

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Національний університет «Острозька академія»**  
**Економічний факультет**  
**Кафедра економіко-математичного моделювання та інформаційних технологій**

**КВАЛІФІКАЦІЙНА РОБОТА**  
на здобуття освітнього ступеня магістра

на тему: **«УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ САЙТУ  
НА ПЛАТФОРМІ WORDPRESS»**

**Виконав:** студент 2 курсу, групи МУП-21  
другого (магістерського) рівня вищої освіти  
спеціальності 122 Комп'ютерні науки  
освітньо-професійної програми «Управління проєктами»  
*Стретович Дмитро Андрійович*

**Керівник:** *Місай Володимир Віталійович*

**Рецензент:** *кандидат технічних наук, доцент  
кафедр прикладної математики та кібербезпеки  
Донецького національного університету імені Василя Стуса  
Загоруйко Любов Василівна*

**РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ**

Завідувач кафедри економіко-математичного моделювання та інформаційних  
технологій \_\_\_\_\_ (проф., д.е.н. Кривицька О.Р.)  
Протокол № \_\_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 2024 р.

Острог, 2024

Міністерство освіти і науки України  
Національний університет «Острозька академія»

Факультет: економічний

Кафедра: економіко-математичного моделювання та інформаційних технологій

Спеціальність: 122 Комп'ютерні науки

Освітньо-професійна програма: Управління проєктами

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
Ольга КРИВИЦЬКА

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
на кваліфікаційну роботу студента**

Стретовича Дмитра Андрійовича

1. Тема роботи Управління проєктом розробки сайту на платформі WordPress.  
керівник роботи Місай Володимир Віталійович

Затверджено наказом ректора НаУОА від “03” листопада 2023 року №98

2. Термін здачі студентом закінченої роботи: 05 грудня 2024 р.

3. Вихідні дані до роботи: технічні (ядро WordPress, теми, плагіни, мови програмування PHP та JavaScript, інструменти веб розробки), змістовні, бізнес-вимоги, аналіз цільової аудиторії, контентні, дизайнерські та правові вимоги.

4. Перелік завдань, які належить виконати: описати предметне середовище розробки сайту на платформі WordPress CMS; розглянути успішні сценарії розроблених сайтів на платформі Wordpress CMS; розкрити предметну область розробки; сформувати план розробки сайту; висвітлити методологію розробки веб-сайту; перелічити засоби розробки; сформувати вимоги до технічного та програмного забезпечення; описати фактичну реалізацію проєкту.

5. Перелік графічного матеріалу: рисунки, таблиці.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Місай В.В.	01.12.23	01.12.23
2	Місай В.В.	01.12.23	01.12.23
3	Місай В.В.	01.12.23	01.12.23

7. Дата видачі завдання: 01 грудня 2023 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів	Примітка
1	Затвердження теми роботи	до 20.10.2023 р.	
2	Постановка технічного завдання	до 12.09.2024 р.	
3	Аналіз технічного завдання	до 30.09.2024 р.	
4	Розробка концепції та структури сайту	до 15.10.2024 р.	
5	Розробка дизайну сайту	до 25.10.2024 р.	
6	Технічна підготовка, встановлення та налаштування системи WordPress	до 01.11.2024 р.	
7	Встановлення теми, плагінів та їх налаштування	до 05.11.2024 р.	
8	Наповнення сайту	до 15.11.2024 р.	
9	Тестування сайту	до 20.11.2024 р.	
10	Запуск сайту	до 27.11.2024 р.	
11	Попередній захист кваліфікаційної роботи	до 28.11.2024 р.	
12	Здача кваліфікаційної роботи на кафедрі	до 01.12.2024 р.	

Студент: \_\_\_\_\_ Дмитро СТРЕТОВИЧ

Керівник кваліфікаційної роботи: \_\_\_\_\_ Володимир МІСАЙ

**АНОТАЦІЯ**  
**кваліфікаційної роботи**  
**на здобуття освітнього ступеня магістра**

***Тема:** Управління проектом розробки сайту на платформі WordPress*

***Автор:** Стретович Дмитро Андрійович*

***Науковий керівник:** Місай Володимир Віталійович*

*Захищена «.....» ..... 20\_\_ року.*

***Пояснювальна записка до кваліфікаційної роботи:** 68 с., 14 рис., 1 табл., 9 додатків, 49 джерел.*

***Ключові слова:** розробка, сайт, WordPress, управління проектом.*

***Короткий зміст праці:***

*Робота розглядає процес управління розробкою сайту на платформі WordPress. Описано предметне середовище та проаналізовано успішні сценарії аналогів. Здійснено детальний аналіз предметної області, визначено основні кроки до реалізації мінімально життєздатного продукту та висвітлено методології розробки сайту. Перелічено основні засоби розробки необхідні для реалізації проекту, сформовано вимоги до технічного та програмного забезпечення проекту та поетапно описано процес реалізації проекту.*

---

**ABSTRACT**  
**of the qualification paper**  
**for obtaining a master's degree**

***Paper title:*** Managing a website development project on the WordPress platform

***Author:*** Dmytro Stretovych

***Scientific supervisor:*** Volodymyr Misai

***Defended on*** «.....» ..... **2024**

***Explanatory note to the qualification work:*** 68 pages, 14 pictures., 1 table, 9 appendix, 49 references.

***Key words:*** development, website, WordPress, managing a project.

***Summary of the paper:***

*The paper describes the process of developing a website on the WordPress platform. The subject environment is described and successful analogue scenarios are analysed. A detailed analysis of the subject area is carried out, the main steps to the implementation of a minimum viable product are determined, and the methodology of website development is highlighted. The main development tools required for the project implementation are listed, the hardware and software requirements for the project are formed, and the project implementation process is described step by step.*

---

## ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	9
1.1. Опис предметного середовища (функціональної моделі, процесу діяльності)..	9
1.2. Огляд аналогів успішних сценаріїв.....	15
1.3. Постановка задачі.....	19
Висновки до розділу 1 .....	22
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МЕТОДОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ .....	23
2.1. Аналіз предметної області (вхідні та вихідні дані) .....	23
2.2. Створення плану розробки системи (план розробки/впровадження обов'язкових складових системи для етапу мінімального життєздатного продукту) .....	26
2.3. Методології розробки .....	28
Висновки до розділу 2 .....	32
РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	33
3.1. Засоби розробки .....	33
3.2. Вимоги до технічного та програмного забезпечення.....	34
3.3. Опис реалізації.....	35
3.4. Керівництво користувача .....	47
Висновки до розділу 3 .....	50
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	53
ДОДАТКИ.....	57

## ВСТУП

Сучасний цифровий ландшафт характеризується надзвичайною динамічністю та високим рівнем непередбачуваності, що зумовлено стрімким розвитком інформаційних технологій та постійними змінами у потребах користувачів. Веб-ресурси сьогодні виконують не лише функцію візитної картки компанії чи організації в онлайн-просторі, а й виступають ключовими елементами комунікаційної інфраструктури, забезпечуючи доступ до інформації, товарів та послуг для широкої аудиторії. Усе більше користувачів очікують від веб-сайтів не лише естетичної привабливості, а й високої продуктивності, інтерактивності та зручності в навігації. З огляду на ці вимоги, процес створення та управління веб-проектами стає важливою складовою цифрової трансформації, що вимагає від розробників використання передових технологій та методик для досягнення конкурентних переваг.

Серед різноманітних платформ для створення веб-сайтів особливе місце займає система управління контентом (CMS) WordPress, яка отримала визнання завдяки своїй простоті, гнучкості та широким функціональним можливостям. На сьогодні ця платформа використовується для розробки понад 40% веб-сайтів у світі [1], що підкреслює її популярність серед розробників різного рівня. WordPress надає користувачам можливість створювати як прості блоги, так і складні корпоративні портали з розвиненим функціоналом, інтеграцією сторонніх сервісів та адаптивним дизайном.

Однак процес створення сайту на базі WordPress не обмежується лише вибором шаблону та встановленням необхідних плагінів. Успішна реалізація веб-проекту вимагає від команди розробників не лише технічних знань, але й комплексного підходу до управління проектом. Зокрема, важливими аспектами є планування кожного етапу розробки, аналіз цільової аудиторії, визначення ключових функцій ресурсу та забезпечення відповідності кінцевого продукту очікуванням користувачів. Крім того, важливою складовою є тестування та оптимізація сайту з метою забезпечення високої швидкості завантаження, стабільної роботи та захисту від потенційних кіберзагроз.

Застосування сучасних методів управління проектами, таких як Scrum або Kanban, дозволяє ефективніше координувати роботу команди, адаптуватися до змінних умов та оперативно реагувати на нові виклики. Таким чином, створення веб-сайту на платформі WordPress — це багатогранний процес, який поєднує технічні, управлінські та креативні аспекти для досягнення високоякісного кінцевого продукту, здатного відповідати сучасним вимогам цифрового середовища.

**Мета дослідження:** проаналізувати методології та практики управління проектами, а також їх застосування в контексті розробки веб-сайтів на WordPress.

**Завдання дослідження:**

- описати предметне середовище розробки сайту на платформі WordPress CMS;
- розглянути успішні сценарії розроблених сайтів на платформі Wordpress CMS;
- розкрити предметну область розробки;
- сформулювати план розробки сайту;
- висвітлити методологію розробки веб-сайту;
- перелічити засоби розробки;
- сформулювати вимоги до технічного та програмного забезпечення;
- описати фактичну реалізацію проекту.

**Об'єктом дослідження** є проєкт розробки сайту на платформі WordPress

**Предметом дослідження** є особливості управлінням проєкту розробки сайту на платформі WordPress

**Теоретико-методологічна основа дослідження.** Дослідження було проведено з використанням широкого спектру методів, що забезпечило його всебічність. Аналіз дозволив виявити структуру об'єкта, синтез – об'єднати окремі елементи в єдине ціле, описовий метод – детально описати властивості об'єкта, а порівняльний – оцінити його відносно інших об'єктів.



**Практична апробація.** Результати досліджень окремих аспектів цієї теми презентувалися на XXII Всеукраїнській науково-практичній онлайн-конференції молодих учених та студентів «Проблеми та перспективи розвитку національної економіки в умовах глобалізації» (м. Острого, 20 березня 2024 р.) та під час XXIX наукової викладацько-студентська конференції «Дні науки» (м. Острого, 13-17 травня 2024 р.).

## РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1. Опис предметного середовища (функціональної моделі, процесу діяльності)

У сучасних умовах цифрової економіки веброзробка відіграє ключову роль у створенні інформаційних ресурсів, що забезпечують комунікацію між користувачами та організаціями. Однією з найпопулярніших платформ для створення та управління вебсайтами є система управління контентом (CMS) WordPress (Додаток А), яка забезпечує гнучкість, зручність використання та широкі функціональні можливості. Розробка сайтів на WordPress включає цілу низку процесів, що стосуються роботи з серверним середовищем, базами даних, мовами програмування, бібліотеками, фреймворками та інструментами для забезпечення оптимізації, дизайну та безпеки.

Система управління контентом WordPress базується на архітектурі «клієнт-сервер», яка включає три основні компоненти:

- Клієнтська частина (Front-end) — інтерфейс, що взаємодіє з користувачами;
- Серверна частина (Back-end) — програмний код, що обробляє запити та відповіді клієнта;
- База даних — система зберігання контенту та налаштувань сайту.

Клієнтська частина сайту на WordPress представлена інтерфейсом, який забезпечує взаємодію між користувачем та вебресурсом. Ця частина відповідає за відображення контенту, його структуризацію, стильове оформлення та динамічні елементи, що сприяють інтерактивності. Основними технологіями, які формують клієнтську частину, є HTML, CSS та JavaScript.

HTML (HyperText Markup Language) є базовою мовою розмітки, яка визначає структуру вебсторінки [2]. Завдяки HTML задається розташування текстів, зображень, відео та інших мультимедійних елементів на сторінці. У контексті WordPress HTML автоматично генерується системою на основі шаблонів теми, які використовує сайт. Структура документа формується через використання тегів, що дозволяє створити логічну ієрархію елементів, таких як заголовки, абзаци, списки, таблиці та форми. З огляду на важливість доступності вебконтенту, сучасні

веброзробники повинні дотримуватись стандартів семантичного HTML, що покращує сприйняття сторінки пошуковими системами та допоміжними технологіями для людей з обмеженими можливостями.

CSS (Cascading Style Sheets) відповідає за стилізацію сторінок, надаючи їм естетичний вигляд та забезпечуючи єдину візуальну концепцію [3]. Каскадні таблиці стилів дозволяють визначати параметри зовнішнього вигляду HTML-елементів, такі як кольори, шрифти, відступи, розташування та анімації. У WordPress CSS-файли також можуть бути частиною теми, що дозволяє швидко змінювати вигляд сайту шляхом встановлення нової теми або редагування стилів у вже існуючій. Використання каскадних таблиць стилів значно полегшує підтримку сайту, оскільки всі стилі можуть бути централізовано змінені в одному файлі, що автоматично оновлює всі сторінки ресурсу.

JavaScript забезпечує інтерактивність клієнтської частини сайту, надаючи можливість створювати динамічні елементи, що реагують на дії користувача [4]. Це може бути перевірка даних у формах, динамічне завантаження контенту без перезавантаження сторінки, створення спливаючих вікон або інтерактивних слайдерів. У WordPress JavaScript інтегрується через спеціальні скрипти, які можуть бути частиною теми або додаватися через плагіни. Окрім базового JavaScript, часто використовуються бібліотеки, такі як jQuery, що дозволяють значно спростити написання коду та розширити функціональність сайту.

Завдяки цим технологіям клієнтська частина сайту на WordPress є гнучкою, масштабованою та легко адаптується до різних вимог замовника. HTML забезпечує структурованість контенту, CSS — його візуальну привабливість, а JavaScript додає функціональність та динамічність, що є критично важливими для сучасних вебсайтів. Разом ці компоненти забезпечують не лише естетичний, але й функціональний досвід користувача, який є ключовим фактором успішності вебресурсу.

Back-end WordPress є критично важливою частиною архітектури платформи, яка відповідає за обробку серверних запитів, взаємодію з базою даних, управління контентом та генерацію динамічного HTML-коду для клієнтської частини. Ця

сторона системи виконує ключову роль у забезпеченні функціональності сайту, його продуктивності, масштабованості та безпеки.

Основу серверної частини WordPress становить мова програмування PHP (Hypertext Preprocessor) [5]. PHP забезпечує реалізацію логіки взаємодії між користувачем та системою. Коли користувач надсилає запит до сервера, PHP обробляє цей запит, отримує відповідні дані з бази даних, динамічно формує HTML-код і відправляє його на клієнтську сторону для відображення в браузері. Завдяки гнучкості PHP розробники можуть створювати як прості, так і складні вебсайти з індивідуальними функціями, налаштовуючи код під специфічні потреби проекту.

PHP забезпечує роботу ядра WordPress, яке відповідає за базову функціональність платформи. Зокрема, мова використовується для обробки запитів до бази даних MySQL [6] або MariaDB, де зберігається весь контент сайту: тексти публікацій, коментарі, користувацькі облікові записи, налаштування сайту тощо. Запити до бази даних можуть бути оптимізовані для підвищення швидкості роботи сайту та зменшення навантаження на сервер. Для забезпечення більш складних функцій, таких як створення кастомних пост-типів, таксономій чи власних полів, використовуються розширення на основі PHP.

Окрім PHP, серверна частина WordPress може використовувати різноманітні фреймворки та бібліотеки, які розширюють можливості платформи. Одним із найбільш важливих компонентів є WP REST API — інтерфейс прикладного програмування, який дозволяє взаємодіяти з WordPress через зовнішні додатки або інші сайти [7]. WP REST API забезпечує доступ до даних сайту у форматі JSON, що спрощує інтеграцію з мобільними додатками, сервісами сторонніх розробників та іншими системами. Завдяки REST API можна створювати headless-сайти, де WordPress використовується виключно як бекенд, а фронтенд реалізується на інших технологіях, таких як React або Vue.js.

Файлова система є важливим компонентом бекенду WordPress, оскільки вона містить усі основні файли платформи, плагіни, теми та медіафайли. Структура файлової системи організована таким чином, щоб забезпечити зручний доступ до різних компонентів сайту. Файли ядра WordPress розташовані в основному каталозі

платформи, тоді як додаткові плагіни та теми зберігаються у відповідних папках. Медіафайли, завантажені користувачами, зберігаються в окремій директорії, що дозволяє легко керувати ними через інтерфейс адміністративної панелі.

Файлова система також відіграє ключову роль у безпеці сайту. Правильна конфігурація прав доступу до файлів і директорій допомагає запобігти несанкціонованому доступу до критичних компонентів платформи. Крім того, резервне копіювання файлової системи є важливою частиною стратегії забезпечення безперервної роботи сайту та збереження його даних у випадку технічних збоїв або кібератак.

Таким чином, бекенд WordPress є комплексною системою, де PHP відповідає за виконання основної логіки платформи, фреймворки та бібліотеки розширюють її функціональні можливості, а файлова система забезпечує зберігання та організацію всіх необхідних даних. Взаємодія цих компонентів дозволяє створювати стабільні, функціональні та масштабовані вебсайти, що відповідають сучасним вимогам користувачів і бізнесу.

WordPress використовує базу даних MySQL (або MariaDB), яка забезпечує зберігання та управління контентом сайту:

- посторінковий контент (статті, сторінки, коментарі);
- налаштування сайту (параметри теми, конфігурація плагінів);
- користувацькі дані (облікові записи користувачів, ролі та права доступу).

База даних WordPress організована у вигляді таблиць, де кожна таблиця відповідає за певний тип інформації (наприклад, `wp_posts`, `wp_users`, `wp_options`).

Розробка сайтів на WordPress може здійснюватися як у локальному середовищі, так і на віддалених серверах.

Процес розробки може здійснюватися як у локальному середовищі, так і на віддалених серверах, кожне з яких має свої переваги та недоліки, що впливають на вибір підходу залежно від потреб проєкту.

Локальне середовище є важливим інструментом для початкового етапу розробки, коли сайт створюється, налаштовується та тестується без підключення до мережі Інтернет. Його основною перевагою є безпека та можливість

експериментувати з функціональністю без ризику негативного впливу на кінцевих користувачів. Типове локальне середовище включає такі компоненти, як вебсервер (Apache або Nginx), який обробляє HTTP-запити та відповіді, систему управління базами даних (зазвичай MySQL або MariaDB), а також інтерпретатор PHP, що дозволяє виконувати скрипти WordPress. Таке середовище можна легко налаштувати на персональному комп'ютері за допомогою пакетів програмного забезпечення, таких як XAMPP, MAMP, WampServer або Open Server Panel. Локальна розробка дає змогу розробникам працювати офлайн, прискорює процес тестування та знижує залежність від інтернет-з'єднання.

З іншого боку, віддалене середовище передбачає розміщення вебсайту на сервері, який доступний через Інтернет. Віддалені сервери можуть бути фізичними машинами, розташованими у дата-центрах, або віртуальними серверами у хмарних інфраструктурах, таких як Amazon Web Services, Google Cloud чи Microsoft Azure. Використання віддаленого середовища є обов'язковим на етапі розгортання сайту для широкого доступу. Основна перевага такого середовища полягає в забезпеченні доступності сайту для користувачів з будь-якої точки світу, високій масштабованості та можливості автоматизованого резервного копіювання даних. Крім того, віддалені сервери зазвичай пропонують вищий рівень безпеки, зокрема, завдяки використанню протоколів SSL/TLS, міжмережевих екранів та інших засобів захисту даних.

Таким чином, обидва типи середовищ відіграють важливу роль у процесі розробки на WordPress. Локальне середовище забезпечує комфортні умови для створення та тестування, тоді як віддалене гарантує надійне розгортання та стабільну роботу вебсайту в Інтернеті. Ефективне використання цих середовищ дозволяє оптимізувати процес розробки, підвищити продуктивність роботи та забезпечити високу якість кінцевого продукту.

У процесі написання коду для веброботи використовується широкий спектр інструментів, які забезпечують зручність, ефективність і продуктивність роботи розробників. Кожен з таких інструментів має свої особливості, функціональність і сфери застосування, що дозволяє розробникам обирати найбільш оптимальне рішення для конкретних завдань.

Одним із найпопулярніших редакторів коду є Visual Studio Code, розроблений компанією Microsoft [8]. Цей редактор набув широкого поширення завдяки своїй легкості, безкоштовності та підтримці численних розширень, які можна налаштувати відповідно до потреб розробника. Visual Studio Code підтримує синтаксис багатьох мов програмування, таких як PHP, CSS, HTML, JavaScript, Python, а також дозволяє інтегрувати різні інструменти для налагодження, управління версіями та контролю коду. Завдяки інтеграції з системами контролю версій, такими як Git, Visual Studio Code полегшує командну роботу над проєктами, дозволяючи відстежувати зміни та координувати дії команди розробників.

Ще одним потужним інструментом є PhpStorm, інтегроване середовище розробки, що спеціалізується на мові PHP [9]. PhpStorm надає розширені можливості для роботи з PHP-кодом, включно з автоматичним завершенням коду, статичним аналізом, пошуком помилок та інтеграцією з популярними фреймворками, такими як Laravel, Symfony та Yii. Крім того, PhpStorm підтримує роботу з базами даних, дозволяючи розробникам безпосередньо взаємодіяти з ними з середовища IDE, що значно прискорює процес розробки серверної частини вебдодатків. Завдяки цим функціям PhpStorm часто обирають професійні розробники, які спеціалізуються на розробці складних вебпроєктів.

Для тих, хто шукає легший та менш ресурсозатратний інструмент, підходить Sublime Text [10]. Цей текстовий редактор відомий своєю швидкістю роботи, навіть при роботі з великими файлами, та широкою підтримкою плагінів, які можна налаштувати для роботи з різними мовами програмування. Sublime Text забезпечує функції підсвічування синтаксису, автозавершення коду, а також дозволяє створювати макроси для автоматизації повторюваних завдань. Завдяки простому інтерфейсу та високій швидкості роботи Sublime Text часто використовують для швидкого редагування коду або написання скриптів.

Ще одним популярним редактором є Notepad++, який відрізняється мінімалістичним дизайном і простотою у використанні. Notepad++ підтримує широкий спектр мов програмування та має відкритий вихідний код [11]. Він відзначається низьким споживанням ресурсів системи, що робить його популярним

серед розробників, які працюють на менш потужних пристроях або виконують прості завдання з редагування тексту та коду. Серед основних функцій цього редактора — підсвічування синтаксису, підтримка вкладок для одночасної роботи з декількома файлами та можливість автоматизованого форматування коду.

Таким чином, вибір інструменту для написання коду залежить від потреб конкретного проєкту, рівня складності завдань та особистих вподобань розробника. Інтегровані середовища розробки, такі як PhpStorm, забезпечують глибоку функціональність та інтеграцію з різними сервісами, що підходить для професійних проєктів, тоді як редактори на кшталт Visual Studio Code чи Sublime Text пропонують баланс між легкістю використання та функціональністю, що робить їх універсальними рішеннями для різних типів веброботи.

Розробка сайту на WordPress завершується етапом тестування та оптимізації, що включає:

- Перевірку сумісності з різними браузерами та пристроями;
- Оптимізацію швидкості завантаження сторінок (зменшення розміру зображень, кешування);
- Перевірку безпеки сайту (використання SSL-сертифікатів, регулярне оновлення плагінів).

Предметне середовище розробки сайтів на WordPress є комплексним і багатofункціональним, включаючи широкий спектр інструментів та технологій. Завдяки гнучкості, масштабованості та простоті використання WordPress залишається одним із провідних рішень для створення сучасних вебсайтів різного типу та призначення.

## **1.2. Огляд аналогів успішних сценаріїв**

WordPress CMS (Content Management System) є однією з найпопулярніших платформ для створення вебсайтів у світі [12]. Її успіх пояснюється високою гнучкістю, багатofункціональністю та доступністю для користувачів з різним рівнем технічних знань. На сьогодні WordPress використовується для розробки найрізноманітніших типів сайтів — від особистих блогів і вебсайтів-візиток до



великих корпоративних порталів і масштабних інтернет-магазинів. Завдяки великій кількості доступних плагінів та тем оформлення, ця система управління контентом є універсальним інструментом для задоволення різних потреб як малих підприємств, так і великих міжнародних корпорацій.

Одним із найпоширеніших сценаріїв використання WordPress є створення вебсайтів-візиток. Такі сайти дозволяють представити базову інформацію про компанію чи приватну особу, включаючи контактні дані, перелік послуг чи продуктів, портфоліо та інші важливі деталі. Завдяки низьким витратам на розробку та підтримку, WordPress є привабливим вибором для малого бізнесу.

Наприклад, локальні кафе, перукарні, фотографи чи юридичні консультанти можуть швидко створити ефективний онлайн-простір для взаємодії з клієнтами. Сайт-візитка може включати сторінку «Про нас», розділ для відгуків клієнтів, галерею з фотографіями робіт та інтеграцію з соціальними мережами.

Проте WordPress CMS використовується не лише малими бізнесами. Вебсайти-візитки великих світових брендів також часто створюються на цій платформі (Додаток Б). Наприклад:

- Katy Perry ([katyperry.com](http://katyperry.com)) — вебсайт відомої співачки має понад 400 тисяч органічного трафіку та понад 600 тисяч загального трафіку.
- The Walt Disney Company ([thewaltdisneycompany.com](http://thewaltdisneycompany.com)) — сайт одного з найбільших медіакорпорацій світу залучає 194 тисячі відвідувачів органічно та понад 1,9 мільйона загального трафіку.
- The Rolling Stones ([rollingstones.com](http://rollingstones.com)) — офіційний сайт легендарного рок-гурту має близько 120 тисяч органічних відвідувачів і 350 тисяч загального трафіку.
- Sony Music ([sonymusic.com](http://sonymusic.com)) — ресурс великої музичної компанії приваблює 19 тисяч органічних відвідувачів і 261 тисячу загального трафіку [13].

Ці приклади демонструють, що платформа є надійним вибором для представлення брендів світового рівня, завдяки її стабільності, можливостям SEO-оптимізації та адаптивності до різних пристроїв.

Іншим важливим напрямом використання WordPress є створення інтернет-магазинів. Завдяки плагіну WooCommerce, який є одним із найпопулярніших рішень

для електронної комерції, WordPress дозволяє перетворити будь-який сайт на повноцінний онлайн-магазин.

WooCommerce є універсальним інструментом для створення та управління онлайн-магазинами, який пропонує широкий спектр функціональних можливостей. Платформа забезпечує глибоку інтеграцію з різноманітними платіжними системами, включаючи PayPal, Stripe, Apple Pay та Google Pay, що значно спрощує процес оплати для клієнтів. WooCommerce також надає інструменти для ефективного управління замовленнями та контролю рівня запасів товарів. Гнучка система налаштування дозволяє організувати доставку товарів у будь-яку точку світу. Крім того, платформа підтримує створення різноманітних маркетингових кампаній, таких як акції, знижки та купони, що сприяє збільшенню продажів. Вбудовані аналітичні інструменти дозволяють відстежувати ефективність маркетингових заходів та аналізувати поведінку клієнтів, що є важливим для прийняття обґрунтованих бізнес-рішень.

Серед прикладів успішних інтернет-магазинів на базі WordPress:

- Dr. Scholl's ([drscholls.com](https://www.drsholls.com)) — популярний бренд взуття та ортопедичних виробів;
- DVO Suspension ([dvosuspension.com](https://www.dvosuspension.com)) — виробник велосипедних компонентів;
- Airstream ([airstream.com](https://www.airstream.com)) — відомий виробник кемперів та трейлерів;
- Miga Studio ([migaeyewear.com](https://www.migaeyewear.com)) — онлайн-магазин стильних окулярів.

Такі кейси свідчать про те, що WordPress є потужним інструментом для розвитку електронної комерції, дозволяючи підприємцям легко масштабувати свій бізнес у цифровому середовищі.

Історично WordPress був створений як платформа для ведення блогів, тому він ідеально підходить для проєктів, де контент є основним елементом. Сьогодні блоги та новинні сайти залишаються однією з основних сфер використання цієї CMS.

Завдяки простому інтерфейсу та широким можливостям для форматування контенту, WordPress дозволяє користувачам швидко створювати та публікувати статті, додавати зображення, відео, інтерактивні елементи та інтегрувати сайти з соціальними мережами.

Популярні новинні сайти, створені на базі WordPress (Додаток В):

- Time (time.com) — один із найвідоміших світових журналів із показником 15 мільйонів органічного трафіку та 25 мільйонів загального.
- The PlayStation Blog (blog.playstation.com) — офіційний блог PlayStation із 710 тисячами органічного трафіку та 3 мільйонами загального.
- Spotify For the Record (newsroom.spotify.com) — інформаційний ресурс музичного сервісу Spotify із 319 тисячами органічного трафіку та 1,7 мільйона загального.
- The New York Times (nytimes.com) — один із найбільших новинних сайтів світу з 266 мільйонами органічного та 642 мільйонами загального трафіку [13].

В Україні також є популярні новинні ресурси на базі WordPress, такі як ІТС.ua та Mezha.Media.

Одним з напрямків застосування WordPress є створення сайтів некомерційних організацій. Ці організації часто обмежені в ресурсах, тому для них важлива доступність та простота використання системи. WordPress дозволяє легко створювати та підтримувати веб-сайти, які ефективно передають інформацію про діяльність організації, залучають донорів та волонтерів. Завдяки великій кількості безкоштовних тем та плагінів, WordPress надає некомерційним організаціям можливість створювати професійні веб-сайти без значних фінансових витрат.

Освітні заклади також активно використовують WordPress для створення своїх веб-сайтів. Платформа дозволяє легко створювати навчальні матеріали, форуми для обговорення, а також системи управління навчанням (LMS). Крім того, WordPress інтегрується з різними інструментами для проведення онлайн-тестування та оцінювання студентів. Це робить WordPress ефективним інструментом для організації дистанційного навчання та створення сучасних освітніх платформ.

Державні установи також звертають увагу на WordPress як на зручний інструмент для створення та підтримки своїх веб-сайтів. Платформа дозволяє створювати інформаційні портали, надавати онлайн-послуги та забезпечувати прозорість діяльності державних органів. Завдяки можливості інтеграції з різними системами електронного урядування, WordPress сприяє розвитку електронного урядування та підвищенню ефективності роботи державних органів.

Прикладами таких сайтів є (Додаток ):

- The White House (whitehouse.gov) — офіційний сайт Білого дому;
- NASA (nasa.gov) — сайт Національного управління з аеронавтики та дослідження космічного простору США;
- The Obama Foundation (obama.org) — благодійний фонд колишнього президента США;
- Children's Rights (childrensrights.org) — некомерційна організація, що займається захистом прав дітей [14].

Ці приклади демонструють, що WordPress може бути використаний навіть для критично важливих ресурсів, забезпечуючи високу продуктивність, безпеку та можливість адаптації до вимог сучасного інтернету.

WordPress CMS є універсальною платформою для розробки вебсайтів, яка забезпечує гнучкість, масштабованість та широкий спектр функціональних можливостей. Від сайтів-візиток до складних інтернет-магазинів, від блогів до корпоративних порталів — WordPress залишається одним із найпопулярніших виборів для створення сайтів у всьому світі.

Зважаючи на безперервний розвиток технологій та зростаючу роль цифрових платформ у бізнесі, попит на рішення на базі WordPress буде лише зростати, відкриваючи нові можливості для розвитку як малого, так і великого бізнесу.

### **1.3. Постановка задачі**

Розробка сайту на платформі WordPress є складним і багатоступінчастим процесом, що потребує чіткого планування та визначення конкретних цілей, задач, ресурсів і термінів. Постановка задачі є першим етапом цього процесу і має важливе значення для забезпечення успішної реалізації проекту. Вона включає в себе визначення основних цілей проекту, технічних вимог, доступних ресурсів, очікуваних результатів та ризиків, що можуть виникнути під час розробки.

Основною метою даного проекту є створення вебсайту для компанії або організації, який буде використовувати систему керування контентом WordPress. Сайт повинен бути зручним для користувачів, мати високу продуктивність, швидкість

завантаження та бути оптимізованим для пошукових систем. Крім того, сайт повинен забезпечувати безпеку даних користувачів та зручність адміністрування для внутрішніх користувачів організації.

У результаті розробки повинні бути досягнуті кілька ключових цілей. Сайт має ефективно презентувати компанію, її послуги та продукцію, надавати можливість для взаємодії з користувачами через контактні форми, коментарі та новини, а також збирати дані користувачів, зокрема для підписок на розсилки або збору відгуків. Сайт також повинен бути адаптований для мобільних пристроїв і відповідати сучасним стандартам дизайну.

У рамках цього проекту буде розроблений функціональний сайт, який відповідає всім вимогам бізнесу та технологічним стандартам. Основні задачі, які необхідно вирішити, включають аналіз вимог клієнта, проектування структури сайту, розробку дизайну, налаштування бази даних, програмування та інтеграцію з іншими сервісами, тестування системи, а також впровадження та запуск готового продукту.

На етапі аналізу вимог визначаються потреби користувачів та бізнесу, що включають визначення основних функцій та блоків сайту. Сайт має включати розділи з інформацією про компанію, послуги та продукти, а також можливість для користувачів взаємодіяти з контентом через контактні форми та коментарі. Окрім цього, важливою частиною є SEO-оптимізація для забезпечення високих позицій у пошукових системах.

Проектування структури сайту включає створення карти сайту, що відображає всі основні розділи і підрозділи. Для кожного розділу визначається його функціональність і зміст, а також передбачається зручний інтерфейс для користувачів. Враховується не тільки функціональність сайту, а й зручність навігації, що забезпечить ефективний доступ до основної інформації.

Після проектування структури починається етап розробки дизайну сайту. Дизайн має бути сучасним, функціональним і привабливим для користувачів. Він повинен відповідати брендовим вимогам і вимогам до зручності користування. Окрім цього, дизайн повинен бути адаптивним, що означає, що сайт має коректно

відображатися на різних пристроях — від десктопних комп'ютерів до смартфонів і планшетів.

На етапі розробки сайту на платформі WordPress здійснюється налаштування основних компонентів: вибір теми, налаштування плагінів та інтеграція з необхідними зовнішніми сервісами. Це включає в себе додавання функціональних блоків, налаштування форм для збору даних від користувачів, створення контенту та налаштування адміністративної панелі для зручного управління сайтом. Одним з важливих аспектів є налаштування бази даних для зберігання контенту та користувацьких даних.

Тестування є невід'ємною частиною процесу розробки, оскільки воно дозволяє виявити потенційні проблеми з функціонуванням сайту та забезпечити його коректну роботу. Під час тестування перевіряються всі основні функції сайту, виявляються помилки у роботі плагінів, тестується швидкість завантаження сторінок, перевіряється відповідність дизайну адаптивним вимогам та оптимізація для різних пристроїв. Після тестування проводиться виправлення помилок і доопрацювання функціоналу.

На етапі впровадження та запуску сайт переноситься на сервер, налаштовується його доменне ім'я, перевіряється безпека і проводиться остаточне тестування. Після запуску сайту необхідно забезпечити його технічну підтримку, що включає регулярне оновлення плагінів, перевірку безпеки та контроль за коректністю роботи сайту.

Технічні вимоги для реалізації цього проекту включають налаштування серверного середовища, вибір оптимального хостингу для WordPress, налаштування бази даних MySQL, встановлення необхідних плагінів та використання актуальних версій PHP для забезпечення швидкодії та сумісності. Враховуються також вимоги до безпеки, зокрема використання SSL сертифікатів для захищеного з'єднання та забезпечення захисту від хакерських атак.

Ресурси для реалізації цього проекту включають людські, матеріальні та технічні ресурси. Для виконання проекту необхідна команда фахівців, серед яких веб-розробники, дизайнери, тестувальники, а також проектні менеджери. Важливою

частиною є наявність хостингових ресурсів для розміщення сайту, а також програмних інструментів для розробки і тестування.

Крім того, проект передбачає оцінку ризиків, які можуть виникнути під час розробки. До основних ризиків відносяться проблеми з сумісністю плагінів, можливі помилки в програмуванні, затримки через зміни вимог клієнта або непередбачувані технічні складнощі. Для запобігання цим ризикам необхідно здійснити ретельне планування, регулярне тестування та взаємодію з клієнтом протягом усього процесу розробки.

Постановка задачі є важливим етапом проекту, оскільки вона визначає напрямок і структуру роботи, а також дозволяє чітко розуміти, які результати необхідно досягти на кожному етапі розробки. Визначення цілей, задач, технічних вимог, ресурсів і ризиків є основою для успішної реалізації проекту і забезпечення його високої якості.

## **Висновки до розділу 1**

Було описано предметне середовище необхідне для реалізації проекту. Згадано, що система управління контентом WordPress базується на клієнтській, серверній частинах та базі даних. Клієнтська частина в свою чергу реалізується за допомогою HTML, CSS та JavaScript. Основу серверної частини WordPress становить мова програмування PHP, різноманітні фреймворки та бібліотеки та файлова система. Визначено що процес розробки може відбуватися як на локальному, так і на віддаленому середовищах.

Було розглянуто успішні сценарії сайтів розроблених на WordPress CMS. Виокремлено, що серед популярних успішних сценаріїв є вебсайти-візитки, інтернет-магазини, новинні сайти та блоги та що система використовується як основа для сайтів комерційних та некомерційних, урядових та неурядових структур та організацій. Наведено приклади кейсів сайтів світових відомих брендів та організацій з аудиторіями більше 1 млн. користувачів.

Було поставлено завдання розробити функціональний сайт, який відповідає всім вимогам бізнесу та технологічним стандартам.

## РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МЕТОДОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1. Аналіз предметної області (вхідні та вихідні дані)

Процес розробки вебсайту на платформі WordPress є складним технологічним завданням, яке вимагає аналізу предметної області з урахуванням як вхідних, так і вихідних даних. Вхідні дані формують основу для проектування, розробки та впровадження сайту, тоді як вихідні дані є результатом реалізації проекту і визначають його функціональність, зовнішній вигляд, ефективність роботи та відповідність потребам користувачів і бізнес-цілям замовника.

Вхідні дані є початковою інформацією, яка необхідна для розробки сайту. Вони охоплюють технічні, змістовні та бізнес-вимоги, які формують основу проекту. Першочергово визначаються цілі та завдання проекту, які залежать від сфери діяльності замовника, типу контенту, що буде публікуватися на сайті, а також від очікувань цільової аудиторії. Наприклад, для корпоративного сайту важливим є представлення бренду, контактної інформації та опису послуг. Для інтернет-магазину основною метою є продаж товарів, що передбачає іншу структуру сайту та набір функцій.

Іншим важливим вхідним параметром є аналіз цільової аудиторії. Розробники повинні чітко розуміти, хто буде користуватися сайтом, які їхні потреби та очікування, яким чином вони взаємодіють з контентом і які пристрої використовують для доступу до вебресурсу. Цей аналіз впливає на вибір дизайну, структури сайту, навігаційних елементів та інтеграцію додаткових функцій.

Технічні вимоги є критично важливими вхідними даними. Вони визначають платформу для хостингу, бажану швидкість завантаження сторінок, необхідні інтеграції з іншими системами та плагінами, а також питання безпеки даних. Вибір хостинг-провайдера та доменного імені також є частиною початкових даних, які визначають подальшу інфраструктуру сайту.

Контентні вимоги включають обсяг і тип контенту, який планується розміщувати на сайті. Це можуть бути текстові статті, мультимедійні матеріали (зображення, відео), документи для завантаження та інші елементи. У цьому



контексті важливо також враховувати мовні параметри сайту, якщо передбачається підтримка кількох мов.

Дизайнерські вимоги охоплюють стилістичні рішення, кольорову палітру, шрифти та загальний вигляд сайту. Вхідними даними для цього етапу можуть бути корпоративні стандарти дизайну, логотипи та інші графічні елементи бренду. Крім того, визначаються вимоги до адаптивного дизайну, щоб забезпечити коректне відображення сайту на різних пристроях, включаючи настільні комп'ютери, планшети та смартфони.

Важливим аспектом вхідних даних є також правові та нормативні вимоги. Сайт повинен відповідати законам про захист персональних даних, авторське право, використання файлів cookie та іншим регуляторним вимогам. Вони формують рамки для роботи з персональними даними користувачів та впровадження політики конфіденційності.

Вихідні дані є результатом реалізації проекту, який повинен відповідати вхідним вимогам та цілям. Головним вихідним продуктом є функціональний вебсайт, який має чітко визначену структуру, інтерфейс користувача, набір функцій та інтеграцій.

Функціональні вихідні дані включають основні можливості сайту, такі як навігація, пошук, взаємодія з контентом, реєстрація користувачів, інтеграція платіжних систем та інші елементи, що забезпечують зручність використання сайту для відвідувачів. Вихідний сайт повинен відповідати очікуванням користувачів щодо зручності та інтуїтивності інтерфейсу. Це особливо важливо для сайтів, орієнтованих на широку аудиторію, таких як інтернет-магазини чи медіа-ресурси.

Зовнішній вигляд сайту є ще одним важливим результатом. Дизайн має бути адаптований до потреб користувачів і відповідати корпоративному стилю замовника. Виконаний дизайн повинен бути гармонійним, естетично привабливим і водночас функціональним. Він також має враховувати вимоги до адаптивного дизайну для забезпечення коректного відображення на різних пристроях.

Важливим результатом розробки є оптимізація швидкості завантаження сайту. Швидкість є критичним фактором для успіху вебсайту, оскільки повільне

завантаження може призвести до втрати відвідувачів. Оптимізація включає використання кешування, стиснення зображень, оптимізацію коду та інші технічні заходи.

Іншим вихідним результатом є система управління контентом (CMS), яка повинна бути простою у використанні для адміністраторів сайту. WordPress надає гнучкі інструменти для створення та редагування контенту, що дозволяє швидко оновлювати інформацію, додавати нові сторінки, публікувати статті чи медіа-контент.

Безпека сайту є ще одним важливим вихідним результатом. Реалізація проекту повинна враховувати всі можливі загрози, такі як атаки хакерів, злом паролів або викрадення даних. Для цього використовуються заходи, включаючи регулярне оновлення системи та плагінів, встановлення сертифіката SSL, впровадження системи захисту від спаму та інші методи забезпечення безпеки.

Кінцевий результат розробки сайту також включає його інтеграцію з іншими сервісами, такими як аналітичні системи, CRM, системи електронної пошти або соціальні мережі. Ці інтеграції дозволяють ефективно управляти даними користувачів, аналізувати трафік та проводити маркетингові кампанії.

Після завершення розробки сайту важливим вихідним результатом є також документація проекту. Вона включає технічні специфікації, інструкції з адміністрування, рекомендації щодо подальшого розвитку та підтримки сайту.

Таким чином, аналіз предметної області розробки сайту на WordPress дозволяє визначити ключові вхідні та вихідні дані, які впливають на кожен етап реалізації проекту. Вхідні дані формують основу для створення сайту, забезпечуючи чітке розуміння вимог та очікувань замовника, тоді як вихідні дані є результатом, що визначає якість, функціональність та ефективність кінцевого продукту.

## **2.2. Створення плану розробки системи (план розробки/впровадження обов'язкових складових системи для етапу мінімального життєздатного продукту)**

Створення вебсайту на базі WordPress із метою реалізації мінімально життєздатного продукту (Minimum Viable Product, MVP) є стратегічним процесом, спрямованим на забезпечення базової функціональності, яка дозволяє продукту швидко вийти на ринок [15]. Основне завдання MVP — надати користувачам основний функціонал, що відповідає їхнім потребам, і створити основу для подальшого розвитку сайту на основі зворотного зв'язку.

Розробка MVP починається з визначення мети проєкту. Важливо зрозуміти, яку проблему вирішує сайт, хто є його основною аудиторією та які функції критично необхідні для початкового запуску. Наприклад, сайт може бути створений для електронної комерції, надання інформаційних послуг або створення платформи для комунікації.

На цьому етапі важливо також провести аналіз ринку та конкурентів, що дозволяє виявити сильні та слабкі сторони існуючих рішень у відповідній ніші. Це допомагає визначити, які функції можна використати як конкурентну перевагу. Зазвичай аналіз охоплює функціональні можливості конкурентів, зручність користувацького інтерфейсу, дизайн та інтеграції, які вони використовують.

Після цього формуються основні функціональні вимоги. Вони можуть включати створення базової структури сайту з головною сторінкою, сторінкою про компанію, контактною інформацією, можливістю реєстрації користувачів, пошуку по сайту та інтеграції із соціальними мережами.

Планування структури сайту починається зі створення ієрархії сторінок. Вона повинна бути логічною та інтуїтивно зрозумілою для користувачів. Наприклад, сайт може містити головну сторінку з коротким описом продукту або компанії, сторінку «Про нас» з історією та досягненнями, сторінку «Контакти» з формою зворотного зв'язку та контактною інформацією, а також каталог товарів або послуг, якщо це інтернет-магазин.

Наступним кроком є вибір теми для WordPress. Для MVP найчастіше використовують готові теми, які забезпечують адаптивність, сумісність із популярними плагінами та простоту налаштування. Наприклад, можна вибрати тему, яка відповідає бізнес-цілям проєкту, має сучасний дизайн і дозволяє легко модифікувати елементи сайту.

Встановлення WordPress є наступним важливим етапом. Це можна зробити як на локальному сервері для тестування, так і на хостинговій платформі для запуску у виробниче середовище. Після встановлення слід налаштувати базову конфігурацію, яка включає вибір мови, введення назви сайту та інших параметрів.

Далі встановлюється обрана тема. Це відбувається через панель адміністратора WordPress, де тема завантажується, активується та налаштовується відповідно до потреб проєкту. Наступним кроком є встановлення ключових плагінів для розширення функціональності сайту. Для забезпечення безпеки можна використовувати плагіни, що захищають від атак і спаму. Оптимізація сайту для пошукових систем здійснюється за допомогою SEO-плагінів, а інтеграція з соціальними мережами спрощує взаємодію з аудиторією. Якщо сайт передбачає електронну комерцію, важливо налаштувати плагін для онлайн-продажів, наприклад WooCommerce, який дозволяє додавати товари, керувати замовленнями та інтегрувати платіжні системи.

Після налаштування базових компонентів розробляється навігація сайту. Меню створюється у панелі адміністратора, де додаються всі необхідні сторінки та налаштовується їх порядок. Важливо забезпечити зручне розташування меню, щоб користувачі могли швидко знаходити потрібну інформацію.

Створення сторінок є наступним етапом. Головна сторінка зазвичай містить ключову інформацію про проєкт, заклики до дії та посилання на інші важливі розділи. Сторінка контактів містить форму зворотного зв'язку, інтегровану за допомогою відповідного плагіна, що дозволяє користувачам швидко зв'язатися з адміністратором сайту. Якщо сайт є інтернет-магазином, додаються сторінки для каталогу товарів, де кожен товар має окрему сторінку з описом, ціною та кнопкою для покупки.

Перед запуском MVP необхідно провести тестування, щоб перевірити коректність роботи сайту. Тестуються всі основні функції, швидкість завантаження сторінок, а також робота сайту на різних пристроях і в різних браузерах. Важливо перевірити роботу форм, посилань та інших інтерактивних елементів.

Після завершення тестування MVP можна запускати у виробниче середовище. На цьому етапі важливо забезпечити збирання зворотного зв'язку від користувачів, що дозволить оцінити, наскільки сайт відповідає їхнім очікуванням та які функції необхідно додати або вдосконалити у майбутньому. Зворотний зв'язок можна отримувати через форми опитування, коментарі або аналітичні інструменти, які дозволяють відстежувати поведінку користувачів на сайті.

Таким чином, розробка мінімально життєздатного продукту на WordPress є складним, але ефективним процесом, який дозволяє швидко вивести продукт на ринок, мінімізуючи ризики та оптимізуючи витрати на розробку. MVP стає основою для подальшого розвитку та вдосконалення вебсайту, забезпечуючи можливість адаптації до змінних потреб користувачів і ринку.

### **2.3. Методології розробки**

Розробка сайту є багатогранним процесом, що вимагає ретельного планування, організації роботи команди та вибору оптимальної методології для досягнення поставлених цілей. Залежно від того, чи працює над проектом одна людина, чи команда розробників, підхід до організації процесу може суттєво відрізнятись.

Для невеликих проектів або індивідуальної роботи часто використовується послідовний підхід Waterfall, який передбачає проходження через усі етапи розробки – від планування до впровадження – у суворій послідовності [16]. Втім, для більш складних проектів, особливо за участі кількох фахівців, більш доцільно застосовувати гнучкі методології, такі як Scrum, Kanban або їх комбінацію – Scrumban.

Waterfall є класичною методологією управління проектами, що базується на лінійному виконанні етапів. Спочатку створюється детальний план проекту, що включає визначення цілей, ключових функцій сайту, аналіз цільової аудиторії та вибір технологічної платформи для розробки. Наступним етапом є розробка дизайну,

де визначається зовнішній вигляд сайту, його структура та інтерфейс користувача. Далі йде етап безпосередньої розробки, під час якого створюється функціональний сайт, що відповідає технічним вимогам. Завершальним етапом є тестування та впровадження, після чого сайт стає доступним для кінцевих користувачів.

Хоча цей підхід є структурованим та забезпечує чітке розуміння всіх етапів розробки, його основним недоліком є низька гнучкість. Внесення змін до проекту на пізніх етапах є складним і часто потребує повернення до попередніх стадій, що може суттєво вплинути на строки реалізації.

Для більш складних і динамічних проєктів із високою ймовірністю змін під час розробки краще підходять гнучкі методології. Однією з найпоширеніших таких методологій є Scrum [17]. В основі цього підходу лежить ітеративний процес роботи, поділений на короткі цикли – спринти. Кожен спринт має чітко визначені завдання та цілі, які команда повинна досягти за фіксований проміжок часу, зазвичай від одного до чотирьох тижнів. На початку кожного циклу відбувається детальне планування, де визначаються завдання, які потрібно виконати. Після завершення спринту команда проводить огляд виконаної роботи, аналізує успіхи та недоліки, щоб внести корективи в наступні ітерації. Такий підхід дозволяє командам зосередитися на досягненні конкретних результатів за короткий час і забезпечує швидку реакцію на зміни в проєктах, де вимоги замовника можуть змінюватися в процесі роботи.

Kanban є ще однією популярною гнучкою методологією, яка базується на візуалізації робочого процесу [18]. Основним інструментом є дошка Kanban, яка поділена на стовпці, що відповідають різним етапам виконання завдань, наприклад: «Очікує виконання», «У процесі», «Тестування», «Завершено».

Картки із завданнями переміщуються між стовпцями в міру їх виконання, що дозволяє легко відстежувати прогрес роботи та виявляти вузькі місця. Kanban не має фіксованих ітерацій, як у Scrum, тому команди можуть гнучко керувати завданнями залежно від пріоритетів та наявних ресурсів.

Основні переваги Kanban полягають у його простоті, можливості швидко адаптувати процес під нові умови та забезпеченні високої прозорості для всіх учасників проєкту [19].

Для складних проектів, де необхідно ефективно поєднати переваги ітеративного підходу Scrum з гнучкістю Kanban, застосовується комбінована методологія Scrumban [20]. Ця методологія дозволяє зберегти чітку структуру планування та контроль над прогресом через спринти, які характерні для Scrum, і одночасно забезпечує гнучкість завдяки використанню Kanban-дошки для візуалізації та управління потоком завдань. Таке поєднання дозволяє адаптувати процес розробки до змінних вимог і непередбачуваних ситуацій, що часто трапляються в складних проектах.

Основною перевагою Scrumban є можливість зберігати чітке планування на етапах проекту, характерне для Scrum, і в той же час забезпечити візуальний моніторинг завдань, що дозволяє команді оперативно реагувати на зміни. Наприкінці кожного спринту команда проводить ретроспективу для оцінки досягнутих результатів і визначення напрямів для подальшої роботи, а також коригує плани на наступний спринт. Це дозволяє забезпечити послідовність у виконанні завдань і одночасно бути гнучкими в управлінні змінами.

Важливим елементом є використання Kanban-дошки для візуалізації потоку завдань. Дошка дозволяє команді побачити всі завдання, їхній статус (зроблено, в процесі, тестування тощо) та ідентифікувати вузькі місця в процесі. Кожне завдання має свою картку, і команда може легко відстежувати прогрес у виконанні, виявляти затримки та коригувати пріоритети. Візуалізація завдань також допомагає створити відчуття прозорості в процесі роботи, що важливо для покращення комунікації між членами команди і замовником.

Scrumban особливо ефективний для проектів, де вимоги можуть змінюватися в процесі розробки або якщо необхідно впроваджувати нові функції після кожного циклу спринтів. Цей підхід дозволяє зберігати гнучкість і адаптивність, що є важливими для забезпечення швидкого реагування на зміни умов або виявлені проблеми. Замість того, щоб суворо дотримуватися фіксованого плану, команда може адаптувати свої дії залежно від нових пріоритетів або відгуків, що дає можливість досягати високих результатів у динамічному середовищі.

Одним із основних аспектів цієї методології є регулярні щоденні зустрічі, що дозволяють команді обговорювати поточний стан проекту, виявляти проблеми та коригувати плани. Під час цих зустрічей учасники можуть обмінюватися інформацією про прогрес, визначати завдання для наступного дня і в разі потреби переглядати пріоритети або дії. Такий регулярний обмін допомагає підтримувати командний дух і знижує ризик непорозумінь, що можуть виникати в процесі виконання завдань.

Крім того, Scrumban дозволяє комбінувати найкращі практики обох методологій для оптимізації робочих процесів. Якщо Scrum дозволяє розподілити роботу на короткі, чітко визначені етапи, то Kanban дає змогу зберігати гнучкість і швидко коригувати напрямок роботи в межах цих етапів. Така структура дозволяє команді працювати з максимальним ефектом, швидко реагувати на нові вимоги та, в той же час, забезпечувати високий рівень організації роботи.

Цей підхід особливо підходить для проектів, де складність і обсяг завдань може варіюватися, і важливо, щоб команда мала можливість адаптуватися до нових умов без втрати якості. Наприклад, у розробці складних вебсайтів або програмних продуктів, де в процесі роботи можуть виникати нові технічні вимоги, важливо швидко змінювати план дій або коригувати пріоритети для забезпечення досягнення бажаних результатів.

Завдяки поєднанню структури планування і гнучкості реагування, Scrumban дозволяє командам бути більш продуктивними, забезпечувати високоякісні результати та підтримувати адаптивність до змін у зовнішньому середовищі. Це робить методологію ефективною в багатьох сферах, включаючи розробку програмного забезпечення, вебдизайн, маркетинг і багато інших галузей, де швидкість реагування на зміни та постійне вдосконалення процесів є критичними для успіху.

Застосування гнучких методологій у процесі розробки сайту на WordPress дозволяє ефективно організувати роботу команди, забезпечити прозорість процесу та швидко адаптуватися до змін. Вибір конкретного підходу залежить від особливостей проекту, складу команди та вимог замовника.



Scrum забезпечує чітку структуру та поступове досягнення результатів через спринти, що є важливим для проектів із визначеними етапами. Kanban надає гнучкість та можливість швидкого реагування на зміни, що є важливим для динамічних проектів із постійними змінами вимог. Scrumban поєднує обидва підходи, дозволяючи зберігати баланс між структурою та гнучкістю.

Таким чином, правильний вибір методології та її адаптація під конкретні умови проекту є ключовими факторами успішної реалізації розробки сайту на WordPress, що відповідає вимогам замовника та очікуванням кінцевих користувачів.

## **Висновки до розділу 2**

У розділі було здійснено детальний аналіз предметної області, що включав визначення та обробку вхідних і вихідних даних. Серед вхідних даних було виокремлено технічні, змістовні, бізнес-вимоги, аналіз цільової аудиторії, контентні, дизайнерські та правові вимоги. Вихідним продуктом проекту є функціонуючий веб-сайт, який має чітко визначену структуру, інтерфейс користувача, набір функцій та інтеграцій.

Було визначено основні кроки до реалізації мінімально життєздатного продукту. Серед кроків можна виділити визначення мети проекту, аналіз ринку та конкурентів, визначення основних функціональних вимог, створення ієрархії сторінок, встановлення системи, вибір і встановлення теми, плагінів, розробка навігації сайту, створення та наповнення сторінок, тестування і запуск проекту.

Було висвітлено методології розробки сайту. Для командної роботи запропоновано використати комбінацію Scrum та Kanban (Scrumban).

## РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 3.1. Засоби розробки

Ядро WordPress є фундаментальною складовою платформи, що забезпечує базову структуру та функціональність для створення веб-сайтів [21]. Воно включає в себе систему управління контентом, яка дозволяє користувачам додавати, редагувати та публікувати різні типи матеріалів. Крім того, ядро надає шаблони сторінок, які визначають загальний вигляд сайту, та механізм для встановлення та управління плагінами, що розширюють можливості платформи.

Теми WordPress представляють собою готові дизайни для веб-сайтів, які визначають їхній візуальний стиль та структуру. Вони включають в себе шаблони для різних типів сторінок, стилі оформлення, меню навігації та інші елементи інтерфейсу користувача. Теми можуть бути безкоштовними або платними, а їхній рівень складності варіюється від простих до високоадаптивних.

Плагіни є розширеннями для WordPress, які додають новий функціонал до базової платформи. Вони можуть виконувати різноманітні завдання, такі як оптимізація пошукових систем (SEO), створення форм зворотного зв'язку, додавання функцій електронної комерції, забезпечення безпеки сайту тощо. Плагіни можуть бути розроблені як самостійними розробниками, так і командою WordPress, і їх кількість та різноманітність є величезною.

PHP є серверною мовою програмування, яка лежить в основі WordPress. Вона використовується для створення логіки роботи сайту, обробки запитів користувачів, взаємодії з базою даних та генерації HTML-коду. Знання PHP дозволяє розробникам створювати власні теми та плагіни, а також налаштовувати поведінку існуючих компонентів.

JavaScript є клієнтською мовою програмування, яка використовується для додавання інтерактивності на веб-сторінки. В контексті WordPress, JavaScript може бути використаний для створення динамічних елементів, таких як слайдери, меню, модальні вікна, а також для покращення користувацького досвіду взаємодії з сайтом.

Інструменти розробки є додатковими програмами та ресурсами, які допомагають розробникам ефективніше працювати з WordPress. Вони включають в себе редактори коду для написання та відлагодження PHP та JavaScript, системи контролю версій для управління змінами в коді, фреймворки для прискорення розробки, а також інструменти для тестування та оптимізації веб-сайтів.

### **3.2. Вимоги до технічного та програмного забезпечення**

Вимоги до технічного забезпечення сервера для сайту на WordPress залежать від багатьох факторів, таких як очікуваний трафік, кількість відвідувачів, складність сайту та використовувані плагіни. Однак, існують деякі базові вимоги, які необхідно враховувати:

- Процесор. Чим більше ядер і вища частота процесора, тим краще. Для невеликих сайтів достатньо процесора середньої потужності, але для сайтів з високим трафіком потрібен більш потужний процесор.
- Оперативна пам'ять (RAM). Кількість оперативної пам'яті впливає на швидкість роботи сайту. Для невеликих сайтів достатньо 2-4 ГБ RAM, але для великих сайтів може знадобитися 8 ГБ або більше.
- Місце на диску. Обсяг дискового простору залежить від розміру вашого сайту, кількості медіафайлів та резервних копій. Для невеликих сайтів достатньо 1-5 ГБ, але для великих сайтів може знадобитися більше 20-50 ГБ.
- Пропускна здатність. Швидкість передачі даних впливає на швидкість завантаження сайту. Для невеликих сайтів достатньо 100 Мбіт/с, але для сайтів з високим трафіком може знадобитися більше 10 Гбіт/с [22].

Вимоги до програмного забезпечення сайту на WordPress включають кілька аспектів:

- Веб-сервер. Apache та Nginx є найпопулярнішими веб-серверами для WordPress, однак підтримуваними також є Angie, LiteSpeed Web Server, OpenLiteSpeed [23].
- База даних. WordPress може використовувати MySQL (версії 8.0 або вище) або MariaDB (версії 10.5 або вище) для зберігання даних.

- PHP. Версія PHP повинна бути не нижче 7.4 для оптимальної продуктивності та безпеки [24].

Крім того, важливо враховувати такі фактори, як наявність SSL-сертифікату для забезпечення безпеки, систему резервного копіювання даних та регулярне оновлення ядра WordPress та тем з плагінами.

### 3.3. Опис реалізації

Етап **аналізу та формулювання вимог** є фундаментом успішної розробки веб-сайту на платформі WordPress. Він передбачає детальне дослідження та структурування інформації, необхідної для створення ресурсу, який повністю відповідатиме поставленим завданням та очікуванням замовника.

На цьому етапі здійснюється комплексний аналіз, що охоплює як внутрішні, так і зовнішні фактори, які впливають на розробку сайту. До внутрішніх факторів належать цілі та завдання, які ставить перед собою замовник, а також ресурси, якими він володіє. Зовнішні фактори включають аналіз цільової аудиторії, конкурентного середовища та загальних тенденцій у веб-дизайні та розробці.

Аналіз цілей та завдань полягає у чіткому визначенні мети створення сайту. Це може бути інформування користувачів, продаж товарів або послуг, створення онлайн-спільноти тощо. Важливо сформулювати цілі конкретно та вимірювано, щоб мати можливість оцінити ефективність розробленого ресурсу.

Аналіз цільової аудиторії передбачає детальне дослідження потреб, інтересів та поведінки користувачів, на яких орієнтований сайт. Це дозволяє розробити інтерфейс та контент, які будуть максимально релевантними для цільової аудиторії.

Конкурентний аналіз дає змогу оцінити сильні та слабкі сторони конкурентів, визначити їхні успішні практики та уникнути типових помилок. На основі цього аналізу формується унікальна пропозиція, яка відрізнятиме сайт від конкурентів.

Формулювання вимог є результатом проведеного аналізу. Вимоги можуть бути функціональними (наприклад, наявність форми зворотного зв'язку, каталогу товарів) та нефункціональними (наприклад, сучасний дизайн, висока швидкість завантаження сторінок). Важливо, щоб вимоги були чіткими, конкретними та вимірюваними.

Результатом етапу аналізу та формулювання вимог є детально розроблене технічне завдання, яке слугує основою для подальшої розробки сайту. Технічне завдання містить детальний опис функціоналу, інтерфейсу, структури сайту, а також вимоги до дизайну, технологій та інших аспектів розробки.

Етап **розробки концепції та структури сайту** є важливим у процесі створення веб-ресурсу. Саме на цьому етапі формується бачення майбутнього сайту, його цілі та призначення. Концепція сайту – це своєрідний каркас, який визначає його зовнішній вигляд, стиль, функціональність та зміст. Структура сайту, своєю чергою, визначає логічну організацію інформації, яка забезпечує зручну навігацію для користувачів.

Процес розробки концепції та структури сайту можна розглядати як подорож від загальної ідеї до деталізованого плану. Спочатку формулюється чітка мета сайту, яка визначає його основну функцію. Чи буде це онлайн-магазин, інформаційний портал, блог чи соціальна мережа? Відповідь на це питання впливає на всі подальші рішення.

Паралельно з формулюванням мети проводиться глибокий аналіз цільової аудиторії. Хто є потенційними користувачами сайту? Які їхні потреби, інтереси та очікування? Відповіді на ці питання дозволяють створити сайт, який буде максимально релевантним для користувачів.

Наступним кроком є визначення функціоналу сайту. Які функції він повинен виконувати? Які інструменти та можливості необхідно передбачити для досягнення поставлених цілей? Відповіді на ці питання дозволяють скласти детальний список функцій, які будуть реалізовані на сайті.

Після того, як визначені мета, цільова аудиторія та функціонал, розробляється інформаційна архітектура сайту. Це своєрідний план, який відображає логічну структуру сайту, включаючи головні розділи, підрозділи та сторінки. Добре продумана інформаційна архітектура дозволяє користувачам легко орієнтуватися на сайті та швидко знаходити потрібну інформацію.

Паралельно з розробкою інформаційної архітектури визначається загальний стиль сайту. Вибирається колірна гама, шрифти, ілюстрації та інші елементи

візуального оформлення, які створюють єдиний образ сайту та відповідають його концепції.

Одним із важливих етапів є створення прототипу сайту. Прототип – це спрощена версія майбутнього сайту, яка дозволяє оцінити його структуру та зручність використання. На основі прототипу вносяться необхідні корективи в дизайн та структуру сайту.

Результатом цього етапу є детально розроблена концепція сайту та інформаційна архітектура, які слугують основою для подальшої розробки. Цей етап є критично важливим, оскільки саме він визначає успіх всього проекту. Добре продумана концепція та структура дозволяють створити сайт, який буде не тільки привабливим зовні, але й зручним та ефективним у використанні.

Етап **розробки дизайну** сайту є одним з найкреативніших та найвідповідальніших у процесі створення веб-ресурсу. Саме дизайн формує перше враження користувача про сайт, впливає на його сприйняття та взаємодію з контентом.

Розробка дизайну – це не просто створення красивих картинок, а комплексний процес, який включає в себе аналіз цільової аудиторії, розробку концепції візуального стилю, створення макетів та їх адаптацію під різні пристрої.

Аналіз цільової аудиторії є визначальним фактором при розробці дизайну. Розуміння потреб, інтересів та вподобань користувачів дозволяє створити дизайн, який буде резонувати з ними на емоційному рівні.

Розробка концепції візуального стилю передбачає визначення загального настрою, кольорової гами, типографіки та інших елементів дизайну, які будуть об'єднувати всі сторінки сайту в єдине ціле. Концепція має відповідати загальній ідеї сайту та відображати його унікальність.

Створення макетів – це процес перетворення концепції дизайну в конкретні візуальні образи. Дизайнер створює макети головної сторінки, внутрішніх сторінок, форм, кнопок та інших елементів інтерфейсу. Макети дозволяють оцінити, як буде виглядати готовий сайт, і внести необхідні корективи.

Адаптивна верстка – це процес створення дизайну, який буде коректно відображатися на різних пристроях: від великих моніторів до маленьких екранів смартфонів [25]. Адаптивний дизайн забезпечує комфортне користування сайтом незалежно від того, з якого пристрою користувач заходить на сайт.

Тестування дизайну проводиться для того, щоб оцінити, наскільки дизайн відповідає поставленим завданням і чи зручний він для користувачів. Тестування може проводитися як за допомогою спеціальних інструментів, так і за допомогою тестування з участю реальних користувачів.

Результатом етапу розробки дизайну є комплект макетів, які готові до верстки. Добре розроблений дизайн не тільки приваблює користувачів, але й сприяє досягненню цілей сайту, підвищує конверсію та покращує користувацький досвід.

**Технічна підготовка** сайту на WordPress є критичним етапом, що передуює запуску веб-ресурсу. Цей процес включає в себе комплекс заходів, спрямованих на створення стійкого, безпечного та ефективного середовища для функціонування сайту.

На першому етапі проводиться ретельний аналіз вимог до сайту. Визначаються необхідні функціональні можливості, очікуване навантаження, рівень безпеки та інші параметри. Ця інформація є основою для вибору оптимальної конфігурації сервера, встановлення необхідних плагінів та тем.

Одним з ключових рішень на цьому етапі є вибір хостингу та домену. Хостинг – це сервер, на якому буде розміщено сайт, а домен – це його унікальна адреса в мережі Інтернет. Від якості хостингу залежить швидкість завантаження сайту, його доступність та стабільність. Тому до вибору хостинг-провайдера необхідно підходити з особливою увагою, враховуючи такі фактори як надійність, швидкість, технічна підтримка та вартість. Домен слід обирати коротким, запам'ятовуваним та релевантним до тематики сайту.

Налаштування SSL-сертифікату є невід'ємною частиною технічної підготовки сучасного веб-сайту [26]. SSL-сертифікат забезпечує шифрування передачі даних між сервером та браузером користувача, що є особливо важливим для сайтів, які обробляють конфіденційну інформацію, наприклад, дані для входу, платіжні дані

тощо. Крім того, наявність SSL-сертифікату позитивно впливає на SEO-оптимізацію сайту та підвищує довіру користувачів.

Процес налаштування SSL-сертифікату включає в себе кілька етапів. Спочатку необхідно вибрати тип сертифікату. Існують різні типи сертифікатів, які відрізняються рівнем перевірки та вартістю. Найпоширенішими є Domain Validation (DV), Organization Validation (OV) та Extended Validation (EV). DV-сертифікати є найпростішими та найдешевшими, але вони не підтверджують ідентичність організації. OV-сертифікати підтверджують ідентичність організації, а EV-сертифікати – найвищий рівень перевірки, який дозволяє відображати повну назву компанії в адресному рядку браузера.

Після вибору типу сертифікату необхідно генерувати CSR (Certificate Signing Request) – спеціальний запит на видачу сертифікату. CSR містить інформацію про домен, на який буде встановлений сертифікат, та криптографічні ключі.

Далі CSR надсилається до центру сертифікації (CA), який перевіряє надану інформацію та видає сертифікат. Після отримання сертифікату його необхідно встановити на сервер. Спосіб установки залежить від типу хостингу та панелі керування. Зазвичай, це робиться через панель керування хостингом або за допомогою FTP.

Крім того, необхідно налаштувати веб-сервер для використання SSL-сертифікату. Це включає в себе вказання шляхів до файлів сертифіката та приватного ключа, а також конфігурацію віртуальних хостів.

Після установки сертифікату необхідно перевірити його працездатність. Для цього можна скористатися спеціальними онлайн-сервісами або інструментами браузера. Якщо все налаштовано правильно, у адресному рядку браузера з'явиться зелений замок, що свідчить про безпечне з'єднання.

Важливо зазначити, що процес налаштування SSL-сертифікату може відрізнитися в залежності від хостинг-провайдера та типу сервера.

Наступним етапом є налаштування бази даних. База даних зберігає всю інформацію про ваш сайт, таку як записи, коментарі, налаштування та інше. Важливо



правильно налаштувати базу даних, щоб забезпечити оптимальну продуктивність сайту.

Результатом етапу технічної підготовки сайту на WordPress є створення стабільного та безпечного середовища для функціонування веб-ресурсу. Завдяки ретельному аналізу вимог, вибору оптимального хостингу та домену, налаштуванню бази даних та встановленню SSL-сертифікату, забезпечується надійна робота сайту, захист переданих даних та підвищення довіри користувачів.

Процес **встановлення та першого налаштування системи WordPress** є ключовим етапом у створенні веб-сайту. Після того, як обрано хостинг, зареєстровано домен та налаштовано базу даних, можна переходити безпосередньо до встановлення CMS.

На першому етапі здійснюється завантаження інсталятора WordPress. Зазвичай, файли інсталятора можна завантажити з офіційного сайту WordPress. Далі, за допомогою FTP-клієнта [27] або файлового менеджера хостингу, ці файли завантажуються у кореневу директорію сайту.

Наступним кроком є запуск інсталятора. Для цього в адресному рядку браузера вводиться домен сайту з додаванням шляху до папки з інсталятором (наприклад, [https://ваш\\_домен/wp-admin/install.php](https://ваш_домен/wp-admin/install.php)) [28]. Система запропонує ввести необхідну інформацію: мову встановлення, назву сайту, логін, пароль та електронну пошту адміністратора (Додаток Е). Важливо обрати надійний пароль, щоб захистити свій сайт від несанкціонованого доступу.

Після заповнення всіх полів, запускається процес встановлення. Система автоматично створить необхідні таблиці в базі даних, встановить файли WordPress та завершить інсталяцію. Після успішного завершення встановлення користувач буде перенаправлений на сторінку входу в адміністративну панель.

Результатом встановлення та першого налаштування WordPress є функціональний сайт, готовий до заповнення контентом. Важливо розуміти, що це лише перший крок у створенні веб-ресурсу. Після завершення базових налаштувань необхідно продовжити роботу над дизайном, контентом та SEO-оптимізацією сайту.

**Вибір теми та плагінів** є фундаментальним етапом у створенні сайту на платформі WordPress. Ці елементи не лише визначають зовнішній вигляд веб-ресурсу, а й формують його функціональні можливості, взаємодію з користувачем та загальне враження.

Тема WordPress – це своєрідний каркас сайту, що визначає його загальний вигляд, макет сторінок, розташування елементів та інші візуальні аспекти. Від вдалого вибору теми залежить перше враження від сайту, а також зручність його використання.

Плагіни WordPress – це модулі, які розширюють функціональні можливості платформи. Завдяки плагінам можна реалізувати різноманітні функції, такі як створення форм зворотного зв'язку, галерей, інтернет-магазинів, оптимізацію для пошукових систем та багато іншого.

Комбінація теми Blocksy [29] та плагіна Elementor [30] є одним з популярних та ефективних рішень для створення сучасних веб-сайтів на WordPress. Ця тема відрізняється високою швидкістю завантаження, гнучкими налаштуваннями та сумісністю з більшістю популярних плагінів. Elementor є одним з найпопулярніших візуальних конструкторів сторінок для WordPress. Він дозволяє створювати складні макети сторінок за допомогою інтуїтивно зрозумілого інтерфейсу. Він пропонує широкий набір готових блоків та елементів, які можна легко налаштовувати.

Додаткові плагіни, які будуть встановлені (Додаток):

- **Cyr-to-lat** — цей плагін автоматично транслітерує кириличні символи в латинські, що покращує SEO та зручність для користувачів, які використовують різні мови [31].
- **Polylang** — призначений для створення багатомовних сайтів, що особливо корисно для проектів, орієнтованих на міжнародну аудиторію [32]. В нашому випадку буде використовуватися українська та англійські локалізації сайту.
- **Contact Form 7** — цей плагін дозволяє легко створювати різноманітні форми зворотного зв'язку, які можна налаштовувати під свої потреби [33].

- Flamingo — плагін, який інтегрується з Contact Form 7 і дозволяє зберігати всі відправлені через форми повідомлення в базі даних для подальшого аналізу [34].
- Open Graph Protocol Framework — дозволяє налаштувати відображення попереднього перегляду ваших сторінок у соціальних мережах [35].
- Post Views Counter — дозволяє відстежувати кількість переглядів кожної сторінки і публікації на сайті [36].
- WPCode Lite — цей плагін дозволяє додавати на сайт власний код без необхідності редагувати файли теми [37]. В даному випадку буде використаний для додавання коду збору аналітики Google Analytics.
- When Last Login — цей плагін дозволяє відстежувати останній час входу користувачів на сайт [38]. Корисний для проектів коли сайт буде адмініструвати і наповнювати команда з декількох людей.

На етапі налаштування теми та плагінів WordPress відбувається перетворення абстрактних ідей та макетів у функціональний веб-сайт. Саме тут реалізується візуальний стиль, структура та функціональні можливості сайту.

Після встановлення обраної теми Blocksy, необхідно провести її початкову конфігурацію. Це включає в себе налаштування кольорової гами, шрифтів, розмірів елементів та інших візуальних аспектів. Більшість сучасних тем, зокрема Blocksy, пропонують широкий спектр налаштувань, які дозволяють створити унікальний дизайн без значних зусиль. Крім того налаштовується header, footer сайту та наявності sidebar відповідно до створених на попередніх етапах макетів. Важливо звертати увагу на адаптивність сайту на різних пристрою і переглянути всі варіанти (наприклад, вигляд для комп'ютера, планшета та для смартфона) візуального відображення сайту. За потреби підкоригувати налаштування до тих пір поки не буде досягнута відповідність вимогам замовника.

На етапі **наповнення сайту** WordPress контент, який був раніше підготовлений, інтегрується в готову структуру. Цей процес передбачає не лише розміщення тексту та зображень, але й забезпечення логічної структури, зручності навігації та відповідності загальному стилю сайту.

Першим кроком є визначення місця для кожного елемента кожної сторінки в межах створених макетів. Текстовий контент розміщується в відповідних блоках, зображення встановлюються у визначені області, а мультимедійні файли інтегруються за допомогою спеціальних модулів. Важливо дотримуватися раніше розробленої структури сайту, щоб користувачі могли легко знайти потрібну інформацію.

Перед розміщенням зображень на сайті їх необхідно оптимізувати. Це дозволить зменшити розмір файлів, прискорити завантаження сторінок та покращити загальну продуктивність сайту. Останні версії WordPress зазвичай мають автоматичне створення зображень різного розширення. Однак за потреби можна виконати додаткову ручну обробку або скористатися додатковими плагінами для цього.

Для забезпечення зручної навігації по сайту необхідно створити логічну структуру сторінок та записів. Це передбачає використання категорій, тегів та вкладених сторінок. Кожен елемент контенту повинен бути віднесений до відповідної категорії або тегу, що дозволить користувачам легко знайти потрібну інформацію.

При наповненні сайту контентом необхідно враховувати принципи SEO-оптимізації. Це означає використання ключових слів, створення унікальних заголовків та описів, а також оптимізацію мета-тегів. Ці заходи допоможуть підвищити видимість сайту в пошукових системах.

Для створення логічної структури сайту та покращення користувацького досвіду необхідно використовувати внутрішні посилання. Внутрішні посилання дозволяють користувачам легко переходити з однієї сторінки на іншу.

Після завершення наповнення сайту контентом необхідно ретельно перевірити весь матеріал на наявність помилок, орфографічних помилок та неточностей. Також слід перевірити, чи правильно відображаються всі елементи на різних пристроях та в різних браузерах.

Наповнення сайту контентом – це не одноразовий процес. Для підтримки інтересу користувачів необхідно регулярно оновлювати контент, додавати нові матеріали та актуалізувати старі.

Наповнення сайту WordPress контентом є важливим етапом у створенні функціонального та інформативного веб-ресурсу. Використання візуального конструктора Elementor значно спрощує цей процес, дозволяючи реалізувати будь-які, навіть найскладніші дизайнерські рішення. Однак, для досягнення успіху необхідно дотримуватися певних правил, таких як логічна структура, SEO-оптимізація та регулярне оновлення контенту.

Етап **тестування** є критично важливим у процесі створення сайту на платформі WordPress. Після того, як сайт наповнений контентом і налаштований, необхідно ретельно перевірити його на наявність помилок, неточностей та відповідність поставленим завданням.

Тестування WordPress-сайту передбачає комплекс заходів, спрямованих на оцінку його функціональності, зручності використання, сумісності з різними браузерами та пристроями, а також відповідності вимогам пошукових систем.

Функціональне тестування полягає у перевірці кожної функції сайту на правильність виконання. Це включає в себе тестування форм, кнопок, пошуку, галерей та інших інтерактивних елементів. Особливу увагу слід приділити перевірці роботи плагінів, які були встановлені для розширення функціональності сайту.

Тестування юзабіліті спрямоване на оцінку зручності використання сайту для користувачів. Важливо перевірити інтуїтивність навігації, логічність структури, читабельність тексту та загальне враження від користування сайтом. Для цього можна залучити бета-тестерів або провести власне тестування.

Тестування на різних пристроях та браузерах є необхідним для забезпечення адаптивності сайту [39]. Сайт повинен коректно відображатися на різних розмірах екранів, від мобільних телефонів до великих моніторів. Також важливо перевірити сумісність сайту з популярними браузерами (Chrome, Firefox, Safari, Edge тощо).

Тестування швидкості завантаження сторінок є важливим аспектом, оскільки користувачі очікують швидкого доступу до інформації. Для цього можна використовувати спеціальні інструменти, які дозволяють виміряти швидкість завантаження сторінок та виявити вузькі місця.

SEO-тестування спрямоване на перевірку відповідності сайту вимогам пошукових систем. Необхідно перевірити наявність мета-тегів, заголовків, описів, а також внутрішню та зовнішню посилання.

Тестування безпеки дозволяє виявити потенційні вразливості сайту, які можуть бути використані зловмисниками. Рекомендується проводити регулярні аудити безпеки для захисту сайту від хакерських атак.

Тестування на стресостійкість дозволяє оцінити, як сайт працює під великим навантаженням. Це особливо актуально для сайтів з високою відвідуваністю.

Етап тестування є невід'ємною частиною процесу створення сайту на WordPress. Ретельне тестування дозволяє виявити та усунути помилки, забезпечити високу якість сайту та задоволеність користувачів. Важливо пам'ятати, що тестування – це постійний процес, який необхідно проводити не тільки після завершення розробки сайту, але й регулярно протягом його життя.

Етап **запуску сайту** WordPress є кульмінацією процесу його створення. Після завершення всіх попередніх етапів, таких як розробка дизайну, наповнення контентом та тестування, сайт готовий до публікації в мережі Інтернет. Запуск сайту включає в себе комплекс заходів, спрямованих на забезпечення безперебійної роботи ресурсу та його доступності для користувачів.

Перевірка всіх налаштувань перед запуском є обов'язковою процедурою. Необхідно повторно переконатися в тому, що всі плагіни налаштовані належним чином, форми працюють коректно, а посилання ведуть на правильні сторінки. Особливу увагу слід приділити налаштуванням SEO, щоб сайт був добре індексований пошуковими системами.

Створення резервної копії сайту є важливим заходом, який дозволить відновити всі дані у випадку виникнення будь-яких проблем. Резервні копії слід створювати регулярно і зберігати їх у безпечному місці.

Перевірка доступності сайту після публікації є обов'язковою процедурою. Необхідно переконатися в тому, що сайт коректно відображається у всіх популярних браузерах і на різних пристроях. Також слід перевірити швидкість завантаження сторінок і відсутність помилок.

Реєстрація сайту в пошукових системах є важливим кроком для підвищення його видимості в результатах пошуку. Реєстрація сайту в Google Search Console [40] та інших інструментах пошукових систем дозволить відстежувати індексацію сайту, виявляти помилки та отримувати додаткову інформацію про трафік.

Налаштування аналітики дозволить відстежувати ефективність сайту, аналізувати поведінку користувачів та вносити необхідні зміни. Найпопулярнішими інструментами аналітики є Google Analytics [41] та Meta Pixel [42]. Для роботи аналітик необхідно із кабінетів відповідних сервісів скопіювати коди відстеження (Додаток И) та додати їх в код сторінок сайту використовуючи функціонал плагіну WPCode Lite.

Просування сайту є невід'ємною частиною запуску. Після публікації сайту необхідно розробити стратегію його просування в мережі Інтернет. Це може включати в себе SEO-оптимізацію, контекстну рекламу, соціальні мережі та інші маркетингові інструменти.

Запуск сайту WordPress – це складний процес, який вимагає уважності до деталей і системного підходу. Правильно проведений запуск забезпечить успішний розвиток сайту та досягнення поставлених цілей.

Етап **подальшої підтримки сайту WordPress** є не менш важливим, ніж його створення та запуск. Це безперервний процес, спрямований на підтримання сайту в актуальному стані, забезпечення його безпеки, а також на підвищення ефективності та досягнення поставлених цілей.

Регулярне оновлення ядра WordPress, тем та плагінів є одним з ключових аспектів підтримки. Оновлення забезпечують усунення вразливостей, покращення функціональності та додавання нових можливостей. Важливо стежити за виходом нових версій і своєчасно їх встановлювати.

Моніторинг роботи сайту дозволяє виявляти та усувати проблеми, які можуть виникнути в процесі експлуатації. Це включає в себе відстеження доступності сайту, швидкості завантаження сторінок, помилок у роботі, а також аналіз поведінки користувачів.

Аналіз статистики дозволяє оцінити ефективність сайту та внести необхідні зміни. За допомогою аналітичних інструментів можна відстежувати кількість відвідувачів, джерела трафіку, поведінку користувачів на сайті та інші важливі показники.

Розширення функціональності сайту може бути необхідним для досягнення нових цілей. Це може включати в себе встановлення нових плагінів, розробку додаткових модулів або інтеграцію з іншими системами.

Підтримка користувачів є важливою складовою успішного сайту. Необхідно забезпечити зручну систему зворотного зв'язку з користувачами, оперативно відповідати на їхні питання та вирішувати проблеми.

В підсумку, подальша підтримка сайту WordPress є безперервним процесом, який вимагає систематичного підходу та певних знань. Регулярний моніторинг, оновлення, оптимізація та розвиток сайту дозволять забезпечити його довготривалу ефективність та досягнення поставлених цілей.

### **3.4. Керівництво користувача**

Розробка вебсайту на платформі WordPress передбачає багатоетапне тестування, яке дозволяє виявити можливі помилки та забезпечити стабільну роботу системи в умовах реального використання. Основними методами тестування в цьому процесі були функціональне тестування, тестування продуктивності, юзабіліті-тестування, а також тестування безпеки та сумісності.

Функціональне тестування було ключовим етапом, що передбачав перевірку коректної роботи всіх компонентів сайту. Основна увага приділялася перевірці роботи основних функцій системи управління контентом (CMS), таких як створення, редагування, видалення сторінок та публікацій, робота з медіафайлами, налаштування меню навігації та взаємодія з плагінами. Кожен з етапів функціонального тестування виконувався вручну за допомогою тестових сценаріїв, які моделювали дії користувача на сайті. Наприклад, тестувалася можливість реєстрації нового користувача, авторизація, редагування профілю, а також взаємодія з формами зворотного зв'язку.



Особливу увагу приділялося тестуванню обробки виключних ситуацій, таких як введення некоректних даних у форми або доступ до сторінок, які не існують. Система була налаштована таким чином, щоб обробляти ці ситуації без порушення роботи сайту, забезпечуючи належний рівень користувацького досвіду. Наприклад, при введенні некоректного електронного листа у форму реєстрації система видавала відповідне повідомлення про помилку, зберігаючи при цьому всі інші введені дані.

Тестування продуктивності передбачало перевірку здатності сайту обробляти значний обсяг запитів без суттєвого зниження швидкості завантаження сторінок або падіння сервера. Для цього використовувалися такі інструменти, як Google PageSpeed Insights [43] (Додаток К) та GTmetrix [44], які дозволяють оцінити швидкість завантаження сайту під час роботи з різними типами пристроїв і браузерів. Було виявлено, що оптимізація зображень, налаштування кешування та мінімізація CSS і JavaScript значно покращують продуктивність сайту.

Юзабіліті-тестування спрямовувалося на перевірку зручності використання сайту кінцевими користувачами. Було проведено декілька раундів тестування з залученням тестових користувачів, які оцінювали зрозумілість навігації, доступність контенту та загальну зручність взаємодії з системою. Зворотній зв'язок від користувачів дозволив виявити та усунути недоліки в дизайні інтерфейсу, такі як нечітка структура меню або занадто складний процес заповнення форм.

Тестування безпеки включало перевірку системи на наявність вразливостей, які можуть бути використані для здійснення кібератак. Основну увагу приділяли захисту від SQL-ін'єкцій [45], міжсайтового скриптингу (XSS) [46], а також забезпеченню безпеки авторизаційних даних. Для цього використовувалися плагіни безпеки, такі як Wordfence [47] та iThemes Security [48], які дозволяють відстежувати спроби несанкціонованого доступу та здійснювати регулярне сканування сайту на предмет вразливостей.

Тестування сумісності передбачало перевірку коректної роботи сайту на різних платформах, браузерах та пристроях. Сайт було оптимізовано для роботи в останніх версіях браузерів Google Chrome, Mozilla Firefox, Microsoft Edge, а також Safari. Окрім того, було проведено перевірку коректності відображення контенту на мобільних

пристроях з різними розмірами екранів, що забезпечило адаптивність сайту та зручність його використання на смартфонах та планшетах.

Після завершення процесу тестування було проведено демонстрацію роботи вебсайту на платформі WordPress, яка охоплювала перевірку основних функцій системи у реальних умовах.

Перше тестове завдання полягало у створенні нової статті через адміністративну панель WordPress. Користувач авторизувався в системі, перейшов до розділу «Додати нову публікацію», ввів заголовок, текстовий контент, додав зображення та опублікував статтю. У результаті стаття коректно відобразилася на головній сторінці сайту, а всі додані елементи були належним чином форматовані та адаптовані до різних типів пристроїв.

У другому тестовому завданні було перевірено функцію обробки виключних ситуацій. Користувач спробував ввести некоректні дані у форму зворотного зв'язку, зокрема текст замість адреси електронної пошти. Після натискання кнопки «Надіслати» система видала повідомлення про помилку з поясненням, що введені дані є некоректними. При цьому всі інші поля форми зберегли введену інформацію, що дозволило користувачеві швидко виправити помилку без необхідності повторного заповнення форми.

Третє тестове завдання стосувалося тестування продуктивності сайту під високим навантаженням. За допомогою спеціального інструмента імітувалося одночасне підключення до сайту сотень користувачів, які переглядали сторінки, завантажували медіафайли та взаємодіяли з формами. Попри значне навантаження, сайт залишався доступним, а час завантаження сторінок не перевищував трьох секунд.

Також було продемонстровано систему захисту сайту від несанкціонованого доступу. Користувач спробував увійти до адміністративної панелі з використанням невірного пароля декілька разів поспіль. Після трьох невдалих спроб доступ до панелі було заблоковано на певний період часу, про що користувач отримав відповідне сповіщення. Ця функція дозволяє захистити сайт від атак типу «перебір паролів» (Brute Force Attack) [49].

Завершальне тестове завдання полягало у перевірці коректності відображення сайту на різних типах пристроїв. Сайт було відкрито на настільному комп'ютері, ноутбучі, планшеті та смартфоні. У всіх випадках контент відображався коректно, зберігалася логіка навігації, а елементи інтерфейсу автоматично адаптувалися до розміру екрана.

Таким чином, проведені етапи тестування та демонстрація роботи сайту підтвердили його готовність до використання в реальних умовах. Завдяки всебічному тестуванню вдалося забезпечити стабільну роботу системи, високу продуктивність, зручність використання, а також належний рівень безпеки та сумісності з різними пристроями та браузерами.

### **Висновки до розділу 3**

Було перелічено основні засоби розробки необхідні для реалізації проекту. Серед них виокремлено ядро WordPress, теми, плагіни, мови програмування PHP та JavaScript та інструменти веб розробки.

Сформовано вимоги до технічного та програмного забезпечення проекту. Вказано мінімально необхідні вимоги до продуктивності процесорів серверу, мінімальні обсяги оперативної пам'яті, місця на диску та мінімальну пропускну здатність комунікаційного каналу. Окреслено також програмні вимоги до типу і версій серверу, бази даних та PHP. Зазначено додаткові вимоги для підтримки і роботи сайту.

Описано поетапно процес реалізації проекту.

## ВИСНОВКИ

Під час виконання магістерської роботи було описано предметне середовище розробки сайту, яке складається із клієнтської (реалізується за допомогою HTML, CSS та JavaScript), серверної частини системи (PHP та файлова система), а також бази даних. Зазначено що розробку можна здійснювати як на локальному, так і на віддалених середовищах. Розглянуто успішні сценарії сайтів розроблених на WordPress CMS та наведено перелік сайтів брендів та організацій світового масштабу з аудиторією більше 1 млн. користувачів в кожного. Поставлено завдання розробити функціональний сайт, який відповідає всім стандартам.

Крім того було здійснено детальний аналіз предметної області, який передбачав визначення вхідних та вихідних даних. Вхідну інформацію, яка необхідна для розробки сайту, можна виокремити в технічні, змістовні, бізнес-вимоги, аналіз цільової аудиторії, контентні, дизайнерські та правові вимоги. Вихідні дані, які є результатом виконання проекту, мають відповідати вхідним вимогам та мати вигляд функціонального вебсайту з чітко визначеною структурою, інтерфейсом користувача та адміністратора, набором функцій та інтеграцій. Визначено основні кроки до реалізації мінімально життєздатного продукту. Важливими кроками виокремлено визначення мети проекту, аналіз ринку та конкурентів, визначення основних функціональних вимог, створення ієрархії сторінок, встановлення системи, вибір і встановлення теми, плагінів, розробка навігації сайту, створення та наповнення сторінок, тестування і запуск проекту. Висвітлено методології які можна використати для розробки сайту. Зазначено, що для індивідуальної роботи над проектом підійде підхід Waterfall, однак для команди краще застосувати методи Scrum, Kanban або їх комбінацію – Scrumban. Згадано що комбінований метод Scrumban найкраще підходить для збереження чіткого планування та розділення на етапи і забезпечення візуального контролю моніторингу завдань.

Також було перелічено основні засоби розробки необхідні для реалізації проекту. Найважливішими є ядро WordPress, теми, плагіни, мови програмування PHP та JavaScript, а також інструменти веб розробки. Сформовано вимоги до технічного та програмного забезпечення проекту, які включають мінімально необхідну

продуктивність сервера, обсяги оперативної пам'яті та дискового простору, пропускну здатність, типи веб-серверів, які в переліку підтримуваних системою, типи бази даних, версію php та ін. Описано процес реалізації проекту. Для успішної розробки необхідно проаналізувати потенційні цілі і завдання та на їх основі сформулювати вимоги до проекту (технічне завдання). На основі технічного завдання розробляється концепція, структура сайту та дизайн сайту після чого здійснюється технічна підготовка середовища роботи сайту. Коли технічно все налаштовано можна встановлювати на здійснювати перше налаштування системи, вибір необхідних за функціоналом теми та плагінів. Далі йде етап наповнення сайту контентом та подальше тестування. Якщо все добре сайт можна запускати. Подальша підтримка сайту є також важливим етапом необхідним для успішної майбутньої роботи сайту.

Беручи до уваги все вище зазначене — завдання та мета дослідження були виконані в повному обсязі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Usage Statistics and Market Share of WordPress, November 2024. *W3Techs - extensive and reliable web technology surveys*. URL: <https://w3techs.com/technologies/details/cm-wordpress>.
2. W3Schools.com. *W3Schools Online Web Tutorials*. URL: <https://www.w3schools.com/html/>.
3. W3Schools.com. *W3Schools Online Web Tutorials*. URL: <https://www.w3schools.com/css/>.
4. JavaScript | MDN. *MDN Web Docs*. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
5. PHP: Introduction - Manual. *PHP: Hypertext Preprocessor*. URL: <https://www.php.net/manual/en/introduction.php>.
6. MySQL 8.4 Reference Manual :: 1.2.1 What is MySQL?. *MySQL*. URL: <https://dev.mysql.com/doc/refman/8.4/en/what-is-mysql.html>.
7. REST API Handbook | Developer.WordPress.org. *WordPress Developer Resources*. URL: <https://developer.wordpress.org/rest-api/>.
8. Microsoft. Visual Studio Code - Code Editing. Redefined. *Visual Studio Code - Code Editing. Redefined*. URL: <https://code.visualstudio.com/>.
9. JetBrains. PhpStorm: The PHP IDE by JetBrains. *JetBrains*. URL: <https://www.jetbrains.com/phpstorm/>.
10. Sublime Text - the sophisticated text editor for code, markup and prose. *Sublime Text - Text Editing, Done Right*. URL: <https://www.sublimetext.com/>.
11. Notepad++. *Notepad++*. URL: <https://notepad-plus-plus.org/>.
12. About. *WordPress.org*. URL: <https://wordpress.org/about/>.
13. Minasyan A. 31 Famous WordPress Websites for Inspiration in 2024. *10Web - Build & Host Your WordPress Website*. URL: <https://10web.io/blog/wordpress-websites/>.
14. The best websites, built with WordPress. | WordPress.org. *WordPress.org*. URL: <https://wordpress.org/showcase/>.

15. Minimum Viable Product (MVP): What is it & Why it Matters. *Atlassian*. URL: <https://www.atlassian.com/agile/product-management/minimum-viable-product>.
16. Waterfall Methodology for Project Management | Atlassian. *Atlassian*. URL: <https://www.atlassian.com/agile/project-management/waterfall-methodology>.
17. What is Scrum?. *Scrum.org*. URL: <https://www.scrum.org/resources/what-scrum-module>.
18. The Official Guide to The Kanban Method | Kanban University. *Kanban University Home | Kanban University*. URL: <https://kanban.university/kanban-guide/>.
19. "Top Project Management Methods for Software Development - Waterfall, Agile, Scrum, Kanban - Blog of the Project Management System Worksection". *Worksection*. URL: <https://worksection.com/en/blog/software-development-methodologies.html>.
20. Scrumban: The Best of Two Agile Methodologies [2024] • Asana. *Asana*. URL: <https://asana.com/resources/scrumban>.
21. Component Pages – Make WordPress Core. *Make WordPress Core*. URL: <https://make.wordpress.org/core/components/>.
22. WordPress Server Hosting Requirements. *Clio Websites*. URL: <https://cliowebsites.com/wordpress-server-hosting-requirements/>.
23. Server Environment. *Make WordPress Hosting*. URL: <https://make.wordpress.org/hosting/handbook/server-environment/>.
24. Requirements. *WordPress.org*. URL: <https://wordpress.org/about/requirements/>.
25. Responsive design - Learn web development | MDN. *MDN Web Docs*. URL: [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Responsive\\_Design](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design).
26. What is an SSL/TLS Certificate? - SSL.com. *SSL.com*. URL: <https://www.ssl.com/article/what-is-an-ssl-tls-certificate/>.
27. Що таке ftp client і як ним користуватися?. *ukraine.com.ua*. URL: [https://www.ukraine.com.ua/uk/blog/hosting\\_ukraine/что-такое-ftp-client-i-kak-im-polzovatsya.html](https://www.ukraine.com.ua/uk/blog/hosting_ukraine/что-такое-ftp-client-i-kak-im-polzovatsya.html).

28. How to install WordPress – Advanced Administration Handbook | Developer.WordPress.org. *WordPress Developer Resources*. URL: <https://developer.wordpress.org/advanced-administration/before-install/howto-install/>.
29. How to install the full Blocksy package - Blocksy. *Blocksy*. URL: <https://creativethemes.com/blocksy/docs/general/install-blocksy-theme/>.
30. What Is Elementor For WordPress? » Elementor. *Elementor*. URL: <https://elementor.com/help/wordpress-and-elementor-2/>.
31. Cyr-To-Lat. *WordPress.org*. URL: <https://wordpress.org/plugins/cyr2lat/>.
32. Polylang. *WordPress.org*. URL: <https://wordpress.org/plugins/polylang/>.
33. Contact Form 7. *WordPress.org*. URL: <https://wordpress.org/plugins/contact-form-7/>.
34. Flamingo. *WordPress.org*. URL: <https://wordpress.org/plugins/flamingo/>.
35. Open Graph Protocol Framework. *WordPress.org*. URL: <https://wordpress.org/plugins/open-graph-protocol-framework/>.
36. Post Views Counter. *WordPress.org*. URL: <https://wordpress.org/plugins/post-views-counter/>.
37. WPCode – Insert Headers and Footers + Custom Code Snippets – WordPress Code Manager. *WordPress.org*. URL: <https://wordpress.org/plugins/insert-headers-and-footers/>.
38. When Last Login. *WordPress.org*. URL: <https://wordpress.org/plugins/when-last-login/>.
39. 6 Best WordPress Responsive Testing Tools - Qode Interactive. *Qode Magazine*. URL: <https://qodeinteractive.com/magazine/best-wordpress-responsive-testing-tools/>.
40. How To Use Search Console | Google Search Central | Documentation | Google for Developers. *Google for Developers*. URL: <https://developers.google.com/search/docs/monitor-debug/search-console-start>.
41. Introduction to Google Analytics | Google for Developers. *Google for Developers*. URL: <https://developers.google.com/analytics/devguides/collection/ga4>.



42. Meta-Pixel - Dokumentation - Meta for Developers. *Social Technologies | Meta for Developers*. URL: <https://developers.facebook.com/docs/meta-pixel/>.
43. About PageSpeed Insights | Google for Developers. *Google for Developers*. URL: <https://developers.google.com/speed/docs/insights/v5/about>.
44. Features. *GTmetrix*. URL: <https://gtmetrix.com/features.html>.
45. W3Schools.com. *W3Schools Online Web Tutorials*. URL: [https://www.w3schools.com/sql/sql\\_injection.asp](https://www.w3schools.com/sql/sql_injection.asp).
46. Cross Site Scripting (XSS) | OWASP Foundation. *OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation*. URL: <https://owasp.org/www-community/attacks/xss/>.
47. Wordfence Security – Firewall, Malware Scan, and Login Security. *WordPress.org Україна*. URL: <https://uk.wordpress.org/plugins/wordfence/>.
48. Solid Security – Password, Two Factor Authentication, and Brute Force Protection. *WordPress.org*. URL: <https://wordpress.org/plugins/better-wp-security/>.
49. Brute Force Attacks – Advanced Administration Handbook | Developer.WordPress.org. *WordPress Developer Resources*. URL: <https://developer.wordpress.org/advanced-administration/security/brute-force/>.

## ДОДАТКИ

## ДОДАТОК А

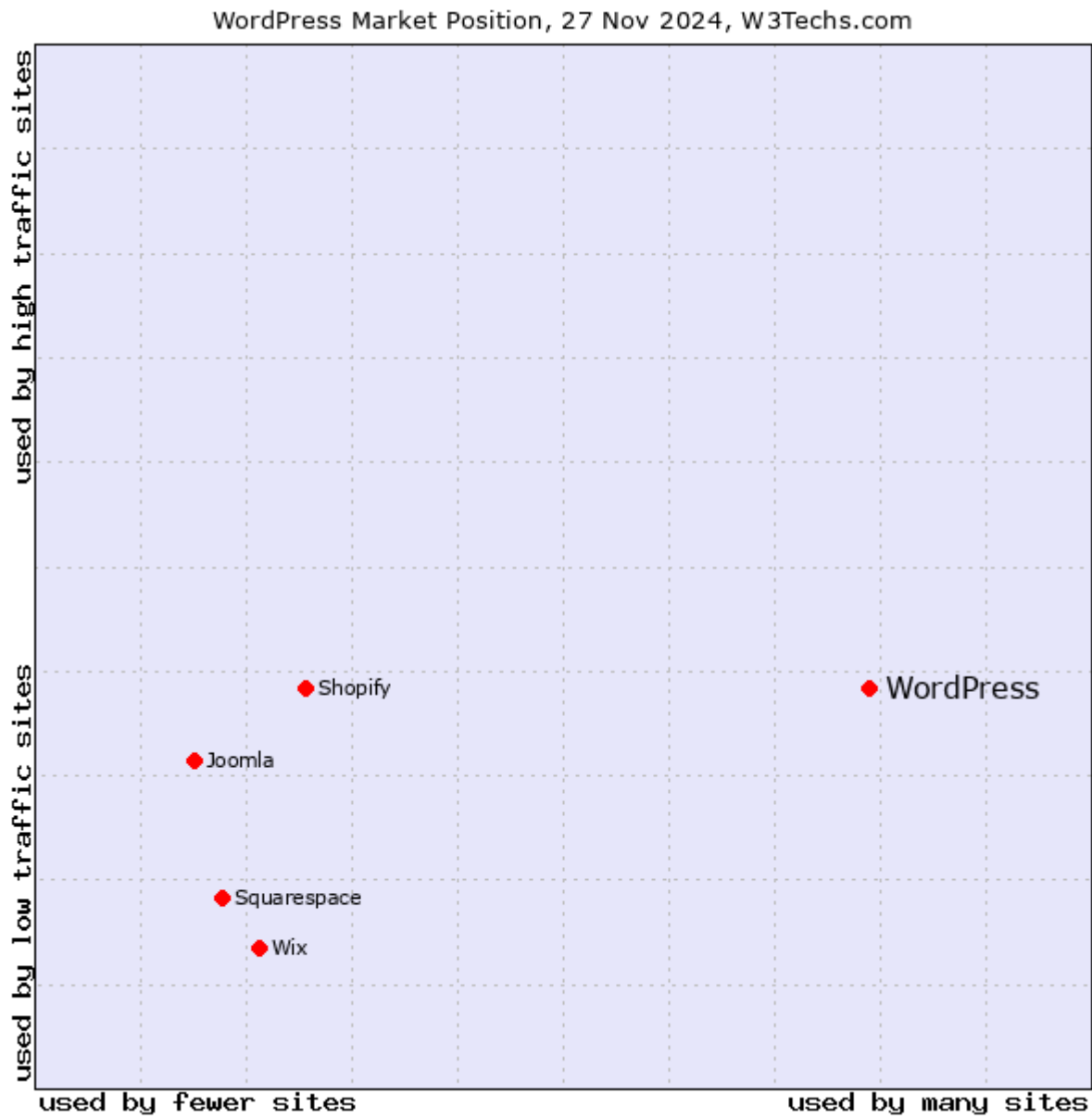


Рис. 1.1. Позиція WordPress на ринку систем управління контентом (CMS) [1]



Рис. 1.2. Скріншот сайту The Walt Disney Company



Рис. 1.3. Скріншот сайту The Rolling Stones

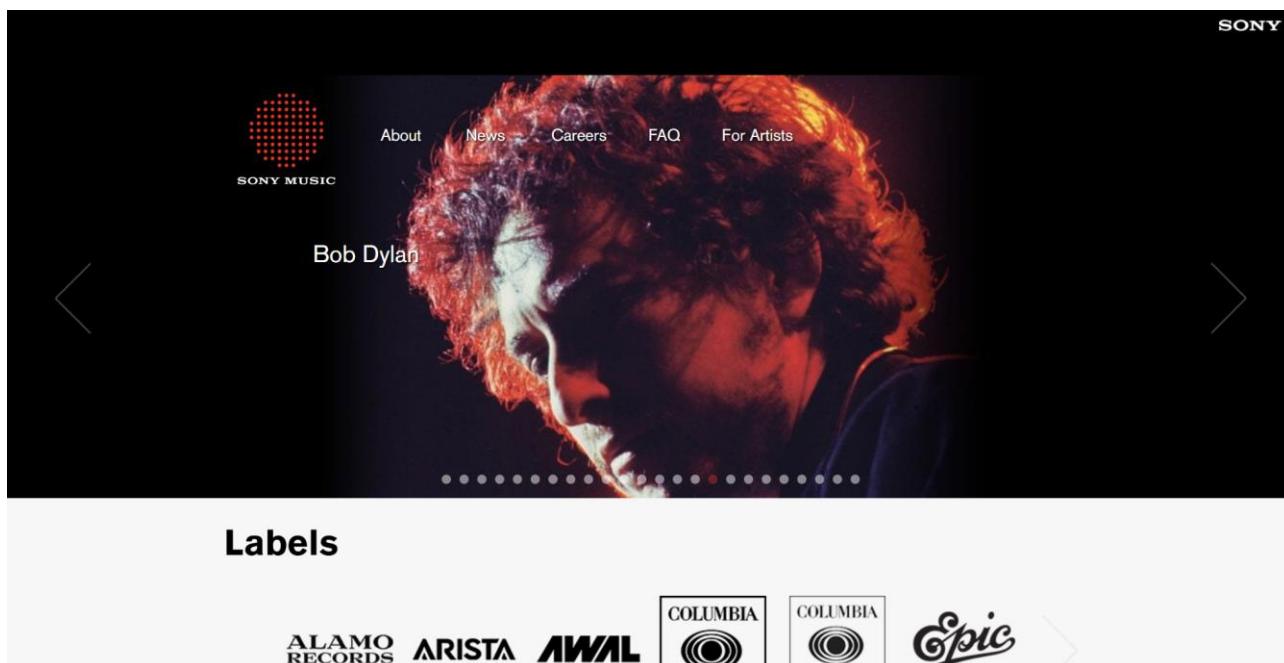


Рис. 1.4. Скріншот сайту Sony Music

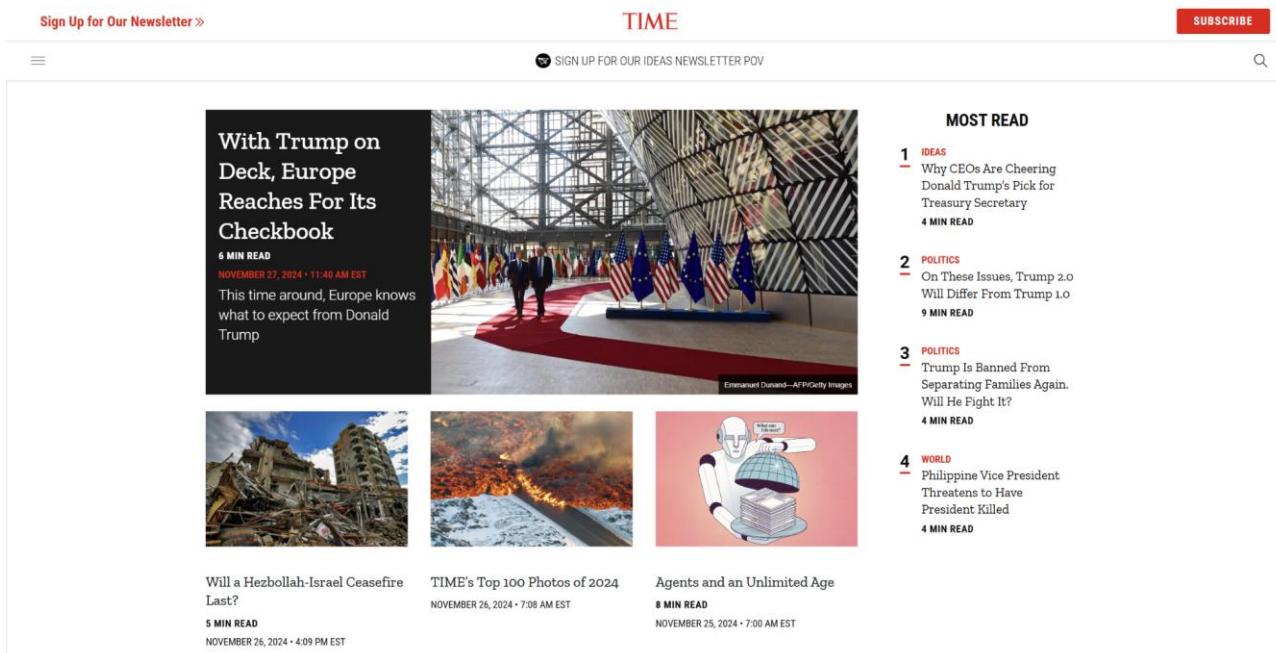


Рис. 1.5. Скріншот сайту Time



Рис. 1.6. Скріншот сайту The New York Times

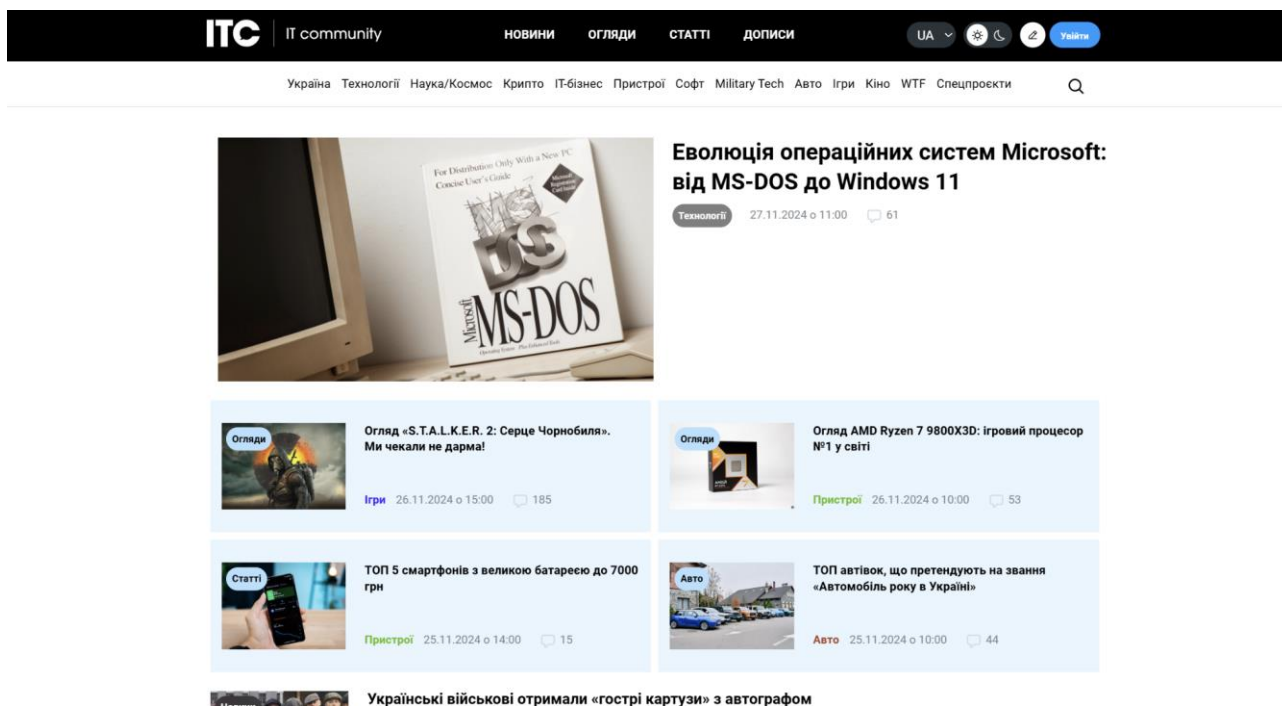


Рис. 1.7. Скріншот сайту ITC.ua

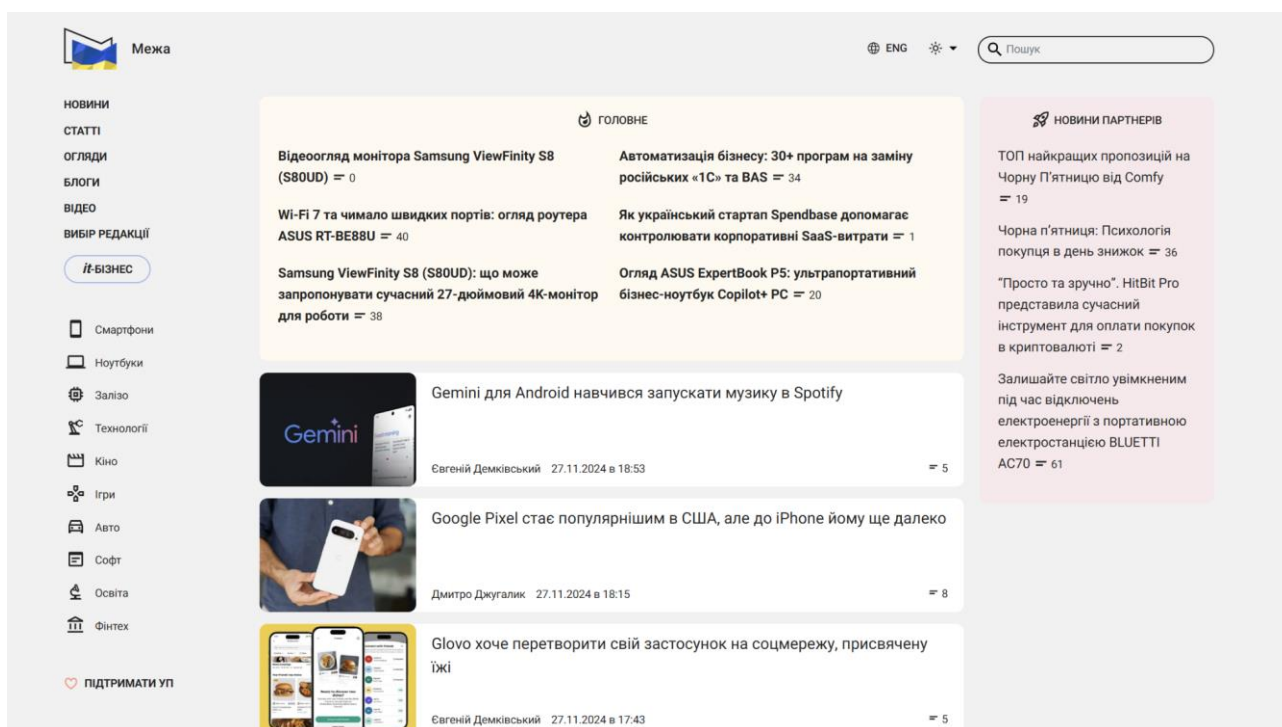


Рис. 1.8. Скріншот сайту Mezha.Media



Рис. 1.9. Скріншот сайту The White House

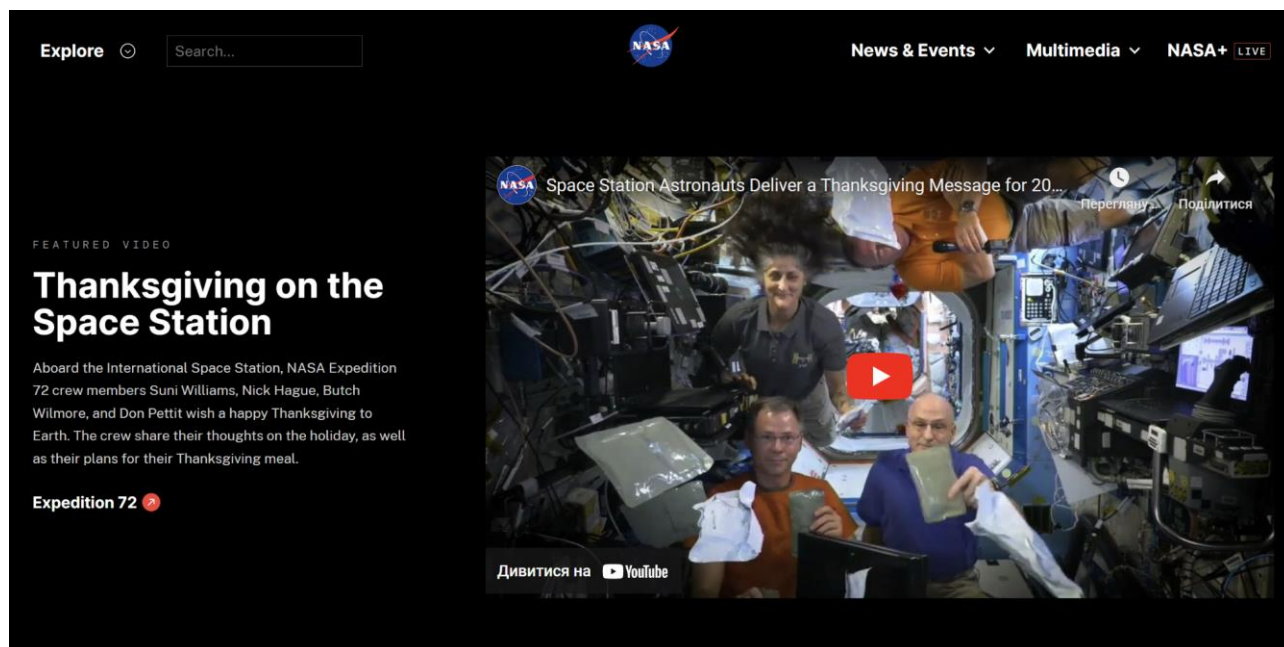


Рис. 1.9. Скріншот сайту NASA

Таблиця 2.1. Поверхнєве порівняння методик Scrum, Kanban та Scrumban  
(<https://www.atlassian.com/agile/project-management/scrumban>)

	<b>Scrum</b>	<b>Kanban</b>	<b>Scrumban</b>
Методологія	Спринт фіксованої довжини Фіксовані ролі Послідовна реалізація	Обмежений обсяг одночасних завдань Відстеження завдань візуально Безперервний робочий процес	Спринти з фіксованою довжиною Обмежений обсяг одночасних завдань Відстеження завдань візуально Безперервний робочий процес
Ролі	Власник продукту Команда розробки Scrum Майстер	Жодної	Жодної
Події	Планування спринту Щоденний мітинг Огляд спринту Ретроспектива спринту	Канбан-зустріч	Планування спринту Щоденний мітинг Ретроспектива спринту
Перебіг подій в проєкті	Беклог продукту Беклог спринту У процесі виконання Перевірка Виконано	До виконання У процесі Виконано	До виконання У процесі Виконано





Нижче ви маєте ввести свої деталі підключення до бази даних. Якщо ви не впевнені, зв'яжіться зі своїм хостинг-провайдером.

**Назва бази даних**   
Назва бази даних, яку ви хочете використовувати з WordPress.

**Ім'я користувача**   
Ваше ім'я користувача бази даних.

**Пароль**  [Показати](#)  
Ваш пароль бази даних.

**Хост бази даних**   
You should be able to get this info from your web host, if localhost does not work.

**Табличний префікс**   
Якщо ви хочете мати одну базу даних для кількох інсталяцій WordPress, змініть це.

Рис. 3.1. Інтерфейс першопочаткового налаштування WordPress

**Плагіни** [Додати новий плагін](#)

Всі (11) | Активні (9) | Неактивні (2) | Автоматичні оновлення відключені (11)

Search installed plugins

Групові дії  11 елементів

<input type="checkbox"/>	Плагін	Опис	Автоматичні оновлення
<input type="checkbox"/>	<b>Akismet Anti-spam: Захист від спаму</b> <a href="#">Activate</a>   <a href="#">Видалити</a>	Akismet, яким користуються мільйони людей, є чи не найкращим у світі способом захистити свій блог від спаму. Антиспам Akismet захищає ваш сайт, навіть коли ви спите. Для початку активуйте плагін Akismet, а потім перейдіть на сторінку налаштувань Akismet, щоб налаштувати ключ API. Версія 5.3.5   Автор Automatic - Anti-spam Team   <a href="#">Переглянути деталі</a>	<a href="#">Увімкнути автоматичні оновлення</a>
<input type="checkbox"/>	<b>Contact Form 7</b> <a href="#">Налаштування</a>   <a href="#">Вимкнути</a>	Конструктор контактних форм. Простий, але гнучкий. Версія 6.0.1   Автор Takayuki Miyoshi   <a href="#">Переглянути деталі</a>	<a href="#">Увімкнути автоматичні оновлення</a>
<input type="checkbox"/>	<b>Cur-To-Lat</b> <a href="#">Налаштування</a>   <a href="#">Вимкнути</a>	Перетворює не-латинські символи в посиланнях записів, сторінок і категорій в латинські. Корисно для створення людinizрозумлимих URL. Засновано на оригінальному плагіні від Антона Скоробогатова. Версія 6.2.3   Автор Сергій Бірюков, Михайло Кобзаров, Igor Гергель   <a href="#">Переглянути деталі</a>	<a href="#">Увімкнути автоматичні оновлення</a>
<input type="checkbox"/>	<b>Elementor</b> <a href="#">Налаштування</a>   <a href="#">Вимкнути</a>   <b>Оновлення зі збіркою зараз!</b>	У конструкторі сайтів Elementor є все: візуальний конструктор сторінок, піксельно досконалий дизайн, адаптивне редагування для мобільних пристроїв і багато іншого. Почніть зараз! Версія 3.25.10   Автор Elementor.com   <a href="#">Переглянути деталі</a>   <a href="#">Документація і часті питання</a>   <a href="#">Відео інструкції</a>	<a href="#">Увімкнути автоматичні оновлення</a>
<input type="checkbox"/>	<b>Flamingo</b> <a href="#">Вимкнути</a>	Надійний плагін для зберігання повідомлень Contact Form 7. Версія 2.5   Автор Takayuki Miyoshi   <a href="#">Переглянути деталі</a>	<a href="#">Увімкнути автоматичні оновлення</a>
<input type="checkbox"/>	<b>Hello Dolly</b> <a href="#">Activate</a>   <a href="#">Видалити</a>	Це не просто плагін, він символізує надію та ентузіазм цілого покоління, що виражені в двох словах з пісні, яка є найбільш відомою у виконанні Луїса Армстронга: Hello, Dolly. Після активації плагіну на кожній сторінці панелі управління в правому верхньому куті буде відображатися випадковий рядок з пісні Hello, Dolly. Версія 1.7.2   Автор Matt Mullenweg   <a href="#">Переглянути деталі</a>	<a href="#">Увімкнути автоматичні оновлення</a>

Рис. 3.2. Перелік встановлених плагінів WordPress

<input type="checkbox"/>	<b>Hello Dolly</b> <a href="#">Activate</a>   <a href="#">Видалити</a>	Це не просто плагін, він символізує надію та ентузіазм цілого покоління, що виражені в двох словах з пісні, яка є найбільш відомою у виконанні Луїса Армстронга: Hello, Dolly. Після активації плагіну на кожній сторінці панелі управління в правому верхньому куті буде відображатися випадковий рядок з пісні Hello, Dolly. Версія 1.7.2   Автор Matt Mullenweg   <a href="#">Переглянути деталі</a>	<a href="#">Увімкнути автоматичні оновлення</a>
<input type="checkbox"/>	<b>Open Graph Protocol Framework</b> <a href="#">Вимкнути</a>   <a href="#">Documentation</a>   <a href="#">Shop</a>	The Open Graph Protocol enables any web page to become a rich object in a social graph. If you would like to support our work, please visit our <a href="#">shop</a> . Версія 1.9.0   Автор itthink   <a href="#">Переглянути деталі</a>	<a href="#">Увімкнути автоматичні оновлення</a>
<input type="checkbox"/>	<b>Polylang</b> <a href="#">Налаштування</a>   <a href="#">Вимкнути</a>	Додає багатомовні властивості до WordPress Версія 3.6.5   Автор WP SYNTEX   <a href="#">Переглянути деталі</a>	<a href="#">Увімкнути автоматичні оновлення</a>
<input type="checkbox"/>	<b>Post Views Counter</b> <a href="#">Settings</a>   <a href="#">Вимкнути</a>	Post Views Counter allows you to display how many times a post, page or custom post type had been viewed in a simple, fast and reliable way. Версія 1.4.8   Автор dFactory   <a href="#">Переглянути деталі</a>	<a href="#">Увімкнути автоматичні оновлення</a>
<input type="checkbox"/>	<b>When Last Login</b> <a href="#">Settings</a>   <a href="#">Вимкнути</a>	See when a user logs into your WordPress site. Версія 1.2.2   Автор Yoofoo Plugins   <a href="#">Переглянути деталі</a>   <a href="#">Settings</a>   <a href="#">Docs</a>   <a href="#">Support</a>	<a href="#">Увімкнути автоматичні оновлення</a>
<input type="checkbox"/>	<b>WPCode Lite</b> <a href="#">Get WPCode Pro</a>   <a href="#">Code Snippets</a>   <a href="#">Вимкнути</a>	Easily add code snippets in WordPress. Insert scripts to the header and footer, add PHP code snippets with conditional logic, insert ads pixel, custom content, and more. Версія 2.2.3.1   Автор WPCode   <a href="#">Переглянути деталі</a>	<a href="#">Увімкнути автоматичні оновлення</a>
<input type="checkbox"/>	Плагін	Опис	Автоматичні оновлення

Групові дії  11 елементів

Дякуємо за творчість з WordPress. Версія 6.7.1

Рис. 3.3. Перелік встановлених плагінів WordPress (продовження)

## Приклад коду відстеження Google Analytics

```

<!-- Google tag (gtag.js) -->
<script async src="https://www.googletagmanager.com/gtag/js?id=G-
XXXXXXXXXX"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());

  gtag('config', 'G-XXXXXXXXXXXX');
</script>

```

## Приклад коду відстеження Meta Pixel

```

<!-- Meta Pixel Code -->
<script>
!function(f,b,e,v,n,t,s)
{if(f.fbq)return;n=f.fbq=function(){n.callMethod?
n.callMethod.apply(n,arguments):n.queue.push(arguments)};
if(!f._fbq)f._fbq=n;n.push=n;n.loaded=!0;n.version='2.0';
n.queue=[];t=b.createElement(e);t.async=!0;
t.src=v;s=b.getElementsByTagName(e)[0];
s.parentNode.insertBefore(t,s)}(window, document,'script',
'https://connect.facebook.net/en_US/fbevents.js');
fbq('init', 'XXXXXXXXXXXXXXXXXX');
fbq('track', 'PageView');
</script>
<noscript></noscript>
<!-- End Meta Pixel Code -->

```

Report from Nov 27, 2024, 3:35:09 AM

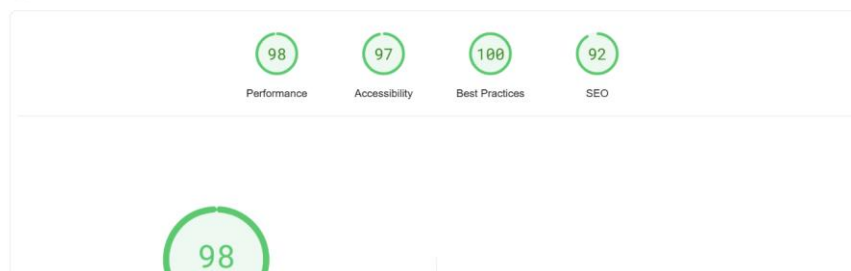
 [Analyze](#) Mobile  Desktop Discover what your real users are experiencing No Data Diagnose performance issues

Рис 3.4. Приклад звіту Google PageSpeed Insights