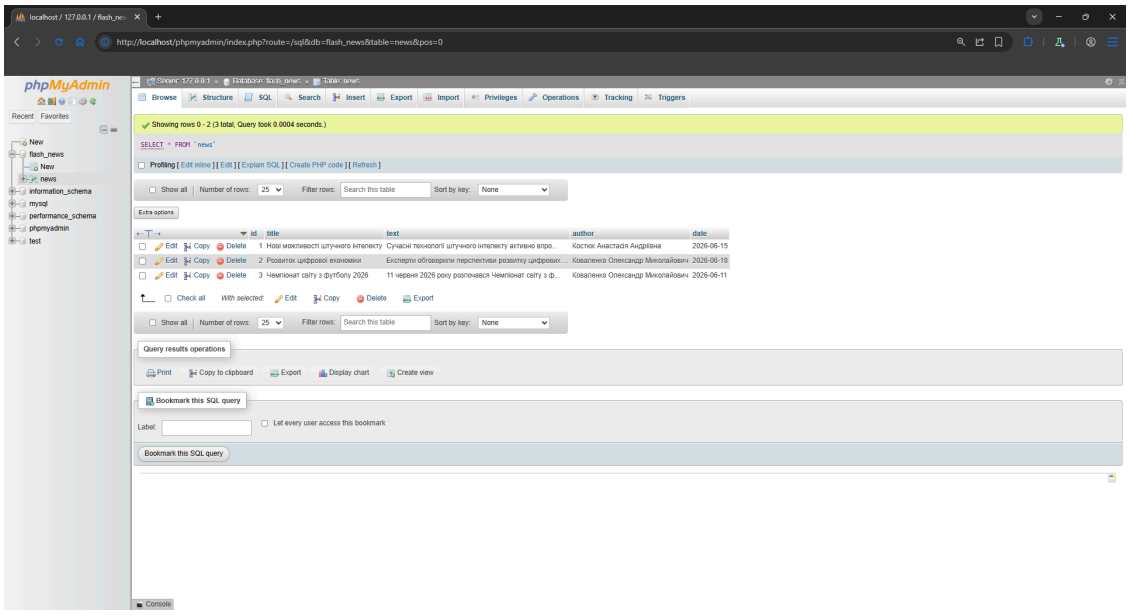


Рисунок 4.10 - Таблиця news після додавання тестових записів

Після завершення створення бази даних її було інтегровано з серверною частиною веб-платформи. Для встановлення з'єднання використовується розширення PDO, яке забезпечує безпечну взаємодію PHP із СКБД MySQL та підтримує виконання підготовлених SQL-запитів. Завдяки цьому веб-платформа отримує можливість автоматично зчитувати інформацію про новини, виконувати пошук за датою публікації та сформувати динамічний вміст сторінок.

У результаті виконання цього етапу створено повністю працездатну реляційну базу даних, яка забезпечує зберігання новинного контенту та його подальше використання серверною частиною веб-платформи.

4.4 Розробка серверної логіки веб-платформи

Після реалізації клієнтського інтерфейсу та створення бази даних було розроблено серверну частину веб-платформи інтернет-газети «Flash News». Вона забезпечує обробку запитів користувачів, взаємодію з базою даних MySQL та автоматичне формування сторінок.

Серверну логіку реалізовано мовою програмування PHP під керуванням веб-сервера Apache. Для роботи з базою даних використано розширення PDO, яке забезпечує безпечне підключення та виконання підготовлених SQL-запитів.

На першому етапі створено окремий файл підключення до бази даних, що містить параметри з'єднання та створення об'єкта PDO. Це дозволило використовувати єдине підключення в усіх PHP-скриптах. Реалізацію підключення до бази даних наведено на (рис. 4.11).

```php
<?php
$host = "localhost";
$db = "flash_news";
$user = "root";
$password = "";
try {

    $pdo = new PDO(
        "mysql:host=$host;dbname=$db;charset=utf8mb4",
        $user,
        $password
    );

    $pdo->setAttribute(
        PDO::ATTR_ERRMODE,
        PDO::ERRMODE_EXCEPTION
    );
} catch (PDOException $e) {

    die("Error: " . $e->getMessage());
}
?>
```

Рисунок 4.11 - Реалізація підключення до бази даних

Після встановлення з'єднання реалізовано отримання останніх новин для головної сторінки. PHP-скрипт виконує SQL-запит до таблиці news, після чого записи передаються до HTML-шаблону та відображаються у вигляді карток. Результат реалізації отримання останніх новин наведено на (рис. 4.12).

```php
<?php
$title = "Новини";
include "includes/header.php";
include "db.php";
$limit = 10;
$page = $_GET['page'] ?? 1;
$page = max(1, (int) $page);
$offset = ($page - 1) * $limit;
$date = $_GET['date'] ?? null;
if ($date) {
    $stmt = $pdo->prepare(
        "SELECT * FROM news
        WHERE date = :date
        ORDER BY date DESC
        LIMIT :limit OFFSET :offset"
    );
    $stmt->bindValue(':date', $date);
} else {
    $stmt = $pdo->prepare(
        "SELECT * FROM news
        ORDER BY date DESC
        LIMIT :limit OFFSET :offset"
    );

}
$stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
$stmt->bindValue(':offset', $offset, PDO::PARAM_INT);
if ($date) {
    $stmt->bindValue(':date', $date);
}
$stmt->execute();
$news = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>
```

Рисунок 4.12 - Реалізація отримання останніх новин

Для сторінки «Новини» реалізовано виведення повного списку публікацій із сортуванням за датою публікації, що забезпечує відображення найактуальніших матеріалів першими.

Також реалізовано пошук новин за датою. Після вибору потрібної дати PHP-скрипт формує підготовлений SQL-запит, що забезпечує безпечну передачу параметрів і запобігає SQL-ін'єкціям. Реалізацію пошуку новин за датою публікації представлено на (рис. 4.13).

```php
$date = $_GET['date'] ?? null;
if ($date) {
    $stmt = $pdo->prepare(
        "SELECT * FROM news
        WHERE date = :date
        ORDER BY date DESC
        LIMIT :limit OFFSET :offset"
    );
    $stmt->bindValue(':date', $date);
} else {
    $stmt = $pdo->prepare(
        "SELECT * FROM news
        ORDER BY date DESC
        LIMIT :limit OFFSET :offset"
    );

}
$stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
$stmt->bindValue(':offset', $offset, PDO::PARAM_INT);
if ($date) {
    $stmt->bindValue(':date', $date);
}
$stmt->execute();
```

Рисунок 4.13 - Реалізація пошуку новин за датою

Для зручності перегляду великої кількості матеріалів реалізовано посторінковий перегляд. Сервер отримує номер сторінки з параметрів GET-запиту, визначає необхідну кількість записів та формує SQL-запит із використанням операторів LIMIT і OFFSET. Реалізацію механізму посторінкового перегляду новин наведено на (рис. 4.14).

```
        <?php endforeach; ?>
        <div class="mt-4">
            <a href="?page=<?= $page + 1 ?><?= $date ? '&date=' . $date : '' ?>" class="btn btn-primary">
                Наступні
            </a>
            <?php if ($page > 1): ?>
                <a href="?page=<?= $page - 1 ?>&date=<?= $date ?>" class="btn btn-secondary">
                    Назад
                </a>
            <?php endif; ?>
```

Рисунок 4.14 - Реалізація посторінкового перегляду

Під час відкриття окремої новини система отримує її ідентифікатор із адресного рядка браузера, знаходить відповідний запис у таблиці news та формує сторінку з повним текстом, автором і датою публікації. Якщо запис відсутній, виводиться повідомлення про помилку. Результат реалізації перегляду окремої новини з отриманням даних із бази даних наведено на (рис. 4.15).

```
        <?php foreach ($news as $item): ?>
            <a href="article.php?id=<?= $item['id'] ?>" class="news-link">
                <div class="news-list-item">
                    <div class="d-flex justify-content-between align-items-start">
                        <div class="me-3">
                            <h5 class="fw-bold mb-2">
                                <?= htmlspecialchars($item['title']) ?>
                            </h5>
                            <p class="text-muted mb-2">
                                <?= htmlspecialchars(
                                    mb_strimwidth($item['text'], 0, 100, "...")
                                ) ?>
                            </p>
                            <small class="text-secondary">
                                <i class="bi bi-person"></i>
                                <?= htmlspecialchars($item['author']) ?>
                            </small>
                        </div>
```

Рисунок 4.15 - Реалізація перегляду окремої новини

Взаємодія компонентів веб-платформи здійснюється за клієнт-серверною архітектурою: браузер надсилає HTTP-запит, веб-сервер Apache передає його PHP-скрипту, який звертається до бази даних, отримує необхідні записи та формує HTML-сторінку для відображення користувачу.

У результаті реалізовано серверну логіку, яка забезпечує коректну взаємодію компонентів системи, автоматичне формування сторінок і безпечну роботу з базою даних.

4.5 Тестування веб-платформи

Після завершення програмної реалізації проведено тестування основних функцій веб-платформи. Метою тестування була перевірка роботи клієнтської та серверної частин, взаємодії з базою даних і стабільності системи.

Під час тестування перевірено сценарії, розроблені на етапі проектування: відображення інформації, роботу навігації, пошук новин та адаптивність інтерфейсу.

Спочатку перевірено завантаження головної сторінки.

Послідовність перевірки:

1. Відкрити головну сторінку.
2. Перевірити навігаційне меню.
3. Перевірити відображення останніх новин.

Очікуваний результат - сторінка завантажується без помилок, усі елементи інтерфейсу відображаються коректно. Результат перевірки завантаження головної сторінки веб-платформи наведено на (рис. 4.16).

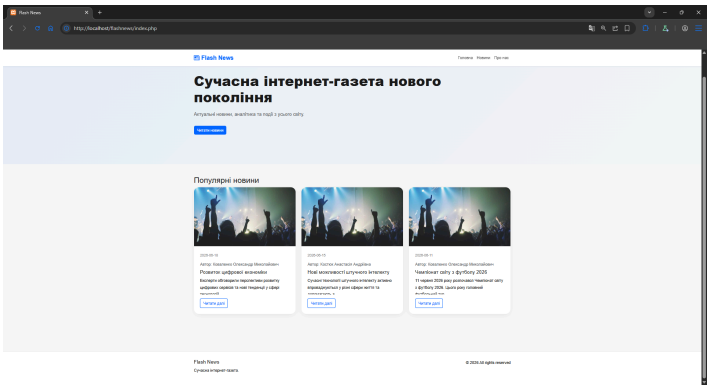


Рисунок 4.16 - Результат перевірки головної сторінки

Наступним етапом перевірено сторінку новин.

Послідовність перевірки:

1. Перейти до розділу «Новини».

2. Перевірити отримання списку новин із бази даних.

3. Перевірити відображення заголовків, текстів, авторів і дат публікації.

Очікуваний результат - записи успішно отримуються з бази даних та відображаються без помилок. Результат перевірки роботи сторінки новин наведено на (рис. 4.17).

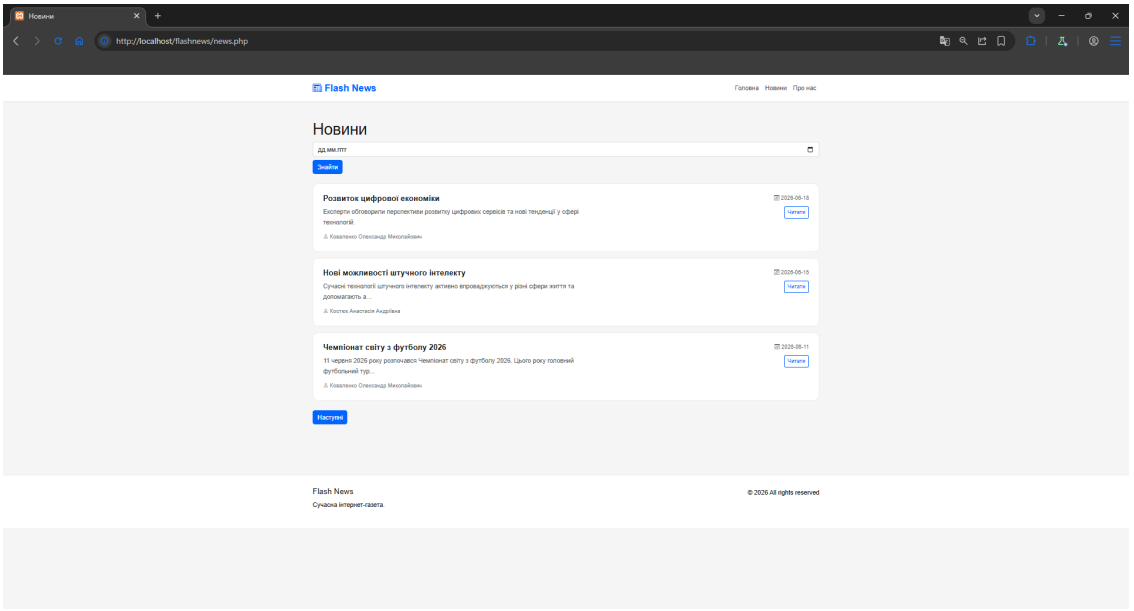


Рисунок 4.17 - Перевірка сторінки новин

Після цього протестовано пошук новин за датою публікації.

Послідовність перевірки:

1. Вибрати дату у формі пошуку.

2. Натиснути кнопку пошуку.

3. Перевірити результати фільтрації.

Очікуваний результат - система відображає новини відповідно до обраної дати. Результат перевірки пошуку новин за датою публікації наведено на (рис. 4.18).

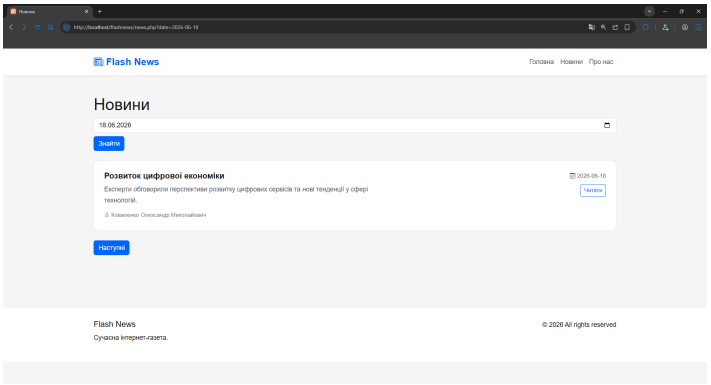


Рисунок 4.18 - Перевірка пошуку новин

Далі перевірено відкриття окремої новини.

Послідовність перевірки:

1. Обрати новину зі списку.

2. Перейти на сторінку повної статті.

Очікуваний результат - відображається повний текст новини, автор та дата публікації. Результат перевірки відкриття окремої новини наведено на (рис. 4.19).

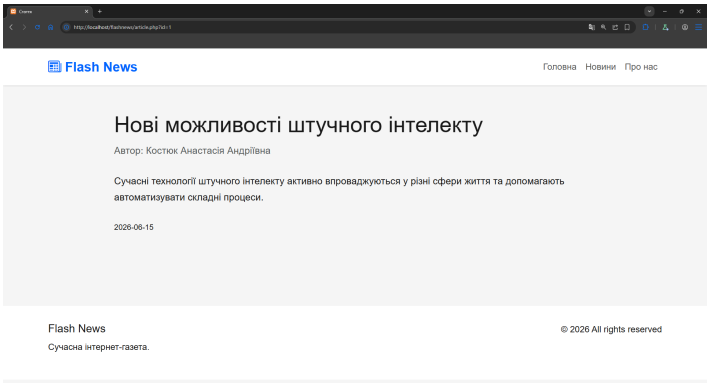


Рисунок 4.19 - Перевірка сторінки окремої новини

Також перевірено посторінковий перегляд новин.

Послідовність перевірки:

1. Перейти на наступну сторінку списку новин.

2. Перевірити відображення записів.

Очікуваний результат - сторінки перемикаються коректно, відображаються відповідні записи.

Результат перевірки - сторінки перемикаються коректно та всі відповідні записи відображаються.

Окремо проведено тестування адаптивності інтерфейсу.

Веб-платформу перевірено на персональному комп'ютері та смартфоні. Встановлено, що елементи інтерфейсу адаптуються до розміру екрана, а навігаційне меню коректно працює в мобільному режимі. Результат перевірки адаптивності веб-платформи на різних пристроях наведено на (рис. 4.20).

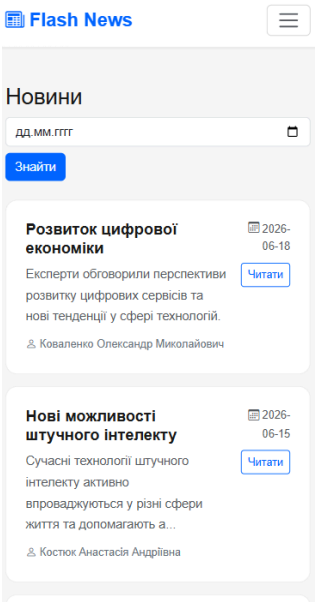


Рисунок 4.20 - Перевірка адаптивності веб-платформи

За результатами тестування встановлено, що основні функції веб-платформи працюють коректно. Помилок під час завантаження сторінок,

роботи з базою даних та відображення інформації не виявлено. Результати підтвердили працездатність системи та її відповідність визначеним функціональним вимогам.

Висновки до розділу 4

У четвертому розділі виконано програмну реалізацію веб-платформи інтернет-газети «Flash News». Налаштовано середовище розробки, реалізовано адаптивний клієнтський інтерфейс, створено та інтегровано базу даних MySQL з серверною частиною веб-платформи.

Серверну логіку розроблено мовою програмування PHP з використанням розширення PDO, що забезпечило безпечну взаємодію з базою даних, автоматичне формування сторінок, перегляд і пошук новин, а також посторінкове відображення результатів.

Проведене тестування підтвердило коректну роботу клієнтської та серверної частин, бази даних і адаптивного інтерфейсу. Отримані результати свідчать про працездатність веб-платформи, її відповідність визначеним функціональним і технічним вимогам та готовність до подальшого використання й розвитку.

ВИСНОВКИ

Під час проектування та розробки веб-платформи інтернет-газети був використаний набір сучасних веб-технологій, забезпечили стабільну роботу системи, адаптивність інтерфейсу та зручність взаємодії користувача. Основними технологіями стали веб-сервер Apache, мова розмітки HTML, каскадні таблиці стилів CSS, мова програмування JavaScript, мова програмування PHP та система керування базою даних MySQL, адміністрування якої здійснювалося за допомогою графічного інструменту phpMyAdmin.

Для створення структури веб-сторінок було використано мову розмітки HTML, яка дозволила змоделювати ключові елементи інтерфейсу: навігаційне меню, блоки новин. За допомогою каскадних таблиць стилів було реалізовано візуальне оформлення проекту. CSS-фреймворк Bootstrap забезпечив коректне відображення. Його використання також дозволило уніфікувати інтерфейс. Впровадження інтерактивності було досягнуто за допомогою мови програмування JavaScript, яка забезпечила динамічну взаємодію з елементами інтерфейсу. Обробка серверної логіки була здійснена за допомогою мови програмування PHP, яка відповідала за обробку запитів користувачів та автоматичне створення сторінок газети на основі інформації з бази даних. Для збереження та структурування всієї інформації сайту було використано СКБД MySQL, а взаємодія з базою даних здійснювалася за допомогою мови SQL. Для спрощення адміністрування та швидкого керування таблицями було інтегровано графічний веб-інтерфейс phpMyAdmin.

Таким чином, поєднання обраних технологій дозволило створити сучасну веб-платформу інтернет-газети, що забезпечує швидку роботу, адаптивний інтерфейс, зручність використання та відповідає поставленим технічним вимогам.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Digital 2026: Global Overview Report [Електронний ресурс] // DataReportal. - 2026. - Режим доступу до ресурсу: https://datareportal.com/reports/digital-2026-global-overview-report

2. Laurie B., Laurie P. Apache: The Definitive Guide. - 3rd ed. - Sebastopol : O'Reilly Media, 2002. - 588 с. https://www.apachefriends.org/

3. XAMPP Documentation [Електронний ресурс] // Apache Friends. - 2026. - Режим доступу до ресурсу:

4. HTML Living Standard [Електронний ресурс] // WHATWG. - 2026. - Режим доступу до ресурсу: https://html.spec.whatwg.org/multipage/

5. Cascading Style Sheets [Електронний ресурс] // W3C. - 2026. - Режим доступу до ресурсу: https://www.w3.org/Style/CSS/

6. Usage statistics and market shares of CSS frameworks [Електронний ресурс] // W3Techs. - 2026. - Режим доступу до ресурсу: https://w3techs.com/technologies/overview/css_framework

7. Bootstrap Documentation [Електронний ресурс] // Bootstrap. - 2026. - Режим доступу до ресурсу: https://getbootstrap.com/

8. Foundation Documentation [Електронний ресурс] // Foundation. - 2026. - Режим доступу до ресурсу: https://get.foundation/

9. Tailwind CSS Documentation [Електронний ресурс] // Tailwind CSS. - 2026. - Режим доступу до ресурсу: https://tailwindcss.com/

10. JavaScript First Steps [Електронний ресурс] // MDN Web Docs. - 2026. - Режим доступу до ресурсу: https://devdoc.net/web/developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps.html

11. Usage Statistics of client-side Programming Languages for Websites [Електронний ресурс] // W3Techs. - 2026. - Режим доступу до ресурсу: https://w3techs.com/technologies/overview/client_side_language

12. Usage Statistics of Programming Languages for Websites [Електронний ресурс] // W3Techs. - 2026. - Режим доступу до ресурсу: https://w3techs.com/technologies/overview/programming_language

13. PHP Documentation [Електронний ресурс] // PHP. - 2026. - Режим доступу до ресурсу: https://www.php.net/docs.php

14. Ruby Documentation [Електронний ресурс] // Ruby. - 2026. - Режим доступу до ресурсу: https://www.ruby-lang.org/en/documentation/

15. Java Documentation [Електронний ресурс] // Oracle. - 2026. - Режим доступу до ресурсу: https://docs.oracle.com/en/java/

16. MySQL Documentation [Електронний ресурс] // MySQL. - 2026. - Режим доступу до ресурсу: https://dev.mysql.com/doc/

17. PostgreSQL Documentation [Електронний ресурс] // PostgreSQL. - 2026. - Режим доступу до ресурсу: https://www.postgresql.org/

18. MongoDB Documentation [Електронний ресурс] // MongoDB. - 2026. - Режим доступу до ресурсу: https://www.mongodb.com/

19. ISO/IEC 9075-1:2023 Standard [Електронний ресурс] // ISO. - 2026. - Режим доступу до ресурсу: https://www.iso.org/standard/76583.html

20. phpMyAdmin Documentation [Електронний ресурс] // phpMyAdmin. - 2026. - Режим доступу до ресурсу: https://docs.phpmyadmin.net/

21. HeidiSQL Documentation [Електронний ресурс] // HeidiSQL. - 2026. - Режим доступу до ресурсу: https://www.heidisql.com/documentation.php

22. MySQL Workbench Manual [Електронний ресурс] // Oracle. - 2026. - Режим доступу до ресурсу: https://dev.mysql.com/doc/workbench/en/

23. Database Administrators Stack Exchange [Електронний ресурс]. // Database Administrator - Режим доступу: https://dba.stackexchange.com/