

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка веб-додатку музею "Трипільська культура" з
вбудованим каталогом»

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 Комп'ютерні науки

(код, найменування спеціальності)

освітньо-професійної програми Комп'ютерні науки

(назва)

*Кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання на відповідне
джерело*

(підпис)

Михайло МОРОЗ
(Ім'я, ПРИЗВИЩЕ здобувача)

Виконав:
здобувач вищої освіти
група КНД-42

Михайло МОРОЗ

Керівник:
науковий ступінь,
вчене звання

Ірина ЦЕРБИНА
к.т.н. доцент

Рецензент:
науковий ступінь,
вчене звання

(Ім'я, ПРИЗВИЩЕ)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерних наук

Ступінь вищої освіти Бакалавр

Спеціальність Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри Комп'ютерних наук

_____ Віктор ВИШНІВСЬКИЙ

«_____» _____ 2024 р

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Морозу Михайлу Олексійовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Розробка веб-додатку музею "Трипільська культура" з вбудованим каталогом.

керівник кваліфікаційної роботи Ірина ЩЕРБИНА, к.т.н., доцент,

(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)

затверджені наказом Державного університету

інформаційно-комунікаційних технологій від «27» 02.2024р. №36

2. Строк подання кваліфікаційної роботи «31» травня 2024р
3. Вихідні дані до кваліфікаційної роботи: вимоги замовника до ВЕБ-додатку музею, документація до програмного забезпечення.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Постановка задачі та вибір технічних засобів для реалізації проекту
Розробка веб додатку музею “Трипільська Культура” з CMS Wordpress
Налаштування SEO оптимізації для веб-додатку музею “Трипільська
Культура”

5. Перелік графічного матеріалу: презентація
- 1 Вимоги замовника на розробку ВЕБ додатку
 - 2 Дослідження та вибір технології
 - 3 Порівняння тем WordPress для мистецьких галерей та музеїв
 - 4 Вибір технології
 - 5 розробка веб-додатку
 - 6 Карта сайту веб-додатку музею “Трипільська Культура”
 - 7 Сторінки веб-додатку
 - 8 Використані плагіни
 - 9 Налаштування SEO оптимізації
 - 10 Оптимізація сайту під пошукові системи
6. Дата видачі завдання **«23» лютого 2024 р.**

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі та аналіз вимог	24.02-29.02.24	Виконано
2	Дослідження та вибір технологій	01.03-15.03.24	Виконано
3	Вибір та налаштування теми	16.03-31.03.24	Виконано
4	Розробка UI/UX дизайну	01.04-15.04.24	Виконано
5	Розробка функціоналу сторінок	16.04-30.04.24	Виконано
6	Розробка каталогу артефактів	01.05-14.05.24	Виконано
7	Оптимізація зображень та SEO налаштування	15.05-22.05.24	Виконано
8	Завершальне тестування та запуск	23.05-30.05.24	Виконано

Здобувач вищої освіти

(підпис)

Михайло МОРОЗ

(Ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи

(підпис)

Ірина ЩЕРБИНА

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 90 стор., 1 табл., 14 рис., 19 джерел.

Мета роботи – побудова веб-додатку музею “Трипільська культура” з вбудованим каталогом під потреби клієнта

Об’єкт дослідження – Технології побудови веб додатків .

Предмет дослідження – Веб додаток музею .

Короткий зміст роботи: Було поставлено задачу, та досліджено технології для розробки веб-додатку музею “Трипільська Культура”. Досліджено функціональні можливості CMS Wordpress. Вибрано та досліджено тему CMS Wordpress для веб-додатку

Створено веб-додаток з використанням CMS Wordpress. Розроблено UI/UX дизайн за допомогою Figma. Розроблено функціонал сторінок веб-додатку, та каталог артефактів

Налаштовано SEO оптимізацію для веб-додатку музею “Трипільська Культура”

КЛЮЧОВІ СЛОВА: ЦИФРОВІЗАЦІЯ, ВЕБ-ДОДАТОК, SEO ОПТИМІЗАЦІЯ, ФРОНТЕНД, БЕКЕНД, UI/UX ДИЗАЙН, ТЕХНІЧНА РЕАЛІЗАЦІЯ

ABSTRACT

The text part of the qualification work for the master's degree: 90 pages, 1 table, 14 figures, 19 sources.

Research task - Research and implementation of optimal technologies and methods for creating an effective web application of the Trypillia Culture Museum.

Purpose - to develop a web application of the museum with a built-in catalog for the needs of the client.

Object of research - CMS Wordpress.

Subject of research - Web application of the museum .

Summary of the work: The task was set and the technologies for developing the web application of the Trypillia Culture Museum were investigated. The functionality of CMS Wordpress was investigated. The CMS Wordpress theme for the web application was selected and researched.

A web application was created using CMS Wordpress. Developed UI/UX design using Figma. Developed the functionality of the web application pages and the catalog of artifacts

SEO optimization for the web application of the Trypillia Culture Museum has been set up

KEYWORDS: DIGITALIZATION, WEB APPLICATION, SEO OPTIMIZATION, FRONTEND, BACKEND, UI/UX DESIGN, TECHNICAL IMPLEMENTATION

ЗМІСТ

ВСТУП.....	9
1 ПОСТАНОВКА ЗАДАЧІ ТА ВИБІР ТЕХНІЧНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОЕКТУ.....	11
1.1 Постановка задачі.....	11
1.2 Дослідження та вибір технології для розробки веб-додатку музею “Трипільська Культура”.....	12
1.3 Дослідження функціональних можливостей Wordpress.....	28
1.4 Вибір теми CMS Wordpress для веб-додатку музею “Трипільська Культура”... 33	
1.5 Дослідження можливостей теми Ozeum.....	37
2 РОЗРОБКА ВЕБ ДОДАТКУ МУЗЕЮ “ТРИПІЛЬСЬКА КУЛЬТУРА” З CMS WORDPRESS.....	42
2.1 Розробка UI/UX дизайну веб-додатку музею “Трипільська Культура” за допомогою Figma.....	40
2.2 Розробка функціоналу сторінок веб-додатку музею “Трипільська Культура”.. 49	
2.3 Налаштування теми та головних плагінів.....	53
2.4 Робота з зображеннями.....	62
2.5 Розробка каталогу для веб-додатку музею “Трипільська Культура”.....	63
3 НАЛАШТУВАННЯ SEO ОПТИМІЗАЦІЇ ДЛЯ ВЕБ-ДОДАТКУ МУЗЕЮ “ТРИПІЛЬСЬКА КУЛЬТУРА”.....	67
3.1 Призначення та основні методи SEO оптимізації.....	67
3.2 Налаштування SEO оптимізації для веб-додатку.....	70
ВИСНОВКИ.....	74
ПЕРЕЛІК ПОСИЛАНЬ.....	76
ДОДАТОК А.....	78
ПРЕЗЕНТАЦІЯ.....	80

ВСТУП

Проект розробки веб-додатку музею "Трипільська Культура" з вбудованим каталогом спрямований на створення сучасного, зручного та функціонального онлайн-ресурсу, який відповідатиме всім потребам музею. Основна мета цього проекту – забезпечити ефективне представлення артефактів музею в Інтернеті, сприяти популяризації трипільської культури та забезпечити зручний доступ до інформації для відвідувачів. При постановці задачі було враховано обмежені ресурси замовника для підтримки сайту та недостатній технічний досвід для роботи з front-end фреймворками. Тому було вирішено використати систему керування вмістом (CMS) WordPress, яка забезпечує легкість у користуванні, мінімальні витрати на обслуговування, а також багатий функціонал для створення та підтримки веб-сайтів. Конструктори сайтів були відкинуті через їх обмежений функціонал.

Проведене дослідження функціональних можливостей WordPress підтвердило доцільність вибору цієї платформи для реалізації проекту. В результаті аналізу різних тем WordPress було обрано тему Ozeum, яка ідеально підходить для музеїв та виставкових центрів завдяки своєму сучасному дизайну та широкому набору функцій. Ця тема включає професійний дизайн, можливість одноразового встановлення демо-контенту, підтримку плагінів Elementor Page Builder, Revolution Slider, Essential Grid та багато інших корисних інструментів. Розробка UI/UX дизайну веб-додатку була здійснена за допомогою Figma, що дозволило створити естетично привабливий та зручний у використанні інтерфейс. Далі, було розроблено функціонал сторінок веб-додатку, налаштовано тему та головні плагіни для забезпечення необхідних функцій. Особливу увагу було приділено роботі з зображеннями, для чого було використано плагін Optimole, що забезпечує швидке завантаження сторінок навіть при великій кількості високоякісних зображень. Для демонстрації артефактів музею було створено каталог на базі WooCommerce, який виконує функцію вітрини без можливості покупки. З метою забезпечення високої видимості сайту у пошукових системах було налаштовано SEO оптимізацію за допомогою плагінів Yoast SEO, Virtual

Robots.txt та Breadcrumb NavXT. Ці інструменти допомогли налаштувати метатеги, XML-карти та покращити структуру сайту для пошукових систем. Проект створення веб-додатку музею "Трипільська Культура" з вбудованим каталогом успішно реалізовано завдяки ретельному дослідженню, вибору відповідних технологій та налаштуванню необхідних інструментів. Використання CMS WordPress, теми Ozeum, а також ряду плагінів дозволило створити сучасний, функціональний та зручний у використанні веб-додаток, який сприятиме популяризації трипільської культури та забезпечить ефективне представлення артефактів музею в онлайн-середовищі.

1 ПОСТАНОВКА ЗАДАЧІ ТА ВИБІР ТЕХНІЧНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОЕКТУ

1.1 Постановка задачі

Передо мною стояла задача створення інтерактивного веб-додатку для музею “Трипільська Культура”. Це державний музей, до зібрана інформація про найдавніших мешканців нашої країни - трипільців. Він презентує найбільші за площами поселення світового енеоліту та є єдиним в Європі Заповідником однієї археологічної культури. До його складу входить 11 поселень-гігантів, найбільше з яких (для додержавного періоду людства) – поселення «Тальянки», мало площу 450 гектарів.

Оскільки фінансування державних музеїв в нашій країні недостатнє, в музею немає фахівців в галузі програмування - головною задачею було побудувати додаток, користування яким не вимагало від замовника знань в області ІТ технологій, з метою оцифрування експонатів та збереження унікальних артефактів української історії та культури. При цьому додаток мав бути достатньо потужним та відповідати сучасним вимогам до аналогічних інтернет ресурсів. Додаток має задовольняти потреби замовника, зокрема, мати можливість адміністрування та змін сторінок після закінчення проекту. Додаток не має потребувати значних витрат під час розробки та після її закінчення. Однією з головних вимог було створення вбудованого каталогу артефактів з можливістю змін та додаванням нових елементів.

Замовником було визначено основні функції та можливості, які повинен надавати веб-додаток, що стало основою для розробки функціональних вимог.

- Каталог артефактів:
 - Система каталогізації з можливістю фільтрації за епохами, типами артефактів тощо.

- Детальна інформація про кожен артефакт, включаючи фотографії, опис, історію виявлення.
- Події та експозиції:
 - Афіша музейних подій, лекцій, відкриттів та тимчасових експозицій.
- Адаптивний дизайн:
 - Забезпечення коректного відображення на різних пристроях (комп'ютерах, планшетах, смартфонах).
- Безпека:
 - Захист від зловживань та несанкціонованого доступу до інформації.
- Адміністрування:
 - Додавання та редагування контенту замовником після здачі проекту

Ця постановка задачі має на меті створення веб-додатку, який забезпечує користувачам цікавий та пізнавальний досвід ознайомлення з музеєм Трипільської культури, через взаємодію з каталогом артефактів та освітленням найближчих подій.

Мною було проаналізовані методи створення веб-додатків, з метою визначити яка технологія буде найбільш доцільною для використання у цьому проекті.

1.2 Дослідження та вибір технології для розробки веб-додатку музею “Трипільська Культура”

Існує багато технологій розробки веб-додатків, і кожна має свої переваги та недоліки. Вибір конкретної технології залежить від специфічних потреб, області застосування та особистих уподобань замовника. Технології розробки веб-додатків охоплюють різноманітні підходи та інструменти. Основні категорії включають конструктори, CMS та повноцінну розробку з використанням фронтенд та бекенд фреймворків.

Конструктори призначені для легкого та швидкого створення веб-додатків без програмування. Вони прості у використанні та швидкості розгортання, але мають обмежену гнучкість та менше можливостей для розширення.

CMS (Content Management System) використовуються для управління та редагування вмісту веб-додатку без глибоких знань програмування. Вони легкі для використання, мають розширені можливості порівняно з конструкторами, але потребують підтримки та оновлень.

Повноцінна розробка з використанням фронтенд та бекенд фреймворків означає створення веб-додатків з розширеними функціональностями та високою гнучкістю. Такий підхід передбачає використання фронтенд фреймворків, таких як React, Angular, Vue.js, для розробки інтерфейсу та бекенд фреймворків, таких як Django, Flask, Express, Laravel, для розробки серверної частини веб-додатків. На рисунку 1.1 представлено три методи розробки веб-додатків які я розглядав у своєму дослідженні.[1]

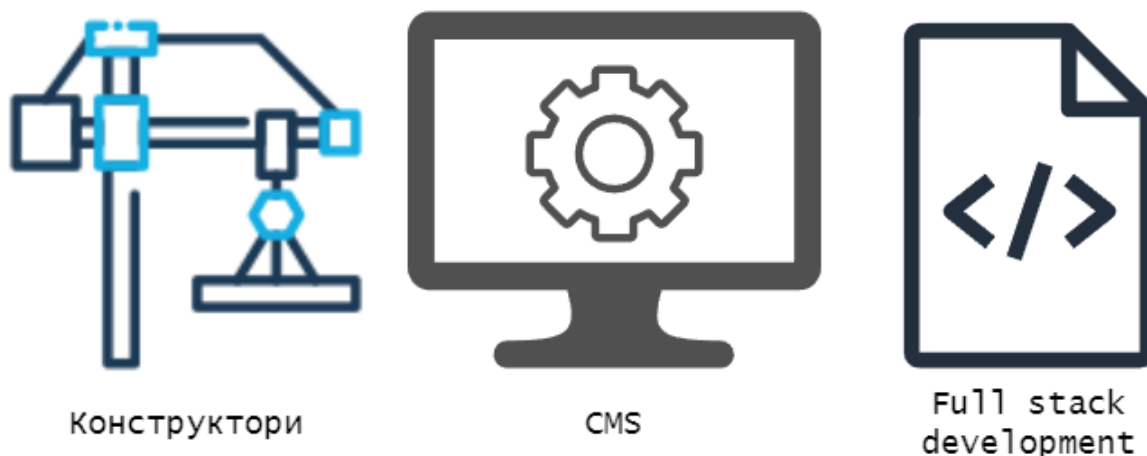


Рисунок 1.1 - Методи розробки веб-додатків

Конструктори веб-додатків - це інструменти, які дозволяють користувачам створювати веб-додатки за допомогою графічного інтерфейсу та елементів перетягування та викидання. Вони призначені для людей без глибоких технічних навичок, щоб спростити процес створення веб-присутності.

Процес роботи конструкторів веб-додатків починається з вибору теми або шаблону. Користувач обирає візуальний шаблон або тему, яка відповідає його потребам та стилю. Додавання контенту відбувається за допомогою блоків або елементів, які можна просто перетягувати та розміщувати на сторінці. Кожен елемент піддається налаштуванню через зміну його розміру, кольорів, шрифтів та інших параметрів. Після завершення роботи користувач публікує свій сайт та робить його доступним для відвідування. В цілях дослідження конструкторів веб-додатків я ознайомився з кількома поширеними конструкторами.

Wix - це один з популярних конструкторів веб-додатків, який надає можливість користувачам створювати веб-додатки без глибоких технічних навичок. Він пропонує інтуїтивний графічний інтерфейс, що дозволяє користувачам додавати та налаштовувати елементи, такі як текстові блоки, зображення, кнопки та інші за допомогою простого перетягування та викидання. Wix також пропонує вбудовану функціональність для створення інтернет-магазинів з можливістю приймати платежі онлайн.

Хоча Wix є потужним конструктором веб-додатків і надає ряд інструментів для створення різноманітних веб-проектів, існують певні функції, які можуть бути важкими або неможливими для реалізації на цій платформі. Якщо потрібно додаткову функціональність, яка не передбачена готовими елементами Wix, важко впровадити власний JavaScript-код. Для великих інтернет-магазинів із складними опціями продукту та обробкою платежів можуть знадобитися більш потужні рішення. Через численні обмеження, Wix може бути проблемним для великих інформаційних порталів та складних додатків, які вимагають роботи з великим обсягом даних. В підсумку Wix чудово підходить для осіб, які шукають простий та ефективний спосіб створити веб-додаток, особливо для особистих портфоліо, невеликих бізнесів та інтернет-магазинів. На рисунку 1.2 представлено панель управління конструктора Wix

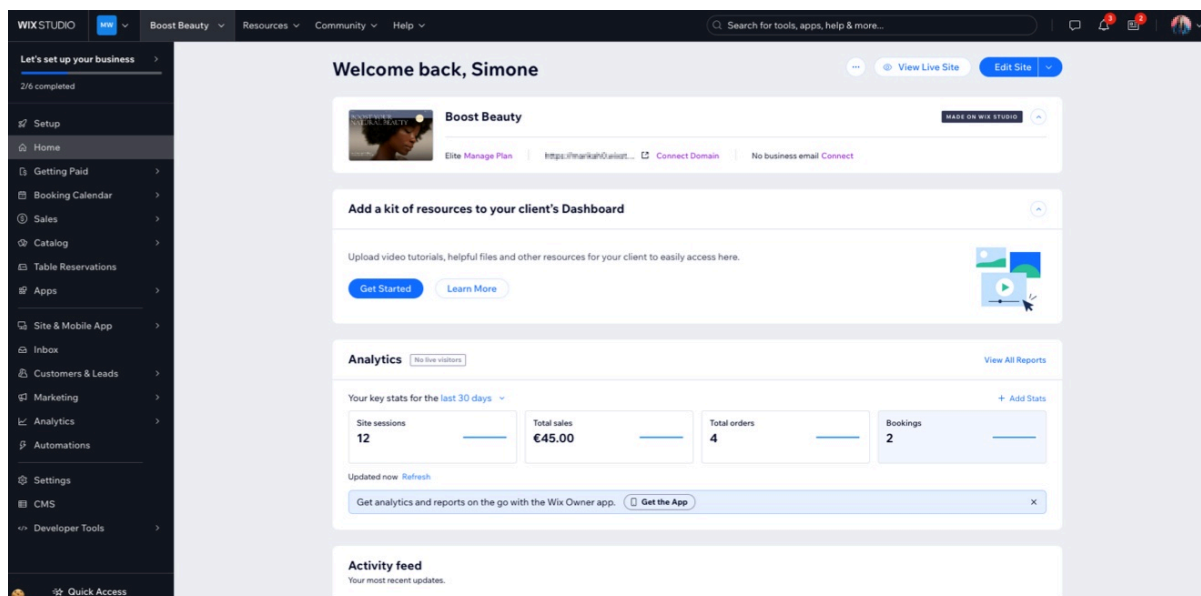


Рисунок 1.2 - Панель управління Wix

Weebly - це інший конструктор веб-додатків, який надає можливість користувачам створювати веб-додатки за допомогою простого та інтуїтивного інтерфейсу. Він популярний серед тих, хто шукає простий спосіб створення веб-присутності.[11]

Weebly має інтуїтивний графічний інтерфейс а також пропонує широкий вибір готових тем і шаблонів для різних галузей та видів веб-додатків. Цей конструктор ставить акцент на легкість використання та інтуїтивність для того, щоб користувачі будь-якого рівня могли легко створювати свої веб-додатки, що призводить до обмеження деяких більш технічних аспектів. Оскільки складні функції можуть ускладнити інтерфейс та процес редагування веб-додатків, Weebly старається зберегти простоту та зрозумілість інтерфейсу для кінцевого користувача.

Weebly підходить для тих, хто шукає простий та швидкий спосіб створити веб-додаток. Особливо для невеликих бізнесів, особистих портфоліо та блогів. Squarespace - це інтегрована платформа для створення та редагування веб-додатків, що надає зручний інтерфейс та широкий спектр функцій для користувачів. Squarespace пропонує стильні та сучасні шаблони, які можна легко налаштувати під свої потреби. Шаблони розроблені так, щоб бути відзначеними

своєю естетикою та відповідати сучасним дизайн-стандартам. Інтерфейс редактора дозволяє користувачам легко перетягувати та розміщувати різні елементи на сторінці, забезпечуючи гнучкість у створенні макетів, а вбудований інструмент для створення інтернет-магазину, дозволяє легко додавати продукти, керувати запасами та обробляти платежі. Платформа також надає інтегровані інструменти аналітики, які дозволяють користувачу відстежувати відвідуваність веб-додатку та отримувати звіти. [12]

Хоча Squarespace надає багато функцій, він може бути менш гнучким для деяких складних проектів порівняно зі скриптовими мовами або CMS. Squarespace є відмінним вибором для тих, хто шукає естетичний та легкий в розумінні інструмент для створення веб-додатків.

У випадках, коли потрібно швидко запустити простий веб-додаток або веб-додаток без великих функціональних вимог та при обмеженому бюджеті, конструктори можуть бути ефективним рішенням. Однак для більш складних та унікальних проектів, де важлива гнучкість та розширення функціональності, може бути доцільніше використовувати повноцінну розробку з використанням фронтенд та бекенд фреймворків.[13]

Повноцінна розробка(Full stack development) передбачає створення веб-додатків з використанням різних інструментів та технологій для обробки як клієнтської, так і серверної частини. Такий підхід є найскладнішим у порівнянні з конструкторами та CMS, оскільки вимагає більшої кваліфікації розробників та значні витрати ресурсів та часу.

З метою дослідження, я вирішив звернути увагу на використання фронтенд та бекенд фреймворків для створення користувацького інтерфейсу та серверної частини відповідно. На рисунку 1.3 представлено частини розробки веб-додатків при повноцінній розробці.[3]

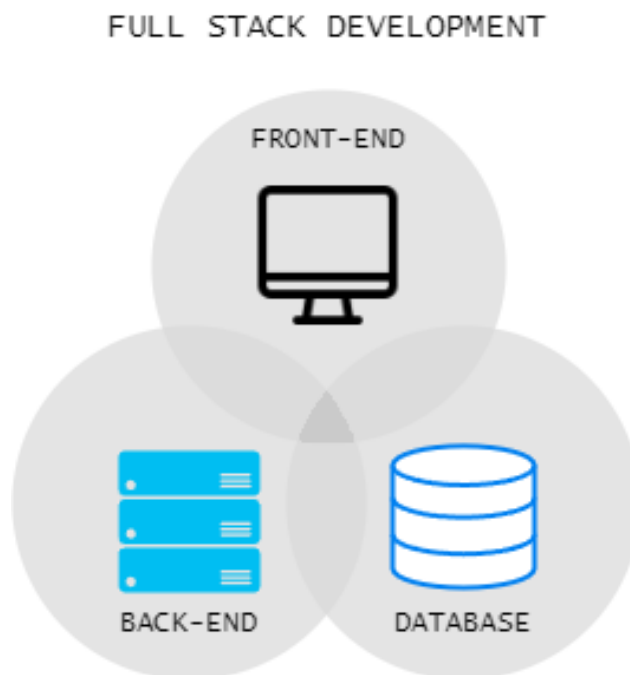


Рисунок 1.3 - Частина розробки WEB додатків

Фронтенд фреймворки - це колекції підготовлених та передбачених заздалегідь інструментів, бібліотек, технологій і стандартів програмування, які призначені для полегшення розробки користувацького інтерфейсу веб-додатків. Ці фреймворки допомагають розробникам зосередитися на створенні функціональності та дизайні, замість того, щоб витратити час на рутинні задачі, такі як маніпуляції з DOM (Document Object Model), обробка подій, анімації тощо. Створення веб-додатків на фронт-енд фреймворках та бібліотеках є важливою частиною розробки веб-технологій. Фронт-енд охоплює все, що стосується взаємодії користувача з веб-додатком або додатком.

React - це бібліотека JavaScript для розробки інтерфейсів користувача. Основна ідея React полягає в тому, щоб створювати веб-сторінки або веб-додатки, які мають динамічний інтерфейс та реагують на зміни даних без перезавантаження сторінки. В React все є компонентом. Компонент може бути маленьким елементом, таким як кнопка чи поле вводу, або складним веб-елементом, наприклад, сторінка або віджет. Компоненти можуть мати власний стан (state), який визначає, як вони повинні виглядати в певний момент часу, і можуть змінюватися відповідно до взаємодії користувача або зміни зовнішніх

даних. Для роботи компонент React використовує JSX (JavaScript XML), розширення синтаксису JavaScript. Основна ідея JSX полягає в тому, щоб зробити розробку інтерфейсу користувача в React більш інтуїтивно зрозумілою та продуктивною. Це робить процес створення компонентів більш зручним та швидким, допомагаючи розробникам створювати потужні та ефективні веб-додатки. Воно дозволяє вписувати HTML-подібний код безпосередньо в JavaScript. Він використовується в React для опису структури користувацького інтерфейсу. JSX дозволяє використовувати звичний синтаксис HTML для створення компонентів React. Наприклад, можна використовувати теги `<div>`, `<span>`, `<p>` та інші теги HTML. Для більш гнучкого налаштування компонент React надає можливість використовувати функції, змінні та інші вирази JavaScript всередині JSX, обгортавши їх в фігурні дужки `{ }`.

React-компоненти використовують пропси(props) для спілкування один з одним. Кожен батьківський компонент може передавати певну інформацію своїм дочірнім компонентам, надаючи їм пропси. Пропси можуть нагадувати атрибути HTML, але їх можна передавати через них будь-які значення JavaScript, включаючи об'єкти, масиви та функції. Пропси - це інформація, яка передається у JSX тег, до прикладу, `className`, `src`, `alt`, `width` і `height` - деякі з пропсів, які можна передати в тег `<img>`. Пропси у React є незамінними, тобто компонент не може змінювати свої пропси. Вони є тільки для читання. Якщо потрібно змінити дані, компонент може використовувати стан (state) або звертатися до батьківського компонента за оновленням пропсів. Пропси роблять компоненти React більш гнучкими та повторно використовуваними, дозволяючи легко передавати дані між компонентами та конфігурувати їх зовні. Вони є важливою частиною розробки в React і допомагають створювати масштабовані та динамічні веб-додатки. [8]

Кожен компонент React має життєвий цикл, який складається з різних фаз, таких як монтаж, оновлення та розмонтаж. Життєвий цикл компоненту в React визначає послідовність подій та методів, які відбуваються під час створення, оновлення та видалення компонента. Це дозволяє розробникам втручатися у різні

моменти життєвого циклу компонента та виконувати певні дії, такі як ініціалізація стану, відправка запитів до сервера, підписка на події та очищення ресурсів.

Angular - це популярний фронт-енд фреймворк, розроблений Google, який використовується для побудови веб-додатків та односторінкових додатків (SPA), побудована на TypeScript. Як платформа, Angular включає в себе компонентний фреймворк для створення масштабованих веб-додатків, набір добре інтегрованих бібліотек, які охоплюють широкий спектр функцій, включаючи маршрутизацію, управління формами, клієнт-серверну взаємодію та набір інструментів для розробників, які допоможуть розробляти, збирати, тестувати та оновлювати код. Angular надає розробнику перевагами платформи, яка може масштабуватися від проектів для одного розробника до додатків корпоративного рівня.

Angular використовує модель-вид-контролер (MVC) архітектуру, що дозволяє розділити логіку додатку на модель (дані), вид (шаблон) та контролер (логіка). Основною одиницею побудови додатків в Angular є компоненти. Кожен компонент включає в себе логіку, шаблон та стилі, які визначають вигляд та поведінку цього компонента. Angular використовує HTML-подібний синтаксис для створення шаблонів компонентів. Крім того, він надає директиви, які дозволяють додавати додаткову функціональність до HTML елементів. Angular забезпечує двостороннє зв'язування даних між шаблонами та моделями даних, що дозволяє автоматично оновлювати вміст сторінки при зміні даних. Angular використовує механізм ін'єкції залежностей для управління залежностями між компонентами. Це дозволяє створювати більш масштабовані та легко тестовані додатки. Angular має вбудований механізм маршрутизації, який дозволяє переходити між різними сторінками та компонентами в додатку без перезавантаження сторінки, та надає зручні інструменти для автоматизованого тестування додатків, включаючи одиничні тести та e2e (end-to-end) тести.

Сервіси в Angular - це класи, які надають певну функціональність та можуть бути використані в усій програмі. Вони дозволяють розділити код програми на логічні частини, що полегшує управління та підтримку коду. Сервіси дозволяють розділити логіку програми на окремі блоки, що сприяє збереженню коду більш

читабельним та організованим. Оскільки сервіси можуть бути використані в усій програмі, вони надають можливість використовувати одну й ту ж функціональність в різних компонентах, директивах або інших сервісах. Сервіси можна використовувати в різних проектах або навіть в різних частинах програми, що сприяє підтримці та розвитку програмного забезпечення. Багато сервісів використовуються для керування станом додатку, таких як зберігання даних, кешування, автентифікація та інше. Сервіси можуть бути інтегровані з іншими сторонніми бібліотеками або фреймворками для надання додаткової функціональності або розширення можливостей програми. Сервіси легко тестувати окремо від інших частин програми, що дозволяє забезпечити надійність та стабільність програми.[9]

Vue.js - це прогресивний фронтенд-фреймворк для створення користувацьких інтерфейсів. Він став популярним завдяки своїй легкості використання, гнучкості та високій продуктивності. Vue використовує реактивний підхід, де зміни в стані автоматично відображаються в інтерфейсі користувача. Це полегшує взаємодію з даними та відображенням. Він побудований на компонентній архітектурі, що дозволяє створювати перевикористовувані та легко управляемі компоненти. Vue надає двостороннє зв'язування даних між станом та інтерфейсом, дозволяючи автоматично оновлювати дані в обидві сторони. Vue використовує директиви для зручного маніпулювання DOM та властивостями компонентів. Використання модульної структури дозволяє розділяти додаток на невеликі та легко управляемі частини. У Vue використовується розширений синтаксис шаблонів з підтримкою різних функцій, таких як умовні оператори, цикли, фільтри та інші, що полегшує створення складних інтерфейсів. У Vue.js директиви використовуються для надання певної функціональності до DOM елементів або компонентів. Існує ряд вбудованих директив, таких як `v-if`, `v-for`, `v-on`, `v-bind`, `v-model`, які використовуються для управління відображенням, обробки подій, зв'язування даних та іншого.

У Vue можна створювати власні директиви для виконання специфічних завдань. Це може бути директива, яка змінює вигляд елементів, додає додаткову

функціональність або виконує певну логіку, яку потім можна використовувати в шаблоні. Директиви у Vue мають власний життєвий цикл, який включає в себе такі хуки, як `bind`, `inserted`, `update`, `componentUpdated`, `unbind`, що дозволяє виконувати певні дії на різних етапах життєвого циклу елемента, до якого застосована директива. Деякі вбудовані директиви можуть приймати аргументи або модифікатори, які змінюють їх поведінку.

Загалом, використання фронтенд фреймворків є важливим для створення сучасних та ефективних веб-додатків, які задовольняють вимоги користувачів та бізнесу. Вони дозволяють швидко розробляти, масштабувати та підтримувати веб-додатки, забезпечуючи зручну та ефективну розробку.[10]

Бекенд фреймворки - це програмні засоби, які використовуються для розробки серверної частини веб-додатків. Вони надають розробникам готові рішення та інструменти для створення, підтримки та розширення серверної логіки додатків а також дозволяють обробляти HTTP-запити від клієнтів та надавати відповіді відповідно до бізнес-логіки додатку. Вони дозволяють створювати різноманітні маршрутизатори, контролери та інші механізми для обробки запитів. Багато веб-додатків потребують доступу до бази даних для зберігання та отримання даних. Бекенд фреймворки надають зручний інтерфейс для взаємодії з різними типами баз даних та виконання операцій CRUD (створення, читання, оновлення, видалення). Бекенд фреймворки часто використовуються для створення API (Application Programming Interface), які дозволяють іншим додаткам або сервісам взаємодіяти з веб-додатком. Вони дозволяють обробляти та аналізувати дані, виконувати розрахунки, генерувати звіти та відправляти повідомлення, та забезпечують можливість інтеграції з різними сторонніми сервісами та API для отримання додаткової функціональності або даних. Загалом, бекенд фреймворки є ключовими компонентами розробки веб-додатків, які забезпечують обробку запитів від клієнтів, взаємодію з базою даних, забезпечення безпеки та автентифікації, а також створення API для взаємодії з іншими додатками. [4]

Express.js - це набір інструментів для розробки веб-додатків на мові JavaScript, що базуються на середовищі виконання Node.js, програмній платформі, яка дозволяє виконувати JavaScript-код поза браузером, на сервері або локальному комп'ютері. Express.js дозволяє визначати маршрути для обробки HTTP-запитів. Він надає можливість встановлювати маршрути для різних методів HTTP (GET, POST, PUT, DELETE) та визначати функції-обробники для кожного маршруту. Express.js базується на концепції middleware, яка дозволяє виконувати певні дії перед або після обробки запитів. Middleware може використовуватися для автентифікації, обробки даних, логування, обробки помилок та інших завдань. Робота зі сторонніми middleware дозволяє Express.js легко підключати сторонні middleware для розширення функціональності додатка. Є багато вбудованих middleware, але також іноді може виникнути потреба встановлювати сторонні middleware такі як: body-parser, helmet, cors, morgan або compression, щоб виконати певні завдання. Іншою властивістю Express.js є зручний інтерфейс для роботи з HTTP-запитами та відповідями. Він дозволяє легко отримувати дані з запитів (як параметри, тіло запиту тощо) та надсилати відповіді клієнтам

Шаблонний движок (або движок шаблонів) - це програмне забезпечення, яке використовується для генерації динамічного контенту, такого як HTML-сторінки, на основі шаблонів та даних. Основна ідея полягає в тому, щоб мати шаблон, який містить місця для вставки змінних, та потім заповнити ці місця реальними даними, щоб створити готовий контент. В Express.js шаблонізація дозволяє створювати HTML-сторінки на основі шаблонів, які містять динамічний контент. Express.js не накладає жорстких обмежень на вибір шаблонного движка. Можна використовувати будь-який шаблонний движок, з яким знайомі розробники. Деякі популярні шаблонні движки включають Pug (раніше відомий як Jade), EJS, Handlebars та Mustache. Шаблонні движки дозволяють вставляти змінні та дані у HTML-шаблони, щоб створювати динамічний контент. Наприклад, можна передати дані з сервера у шаблон та використати їх для відображення на сторінці. Шаблонні движки зазвичай мають конструкції управління потоком, такі як умовні оператори, цикли та інші, що дозволяють створювати складніші шаблони з

різноманітними умовами відображення контенту. Деякі шаблонні движки, такі як Pug або Handlebars, підтримують розширення та наслідування шаблонів. Це дозволяє створювати загальні шаблони, які можна розширювати для створення конкретних сторінок.

Express.js є потужним та гнучким інструментом для розробки веб-додатків на Node.js, який дозволяє швидко створювати серверну частину додатку та керувати HTTP-запитами та відповідями.[14]

Django - це високорівневий веб-фреймворк, написаний на мові програмування Python, призначений для швидкої та ефективної розробки веб-додатків. Django спрощує розробку, розділяючи додаток на невеликі, самодостатні модулі, які можна повторно використовувати. Модульність та повторне використання коду виконуються за допомогою різних концепцій та можливостей фреймворка. Django використовує архітектурну модель MVC (Model-View-Controller), яка розділяє додаток на логічні компоненти: модель (Model), представлення (View) та контролер (Controller). Це сприяє модульності, організації коду та забезпечує його повторне використання. Для роботи з базами даних, Django надає ORM (Об'єктно-реляційне відображення), яке дозволяє визначати структуру бази даних у вигляді класів Python (моделей). Це дозволяє повторно використовувати їх для взаємодії з даними у різних частинах додатку. Для зручного керування даними додатка без необхідності написання додаткового коду, Django має вбудований адміністративний інтерфейс, який автоматично створюється на основі моделей даних.

На відміну від Express.js, Django має свій вбудований шаблонний движок, який базується на мові шаблонів Django (Django Template Language - DTL). Це дає можливість легко використовувати шаблони без необхідності встановлення додаткових компонентів. Шаблонний движок Django є вбудованим компонентом фреймворку Django, тому як правило, немає потреби встановлювати додаткові бібліотеки або інструменти для використання шаблонів. DTL має свій синтаксис мови, він включає умовні оператори, цикли, фільтри та інші конструкції, які дозволяють легко виконувати різні завдання в шаблонах. Django має вбудовані

механізми безпеки для захисту від таких атак, як впровадження коду (injection) та перехоплення змінних сесії. Це робить його безпечним вибором для рендерингу сторінок з користувачами взаємодіє. Шаблонний движок Django підтримує вкладеність та можливість включення одного шаблону в інший, що сприяє повторному використанню коду та спрощує організацію шаблонів. Шаблонний движок Django, хоч і поступається деяким движкам Express.js, досить швидкий, і дозволяє швидко генерувати сторінки без зайвого навантаження на сервер. Django пропонує різні вбудовані функціональності, такі як наслідування шаблонів, блоки, включення, шаблонні теги та фільтри, які роблять процес створення складних шаблонів більш зручним і ефективним.

Django - це потужний веб-фреймворк для розробки веб-додатків на мові Python. Він надає широкий набір інструментів та функціональностей для швидкої розробки безпечних, масштабованих та ефективних веб-додатків. Django має вбудований ORM для роботи з базами даних, вбудований адміністративний інтерфейс, систему аутентифікації та авторизації, а також вбудований шаблонний движок. Він підтримує модульну архітектуру, що дозволяє ефективно організовувати код та забезпечує його повторне використання. Django є популярним вибором для створення різноманітних веб-додатків, від невеликих проєктів до складних платформ.[15]

У цілому, бекенд фреймворки є важливим інструментом для розробки веб-додатків, які дозволяють розробникам ефективно створювати та підтримувати потужні та масштабовані серверні додатки.

Іншою важливою складовою при створенні веб-додатка є база даних. У повноцінній розробці веб-додатку робота з базою даних відбувається безпосередньо через код, який розробляється програмістами. Вони створюють моделі даних, контролери та маршрути для взаємодії з базою даних, а також пишуть SQL-запити або використовують ORM (Object-Relational Mapping) для доступу до даних. З іншого боку у системах управління контентом (CMS), таких як WordPress або Joomla, взаємодія з базою даних здійснюється через інтерфейс користувача. Адміністратори сайту можуть використовувати вбудовані

інструменти для додавання, редагування та видалення контенту, такого як статті, зображення або сторінки. Ці CMS автоматично генерують та виконують необхідні SQL-запити для збереження та отримання даних з бази даних, при цьому розробники мають обмежений доступ до внутрішньої структури бази даних. Отже, основна різниця полягає у рівні контролю та гнучкості: у повноцінній розробці веб-додатку розробники мають повний контроль над взаємодією з базою даних і можуть створювати будь-яку логіку обробки даних, тоді як у CMS більшість цієї логіки забезпечується вбудованими функціями та інтерфейсами.

CMS (Content Management System) - це програмне забезпечення, яке дозволяє створювати, редагувати та організовувати вміст на веб-додатку без необхідності глибоких знань в області програмування чи дизайну. CMS використовується для спрощення управління вмістом та полегшення роботи веб-додатків. Система надає веб-інтерфейс, який дозволяє користувачам зручно додавати, редагувати та видаляти контент на сайті без необхідності робити це через код. Багатокористувацькість та рівні доступу дозволяють розділити обов'язки між різними користувачами, наприклад, редактори можуть редагувати вміст, адміністратори - керувати користувачами та налаштуваннями. Багато CMS надають функціонал керування версіями, що дозволяє вам переглядати та відновлювати попередні версії контенту. Деякі CMS дозволяють встановлювати розширення (плагіни або модулі), які розширюють функціональність сайту. CMS часто використовує шаблони для відображення вмісту на веб-додатку. Це дозволяє змінювати дизайн сайту без необхідності переписування вмісту. Вбудована підтримка для різноманітних мультимедійних елементів, таких як зображення, відео та аудіо.

CMS призначені для простоти використання, і вони часто використовують інтерфейси, аналогічні текстовим процесорам. Створення сайту з використанням CMS може бути значно швидше, ніж розробка власного веб-додатка з нуля. Існує декілька популярних CMS, які можуть відповідати вимогам, що обумовлені в попередньому контексті.

WordPress - це безкоштовна система управління контентом (CMS), яка використовується для створення та управління веб-додатками. Вона має широкі можливості та використовується для створення різних типів сайтів, від особистих блогів до корпоративних веб-додатків і онлайн-магазинів. WordPress надає користувачам інтуїтивний і простий у використанні інтерфейс для додавання та редагування контенту, такого як статті, зображення, відео і сторінки. Він має ряд вбудованих функцій, таких як система категорій і міток, коментарі, SEO-оптимізація, можливість встановлення різноманітних тем і плагінів для розширення функціональності сайту. Інтерфейс WordPress інтуїтивний і легкий у використанні, що дозволяє навіть неініційованим користувачам швидко створювати та редагувати контент на своєму сайті, та дозволяє легко налаштовувати та розширювати функціонал додатку за допомогою тем і плагінів. WordPress є безкоштовним програмним забезпеченням з відкритим кодом, що дозволяє використовувати його безкоштовно без обмежень щодо комерційного використання, а велика спільнота користувачів і розробників, забезпечує швидку підтримку, регулярні оновлення та великий вибір тем і плагінів.[5]

Joomla - це безкоштовна і відкрита система управління контентом (CMS), яка використовується для створення і управління веб-додатками та онлайн-додатками. Вона дозволяє користувачам створювати різні типи веб-додатків, від простих особистих блогів до складних корпоративних порталів і онлайн-магазинів. Joomla має велику кількість розширень, таких як компоненти, модулі і плагіни, які дозволяють розширити функціональність веб-додатку. Однією з особливостей Joomla є багатомовність. Joomla має вбудовані інструменти для управління перекладами контенту. Це дозволяє легко додавати, редагувати та видаляти переклади для різних елементів веб-додатку. Joomla дозволяє вам встановлювати мета-дані, такі як заголовки сторінок, мета-описи та ключові слова, для кожної мови окремо. Це допомагає покращити SEO веб-додатку для різних мовних ринків. Багатомовність у Joomla робить її ідеальним вибором для міжнародних веб-додатків або будь-яких проєктів, які мають спільноту користувачів у різних країнах або мовних групах. Вона

забезпечує зручне та ефективне управління контентом для кожної мовної версії веб-додатку. Загалом, Joomla є потужним інструментом для створення веб-додатків і забезпечує різноманітні можливості для користувачів з різними потребами та рівнями навичок.[6]

Іншою популярною CMS є Drupal. Це відкрита система управління контентом, яка використовується для створення різноманітних веб-додатків, від особистих блогів до корпоративних порталів і інтернет-магазинів. Drupal відомий своєю складністю та гнучкістю. Він надає більше можливостей для створення складних веб-додатків та порталів, що потребують спеціалізованих рішень та розширених функцій. Він добре підходить для великих проектів та високовідвідуваних веб-додатків. Він має потужну архітектуру, яка дозволяє масштабувати веб-додатки для великого навантаження та великої кількості користувачів. Також він відомий своїм високим рівнем безпеки та частими оновленнями для запобігання потенційним загрозам. Він часто використовується для створення веб-додатків, де безпека є критичним аспектом. Drupal має розширену систему модулів, яка дозволяє додавати нові функції та розширювати функціональність веб-додатку. Це робить його популярним вибором для розробки спеціалізованих рішень. Загалом, Drupal часто використовується для великих корпоративних та урядових веб-додатків, де потрібні висока складність, гнучкість та надійність. Його потужність та розширені можливості роблять його відмінним вибором для проектів з високими вимогами до функціональності та безпеки.[7]

Кожна з цих CMS має свої переваги, і вибір буде залежати від конкретних потреб проекту, його функціональних вимог та доступності ресурсів для підтримки. У будь-якому випадку, CMS може виявитися досить ефективним рішенням для створення та управління веб-додатками будь-якого напрямлення, без значних витрат на розробку. На рисунку 1.4 представлено технології розробки Web додатків, які були розглянуті у дослідженні.

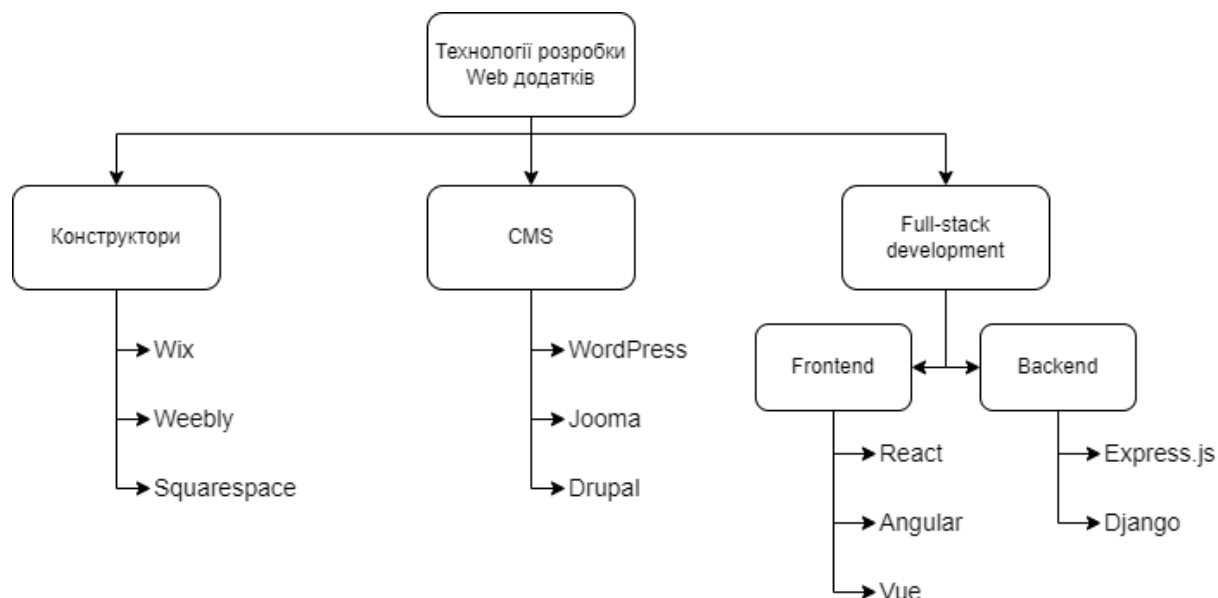


Рисунок 1.4 - Технології розробки Web додатків

В результаті обговорень про постановку задачі розробки веб-додатку музею "Трипільська Культура" з'ясувалося, що замовник не має достатньо ресурсів для підтримки сайту на front-end фреймворках і не має достатньо технічного досвіду для роботи з ними. Також розглядались варіанти використання конструкторів сайтів, але через їх обмежений функціонал, вони не були обрані. Тому було вирішено використовувати систему керування вмістом (Content Management System або CMS) для розробки сайту музею. В результаті вибір впав на WordPress, оскільки він поєднує в собі простоту використання, багатий функціонал та гнучкість, що дозволяє легко налаштовувати сайт під специфічні потреби музею. WordPress також має широку підтримку плагінів та тем, що дає змогу розширити функціональність сайту без необхідності поглиблених технічних знань.

1.3 Дослідження функціональних можливостей Wordpress

WordPress - провідний інструмент для створення веб-додатків у всьому світі, на якому створюється більше половини контенту в Інтернеті. Ця система управління контентом (CMS) з відкритим вихідним кодом є універсальною і

простою у використанні, що робить її ідеальним вибором для користувачів усіх рівнів кваліфікації.

WordPress забезпечує роботу понад 43% сайтів в інтернеті загалом, а також займає понад 60% поміж інших CMS- і ця цифра зростає з кожним днем. На WordPress створюється все - від простих веб-додатків і блогів до складних порталів і корпоративних сайтів, і навіть додатків. На рисунку 1.5 представлено частки ринку найбільш використаних CMS у 2023 році.

Top CMS Platforms by Market Share [2023]

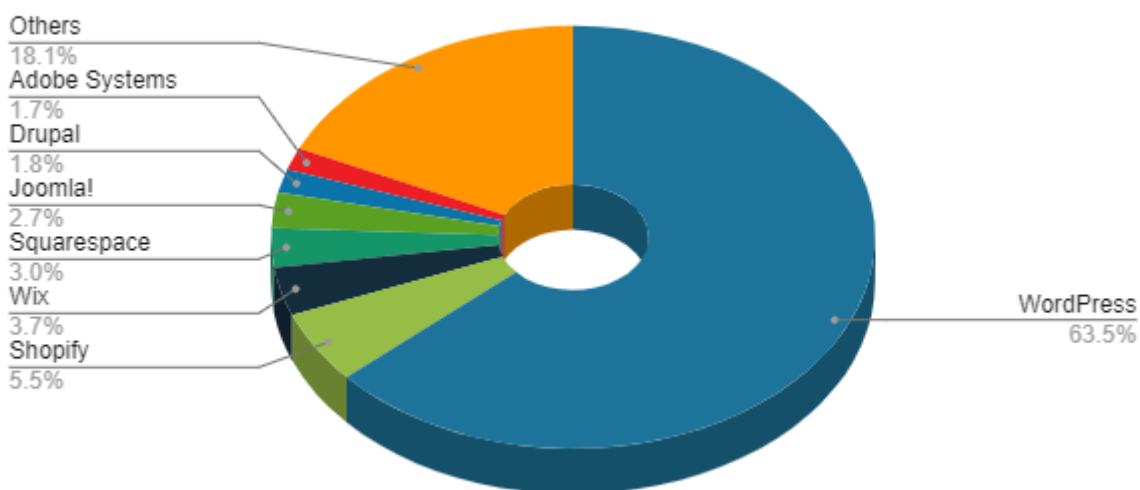


Рисунок 1.5 - найкращі CMS-платформи за часткою ринку

Проект WordPress з відкритим вихідним кодом з часом прогресивно розвивався за підтримки кваліфікованих розробників, дизайнерів, науковців, блогерів та інших ентузіастів. WordPress надає можливість будь-кому створювати та ділитися контентом - від власноруч написаних особистих анекдотів до рухів, що змінюють світ.

Люди з обмеженим технічним досвідом можуть використовувати його "з коробки", а більш технічно підковані люди можуть налаштувати його дивовижним чином.

Базове програмне забезпечення WordPress є простим і передбачуваним, пропонуючи потужні функції для зростання та успіху. Пристрасть дописувачів визначає успіх WordPress, який, у свою чергу, допомагає досягати поставлених цілей. Дописувачі WordPress працюють по всьому світу і присвятили незліченну кількість годин створенню інструменту, який дає можливість кожному висловити свою думку.

WordPress використовує дві ключові технології - PHP та MySQL - для свого функціонування та управління веб-додатком. PHP використовується для вставки динамічного контенту у HTML-сторінки. Наприклад, він обробляє запити на виведення конкретної статті, сторінки або іншого контенту на веб-додатку. Коли користувач відправляє запит до сервера, сервер використовує PHP для обробки цього запиту. PHP взаємодіє з базою даних, щоб отримати необхідну інформацію для відображення на сторінці. Наприклад, якщо користувач хоче переглянути статтю на блозі, PHP виконує запит до бази даних WordPress для отримання контенту цієї статті. На основі інформації з бази даних PHP генерує HTML-код для сторінки. Цей код включає в себе контент статті, заголовки, меню, сайдбари та інші елементи, які мають бути відображені. Згенерований HTML-код відправляється назад на сервер, який потім передає його до браузера користувача. Браузер розуміє цей HTML і відображає сторінку користувачу. Цей процес відбувається кожного разу, коли користувач переходить на нову сторінку або виконує дії на веб-додатку. PHP відповідає за генерацію відповідного контенту для кожного запиту користувача, що дозволяє WordPress відображати сторінки змістовно та динамічно.

У WordPress, інформація про користувачів, публікації, коментарі та інші дані зберігаються у різних таблицях MySQL. Наприклад, таблиці можуть містити дані про користувачів, публікації, коментарі, теми, плагіни тощо. WordPress використовує мову SQL для взаємодії з базою даних MySQL. Під час процесу інсталяції WordPress, користувач має вказати існуючу базу даних MySQL, до якої WordPress буде підключатися. Після цього WordPress використовує з'єднання з базою даних, щоб зберігати та отримувати інформацію, таку як пости, сторінки,

коментарі, налаштування тем і плагінів, користувачі тощо. Запити використовуються для вибору, вставки, оновлення та видалення даних з бази даних. MySQL добре взаємодіє з PHP та WordPress, забезпечуючи швидкий доступ до інформації та оптимальну роботу сайту.[2]

Разом PHP та MySQL створюють потужну комбінацію для розробки та управління веб-додатками, особливо в разі великих та динамічних платформ, таких як WordPress. Ці технології дозволяють взаємодіяти з базою даних, обробляти запити та генерувати динамічний контент, роблячи WordPress ефективним та гнучким рішенням для розробки веб-додатків.

WordPress постачається з REST API, яке дозволяє зовнішнім додаткам взаємодіяти з веб-додатком. Це важлива складова для інтеграції WordPress з іншими системами та створення зовнішніх додатків. REST API в WordPress - це набір програмних інтерфейсів, які дозволяють зовнішнім додаткам взаємодіяти з веб-додатком через стандартні HTTP-запити. REST API дозволяє отримувати доступ до різних типів даних в WordPress, таких як пости, сторінки, користувачі, коментарі тощо. Це дає можливість створювати зовнішні додатки, які можуть виконувати всі основні операції з базою даних, відомі як CRUD (створення, читання, оновлення та видалення). Завдяки REST API можна створювати зовнішні додатки та сервіси, які розширюють функціональність веб-додатку. Наприклад, це може бути мобільний додаток для управління вмістом, інтеграція з іншими платформами або автоматизація операцій. Загалом, REST API в WordPress відкриває широкі можливості для розширення та інтеграції веб-додатку з іншими системами та додатками, що робить WordPress більш гнучким і потужним інструментом для веб-розробки.

WordPress має власний адміністративний інтерфейс, який дозволяє адміністраторам керувати вмістом, темами, плагінами та іншими аспектами веб-додатку без необхідності редагування коду. Адміністративний інтерфейс WordPress - це інтуїтивно зрозумілий і легкий у використанні інтерфейс, призначений для управління різними аспектами веб-додатку, що використовується

для управління вмістом та налаштуваннями сайту. Він функціонує зовсім в іншому просторі ніж фронтенд, що дає змогу користувачам працювати зі своїм сайтом безпосередньо через браузер. Адміністратори можуть додавати, редагувати та видаляти пости, сторінки, коментарі, категорії і теги, користуючись зручними інтерфейсами редагування. За допомогою адміністративного інтерфейсу можна встановлювати, активувати, деактивувати та видаляти теми та плагіни. Також можна вносити налаштування інші настройки пов'язані з темами та плагінами. Загальні налаштування сайту, такі як назву сайту, опис, адресу електронної пошти адміністратора, формат дати та часу, налаштування пошти та інше, також можна налаштовувати через інтерфейс. Додатково, адміністративний інтерфейс може містити різноманітні інші функції, такі як статистика відвідуваності, інструменти для резервного копіювання та відновлення, інструменти для оптимізації швидкості сайту та інші. Всі ці функції реалізовані за допомогою PHP, JavaScript та HTML/CSS, що дозволяє адміністраторам просто і ефективно управляти своїм веб-додатком без потреби в глибоких технічних знаннях.

Система плагінів у WordPress є ключовим елементом, що дозволяє розширювати функціональність веб-додатку, додавати нові можливості та впроваджувати різноманітні зміни без необхідності власноручного кодування або редагування основного коду WordPress. Плагіни можна легко встановлювати з офіційного репозиторію WordPress або інших джерел, наприклад у форматі ZIP-архіву. Після встановлення їх можна активувати в адміністративному розділі сайту. WordPress постачається з базовою функціональністю, але плагіни дозволяють додавати розширення, такі як віджети, короткі коди, адаптивні зображення, SEO-оптимізацію тощо. Плагіни для аналітики дозволяють вам відстежувати відвідуваність, аналізувати поведінку користувачів та отримувати звіти про ефективність контенту. Плагіни для захисту допомагають у попередженні атак, моніторингу безпеки та реагуванні на потенційні загрози.

1.4 Вибір теми CMS Wordpress для веб-додатку музею “Трипільська Культура”

Вибір теми під час створення сайту на WordPress має велику важливість, оскільки тема визначає зовнішній вигляд та стиль веб-додатку. Тема є першим, що бачать відвідувачі, коли заходять на сайт, і вона створює перше враження про нього. Професійний та привабливий дизайн робить сайт більш привабливим для користувачів.

Тема визначає колірну схему, шрифти та інші дизайнерські елементи, що допомагають створити брендовий образ сайту. Важливо, щоб цей образ відповідав корпоративним цінностям та індивідуальності. Тема повинна відповідати типу контенту, який публікується на сайті. Наприклад, якщо замовник блог, то тема повинна підтримувати читабельний текст та можливості виведення останніх публікацій. Сучасні теми повинні бути адаптовані для різних пристроїв, включаючи мобільні та планшетні. Мобільна сумісність важлива для забезпечення зручного перегляду будь-якого сайту на різних пристроях.

Деякі теми вже мають вбудовану оптимізацію для пошукових систем (SEO), що полегшує вам роботу з поліпшення видимості сайту у результатах пошуку. Тема може впливати на функціональність сайту. Деякі теми мають вбудовані функції, такі як слайдери, галереї, підтримка соціальних мереж і т.д. Розробники деяких тем надають регулярні оновлення та підтримку, що важливо для безпеки та сумісності із новими версіями WordPress.

Оскільки тема це по суті лише початковий макет для сайту, вона повинна дозволяти розробникам легко налаштовувати її відповідно до їх потреб. Це включає зміну кольорів, шрифтів, макетів тощо. Важливо, щоб тема була оптимізована для швидкості завантаження сторінок, оскільки це впливає на користувацький досвід та SEO.

Обираючи тему, слід ретельно аналізувати її можливості та відповідність потребам замовника, щоб забезпечити ефективний та професійний вигляд сайту.

Саме тому, моїм завданням було підібрати яка б потенційно підходила по тематиці та атмосфері музею трипільської культури, а також відповідала поставленим вимогам. Ось декілька тем які ми з замовником розглядали під час обговорення дизайну сайту:

Тема WordPress під назвою "Unimate" призначена для мистецьких галерей та музеїв, поєднуючи елегантність і функціональність. Завдяки своєму дизайну, вона підкреслює бізнес та надає зручну навігацію, а також адаптивний дизайн для захопливого онлайн-досвіду для любителів мистецтва. Основні особливості включають:

- Легке імпортування демо-контенту та можливість редагувати зміст.
- Динамічні варіанти відображення подій та виставок, включаючи слайдери, каруселі, ґріди та анімовані сторінки.
- Безпечний магазин WooCommerce для продажу мистецтва.
- Портфоліо для елегантного представлення робіт.
- Простий, але потужний блог для розповіді про художній шлях.
- Легка настройка та можливість адаптації під бренд галереї.
- SEO-оптимізація для високого рейтингу у пошукових системах.
- Інтеграція з розсилками для збереження зв'язку з відвідувачами.
- Використання Elementor для легкої розробки привабливих та відгукнітних макетів.
- Оптимізована продуктивність та технології WordPress для ефективного управління.

Mooseoom - це тема WordPress для мистецьких галерей та виставок, що має в собі унікальні функціональності та блоки, підтримується Aheto плагіном. Вона включає унікальні сторінки для різних сервісів, такі як "Про нас", "Керівництво", "Локації", "Журнал", "Нагороди", "Виставки", "Галереї" та інші, необхідні для успішного створення веб-додатку. Тема може використовуватися не лише для мистецтва, але й легко адаптується для різних застосувань.

Основні особливості:

- Повністю адаптивна та розроблена з урахуванням досліджень, щоб гарантувати чудовий вигляд на екранах планшетів та мобільних телефонів.
- Унікальні сторінки для представлення послуг та інших ключових елементів, таких як "Про нас", "Локації", "Виставки" і т.д.
- Сумісність з популярними плагінами, такими як Contact Form 7, WooCommerce, Booked, Google Maps, Elementor та Gutenberg.
- Висока реактивність та мобільна сумісність для привабливого вигляду на будь-яких пристроях.
- Документація для детального вивчення можливостей теми.
- Професійна технічна підтримка для вирішення будь-яких питань або проблем з використанням теми.
- лише для мистецтва, але й легко адаптується для різних застосувань.

Ozeum - це чудова, свіжа та чиста тема WordPress для мистецьких галерей, музеїв та виставок. З сучасним та функціональним дизайном, вона ідеально підходить для веб-додатків виставкових центрів, мистецьких блогів, онлайн-магазинів музеїв, конференц-центрів чи експоцентрів, публічних бібліотек чи видавничих будинків.

Основні особливості теми "Ozeum":

- Професійний дизайн та можливість одноразової установки демо-контенту.
- Побудована з використанням Elementor Page Builder для легкої модифікації вмісту сторінок.
- Включає в себе плагіни Revolution Slider та Essential Grid для створення незабутнього веб-додатку.
- Підтримка плагіну Contact Form 7 для зручного взаємодії з відвідувачами сайту.
- Повністю адаптивна та мобільно-дружня з можливістю вибору між блочним та повношироким макетами сторінок.

- Вбудована підтримка WooCommerce для створення інтернет-магазину музею.
- Зручний пошук Ajax, взаємодійність з соціальними мережами та інші функціональні можливості.

Тема також підтримує популярні плагіни, такі як The Events Calendar, MailChimp for WP, WP GDPR Compliance, а також має багато інших вбудованих можливостей для налаштування та оптимізації. З "Ozeum" ви можете легко надати експозицію музею чи виставці в онлайн-середовищі. В результаті обговорень і розглядів, замовник прийняв рішення вибрати тему "Ozeum" для створення свого музейного веб-додатку, вважаючи, що вона найбільше відповідає його бізнес-потребам та естетичним вимогам

Таблиця 1.1 - Порівняння тем Wordpress

Тема	Основні переваги	Основні недоліки
Unimate	Легке імпортування демо-контенту. Динамічні варіанти відображення подій та виставок. Безпечний магазин WooCommerce. Елегантне портфоліо. Простий блог. Легка настройка та можливість адаптації під бренд галереї. SEO-оптимізація. Інтеграція з розсилками. Використання Elementor. Оптимізована продуктивність.	Можливо, не так багато унікальних сторінок для представлення різних сервісів, як у Mooseoom.
Mooseoom	Повністю адаптивна. Унікальні сторінки для представлення послуг, таких як "Про нас", "Локації", "Виставки" і т.д. Сумісність з популярними плагінами (Contact Form 7, WooCommerce, Booked, Google Maps, Elementor, Gutenberg). Висока реактивність та мобільна сумісність. Документація та професійна технічна підтримка.	Можливо, складніше налаштування для новачків.

Продовження таблиці 1.1

Ozeum	Професійний дизайн. Одноразова установка демо-контенту. Використання Elementor Page Builder. Плагіни Revolution Slider та Essential Grid. Підтримка Contact Form 7. Повністю адаптивна. Вбудована підтримка WooCommerce. Зручний пошук Ajax. Інтерактивність з соціальними мережами. Підтримка плагінів The Events Calendar, MailChimp for WP, WP GDPR Compliance.	Може потребувати додаткового часу для налаштування всіх функцій.
-------	--	--

Тема Ozeum має ряд переваг, які роблять її привабливою для використання у веб-додатку музею "Трипільська Культура". Її професійний дизайн, можливість легкої установки демо-контенту, використання потужного редактора Elementor, підтримка різноманітних плагінів та адаптивність роблять її найкращим вибором серед розглянутих тем. Ozeum також забезпечує широкі можливості для налаштування та оптимізації, що є важливим для створення високоякісного онлайн-досвіду для відвідувачів музею.

1.5 Дослідження можливостей теми Ozeum

Тема WordPress "Ozeum" призначена для створення веб-додатків музеїв, виставкових галерей та схожих культурних закладів. Тема має сучасний дизайн, який привертає увагу і сприяє зручному користуванню. Ozeum підтримує різноманітні популярні плагіни, такі як Elementor, WooCommerce, Contact Form 7, які розширюють функціональність теми. Тема повністю адаптована під різні пристрої та розміри екранів, що забезпечує її коректне відображення на всіх пристроях, та має вбудовану оптимізацію для пошукових систем, що допомагає сайту піднятися в рейтингу пошукових запитів. Ozeum має вбудовану підтримку

для створення портфоліо та відображення експозицій, що дозволяє ефективно представити арт-роботи або експонати. На рисунку 1.6 зображено панель кастомізації теми Ozeum.

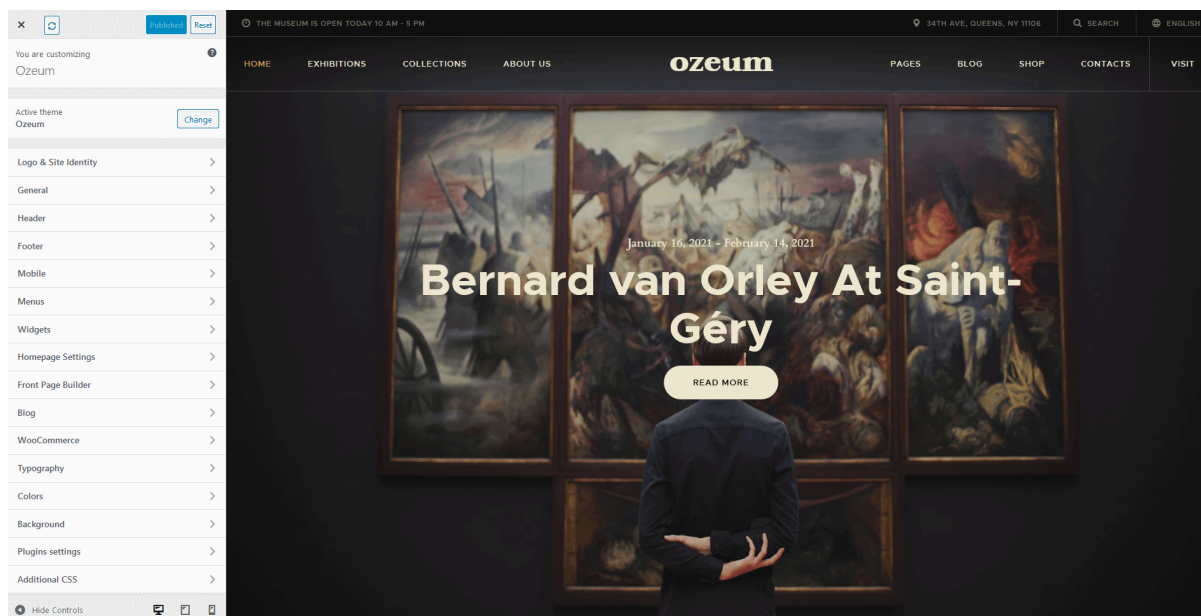


Рисунок 1.6 - Панель кастомізації теми Ozeum

Інтеграція з Elementor дозволяє використовувати популярний конструктор сторінок для створення унікальних макетів сторінок безпосередньо з адміністративного інтерфейсу. Elementor - це популярний конструктор сторінок для WordPress, який дозволяє користувачам створювати красиві та професійні макети сторінок за допомогою інтуїтивного інтерфейсу Drag-and-Drop, це надає можливість перетягувати та розміщувати елементи на сторінці безпосередньо з візуального інтерфейсу, що дозволяє швидко створювати макети сторінок без необхідності знань коду. Елементор містить багато вбудованих елементів, таких як текстові блоки, зображення, кнопки, слайдери, форми, карти тощо, що дозволяє створювати різноманітні макети сторінок, а також має велику бібліотеку готових шаблонів та блоків, які можна використовувати для швидкого створення сторінок. Це дозволяє значно скоротити час на розробку веб-додатку. Щодо інтеграції з темою Ozeum, вона дозволяє використовувати всі можливості конструктора

сторінок Elementor для налаштування вигляду сайту, включаючи зміну макетів, кольорів, шрифтів, додавання віджетів та іншого. Це робить процес створення та налаштування веб-додатку ще більш простим та зручним.

Плагін WooCommerce є одним з найпопулярніших інструментів для створення і управління інтернет-магазинами на базі WordPress. Цей плагін дозволяє створювати повноцінний інтернет-магазин на веб-додатку, і дозволяє додавати товари, категорії товарів, встановлювати ціни, налаштовувати опції доставки та оплати. WooCommerce надає зручний інтерфейс для управління замовленнями у магазині. Замовлення можна переглядати, редагувати та виконувати, встановлювати статуси замовлень та взаємодіяти з клієнтами. WooCommerce підтримує різноманітні платіжні шлюзи, такі як PayPal, Stripe, Square та інші. Це дозволяє клієнтам зручно та безпечно здійснювати оплату за їхні покупки. Оzeum, як і будь-яка інша тема WordPress, інтегрується з WooCommerce, що дозволяє створювати стильні та професійні на вигляд сторінки магазину, які гармонійно впишуться у дизайн сайту. Загалом, інтеграція WooCommerce з темою Ozeum дозволяє створити ефективний інтернет-магазин для музею або галереї, що допоможе просувати товари та послуги онлайн.

Іншим важливим плагіном у роботі теми є Contact Form 7. Це безкоштовний плагін для WordPress, який дозволяє створювати та керувати контактними формами на веб-додатку. Плагін має простий та інтуїтивно зрозумілий інтерфейс для створення та налаштування форм, що дозволяє додавати різні поля форми, встановлювати обов'язковість заповнення, налаштовувати повідомлення про успішне відправлення форми та інше. Contact Form 7 використовує HTML-розмітку для налаштування вигляду та структури форми. Це дозволяє створювати форми з різноманітним виглядом та функціональністю. Плагін підтримує вбудовану CAPTCHA для захисту від спаму, що забезпечує форму від небажаних повідомлень. Contact Form 7 може використовувати Аjax для відправлення форм без перезавантаження сторінки. Це дозволяє користувачам надсилати форми без затримок та зрушень. Плагін має велику кількість

налаштувань, які дозволяють контролювати різні аспекти форми, такі як адреса електронної пошти для надсилання повідомлень, текст кнопки відправки форми, стилізація і багато іншого. Contact Form 7 може інтегруватись з будь-якою темою WordPress, включаючи Ozeum. Ozeum, як і будь-яка інша тема, надає місце для вбудовування контактної форми, створеної за допомогою Contact Form 7. [19]

2 РОЗРОБКА ВЕБ ДОДАТКУ МУЗЕЮ “ТРИПІЛЬСЬКА КУЛЬТУРА” З CMS WORDPRESS

2.1 Розробка UI/UX дизайну веб-додатку музею “Трипільська Культура” за допомогою Figma.

Розробка UI/UX є важливим етапом, який допомагає забезпечити успіх веб-додатка у WordPress, забезпечуючи задоволення користувачів, покращення ефективності та збільшення конверсії. Гарний UX робить веб-додаток привабливим та зручним для користувачів. Це дозволяє забезпечити задоволення від використання продукту та збільшує ймовірність повернення користувачів. Добре спроектований інтерфейс дозволяє користувачам легко знаходити потрібну інформацію та виконувати бажані дії. Це зменшує час, потрібний для досягнення мети, та забезпечує більшу продуктивність. Сучасні користувачі мають високі очікування від веб-додатків. Якщо веб-додаток не забезпечує високий рівень UX, він може втратити конкурентну перевагу. Ефективний дизайн UI/UX може позитивно вплинути на конверсію, збільшуючи кількість користувачів, які виконують бажані дії, такі як покупки, реєстрація або заповнення форм. Гарний дизайн веб-додатка сприяє створенню сильної брендваної ідентичності та дозволяє позиціонувати бренд як професійний та надійний. [16]

Важливим етапом у розробці UI/UX додатку є створення інформаційної архітектури (IA). Це процес організації та структурування інформації на веб-сайті або веб-додатку для забезпечення зручного та ефективного користувацького досвіду. Основні принципи створення IA полягають у розподілі контенту на категорії та підкатегорії, створенні логічної структури та визначенні шляхів навігації для користувачів. Головними цілями створення інформаційної архітектури є: створення легко зрозумілої та інтуїтивної навігації для користувачів, організованість великої кількості інформації у логічну та зрозумілу структуру, та покращення SEO. Створення IA включає в себе такі етапи як аналіз

бізнес-потреб, вивчення цільової аудиторії, створення карт сайту, прототипів та дерев навігації. Цей процес допомагає забезпечити, що веб-додаток буде легким у використанні, логічно організованим та відповідатиме потребам як користувачів, так і бізнесу. На рисунку 2.1 зображено карту сайту веб-додатку музею “Трипільська Культура” розробленої у Figma.

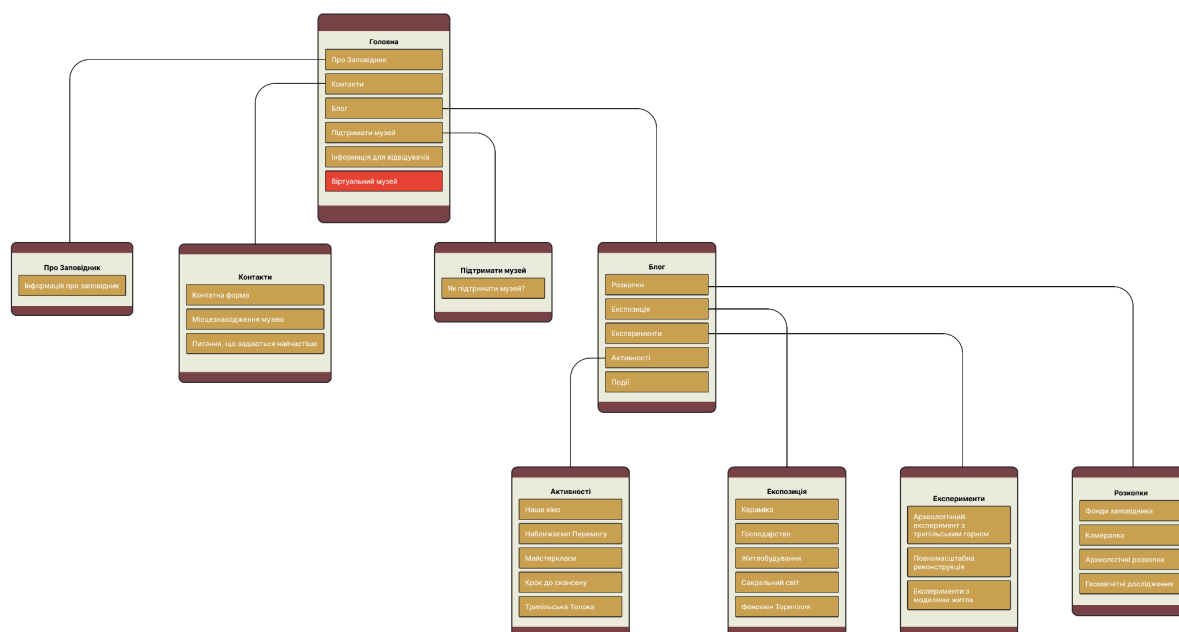


Рисунок 2.1 - карта сайту веб-додатку музею “Трипільська Культура”

Створення карти сайту - це процес визначення структури та організації веб-сайту або веб-додатка. Основна мета - візуалізувати всі сторінки та їхні взаємозв'язки для забезпечення логічної та зрозумілої навігації для користувачів. Одним з перших кроків створення карти сайту є визначення основних категорій або розділів, які будуть присутні на веб-сайті. Це можуть бути, наприклад сторінки, "Головна", "Про нас", "Послуги", "Контакти" тощо. Після визначення основних категорій визначаються підкатегорії або сторінки, які будуть входити в кожен розділ, та взаємозв'язки між розділами та сторінками для створення логічної структури навігації. Наприклад, в головному меню можуть бути посилання на основні розділи, а в додатковому меню - на підкатегорії. Карта сайту може бути представлена у вигляді діаграми, дерева, таблиці або іншого графічного

зображення, яке відображає структуру та взаємозв'язки між сторінками. Створення карти сайту є важливим етапом у процесі розробки веб-додатка, оскільки воно допомагає забезпечити логічну та ефективну організацію контенту та навігацію для користувачів.

Зразком для створення карти сайту була взята структура теми Ozeum, але розділи та сторінки були додані та виправлені відповідно до вимог та потреб веб-додатку музею "Трипільська Культура". Результат - гнучка та інформативна карта сайту, що відповідає всім потребам користувачів.

Figma - це інструмент для дизайну та прототипування веб-додатків і мобільних додатків. Вона працює у веб-браузері і дозволяє дизайнерам створювати векторні графічні елементи, макети сторінок, інтерактивні прототипи, а також спільно працювати над проектом з іншими учасниками команди. Дизайнер може створити макети сторінок веб-додатку, використовуючи різні елементи дизайну, такі як кнопки, текстові блоки, зображення тощо. Figma надає широкий вибір інструментів для створення та редагування елементів. Figma дозволяє створювати інтерактивні прототипи, які демонструють взаємодію елементів на сторінці. Це допомагає дизайнерам визначати логіку та потік користувача. Figma забезпечує можливість спільно працювати над проектом з іншими учасниками команди. Всі зміни відображаються в реальному часі, що дозволяє зручно співпрацювати та обмінюватися відгуками. Після завершення дизайну сторінок їх можна експортувати у вигляді зображень або коду, що полегшує подальший процес розробки. Figma є потужним інструментом для дизайну та прототипування веб-додатків. Він дозволяє дизайнерам створювати візуальні макети сторінок, прототипи, а також спільно працювати над проектом з іншими учасниками команди. В цілому, Figma є потужним інструментом для розробки дизайну сторінок веб-додатку, який дозволяє створювати професійні та ефективні дизайнерські рішення.

Оскільки структура веб-додатку вже була побудована на основі використаної теми, Figma, здебільшого, використовувалася лише для розробки дизайну сторінок додатку.

Першим етапом розробки дизайну сторінок було створення макетів сторінок. Створення макетів сторінок для веб-додатку - це процес розробки візуальних моделей або прототипів інтерфейсу, які відображають, як виглядатимуть різні сторінки та елементи веб-додатку на різних етапах розробки. Це важлива частина процесу розробки, оскільки макети допомагають зрозуміти, яким буде кінцевий продукт та як він буде взаємодіяти з користувачами. Створення макетів сторінок для веб-додатку музею у Figma - це процес, що складається з кількох етапів, які допомагають створити візуальну модель інтерфейсу додатку. На початку аналізувались вимоги до веб-додатку, бізнес-цілі та потреби користувачів. Сторінки музею повинні містити достатньо інформації про експозиції, події, активності та інші аспекти діяльності музею, а користувачі повинні з легкістю знаходити потрібну інформацію та навігувати між різними розділами сторінки. Графічне оформлення сторінок музею повинно бути привабливим та професійним, відповідати корпоративному стилю та вражати користувачів. та бути оптимізовані для відображення на різних пристроях, включаючи мобільні телефони та планшети. Ці вимоги допомагають забезпечити якісну та зручну взаємодію користувачів з веб-додатком музею. Після визначення вимог, створювались перші наброски ідеї на основі аналізу. Для визначення загальної композиції та розташування елементів на сторінці, використовувались скетчі або wireframes. Створення скетчів та wireframes - це перші кроки у процесі дизайну веб-додатку або іншого інтерфейсу. Скетчі - це швидкі, ручні наброски ідеї, як правило, виконані вручну або за допомогою графічних планшетів. Вони можуть бути простими малюнками або схематичними кресленнями, які демонструють загальну композицію та розташування елементів на сторінці. Скетчі допомагають візуалізувати концепції та швидко тестувати різні варіанти дизайну. Wireframes - це більш детальні схеми, які відображають структуру сторінки або додатку. Вони зазвичай містять блоки контенту та елементи

інтерфейсу, такі як кнопки, поля введення, текстові блоки тощо, але не містять деталей дизайну, таких як кольори або зображення. Wireframes дозволяють зосередитися на логіці розміщення елементів та функціональності сторінки без відволікання на деталі. Для розробки дизайну сторінок веб-додатку музею "Трипільська культура", здебільшого використовувалися саме wireframes. Це ефективний підхід, який дозволяє зосередитися на структурі та функціональності сторінок без відволікання на деталі дизайну.

Figma є потужним інструментом для дизайну та прототипування, і одним з його основних переваг є можливість створення та використання компонентів. Компоненти у Figma дозволяють створювати повторювані елементи, які можна використовувати на різних сторінках або в різних проектах. Це значно полегшує підтримку дизайну, зменшує час на внесення змін і забезпечує узгодженість інтерфейсу. Коли я працював з Figma, я створював компоненти для різних елементів інтерфейсу, таких як кнопки, заголовки, форми, іконки та інші стандартні елементи. Основна перевага компонентів полягає в тому, що вони є основними елементами, з яких можна створювати екземпляри. Внесення змін до головного компонента автоматично оновлює всі його екземпляри, що значно полегшує управління дизайном. Наприклад, для веб-додатку музею "Трипільська Культура" я створив компоненти для кнопок різних стилів, форм введення, карток артефактів та інших елементів, які повторюються на різних сторінках. Це дозволило мені швидко оновлювати дизайн у випадку змін вимог або зворотного зв'язку від замовника, забезпечуючи узгодженість і високу якість всього інтерфейсу. Крім того, Figma дозволяє створювати варіанти компонентів, що надає ще більше гнучкості. Наприклад, я створював кнопки з різними станами (нормальний, наведений, натиснутий), що дозволяло легко адаптувати їх до різних сценаріїв використання. Компоненти також можуть бути вкладені один в одного, що дозволяє створювати складніші структури і забезпечувати їх легке управління. Загалом, використання компонентів у Figma дозволяє оптимізувати робочий процес дизайну, забезпечує узгодженість інтерфейсу, зменшує кількість помилок і

значно прискорює процес внесення змін, що є надзвичайно важливим для ефективної розробки та підтримки веб-додатків.

Після схвалення скетчів, дизайнери розробляють детальний дизайн сторінок. Детальний дизайн - це процес, під час якого розробляються конкретні візуальні елементи та стилі для кожної сторінки веб-додатку. Дизайнери використовують Figma або інші інструменти для створення векторних зображень, які включають елементи інтерфейсу, такі як кнопки, поля вводу, меню, заголовки та інші. Визначаються основні та допоміжні кольори, які використовуються на сторінках веб-додатку. Кольори підбираються відповідно до бренду та забезпечують консистентність у всьому дизайні. Вибираються шрифти для заголовків, текстів та інших елементів. Дизайнери встановлюють розміри шрифтів, міжрядкові інтервали, курсиви та жирність для різних типів тексту. Після цього додаються графічні елементи, такі як ілюстрації, іконки, фотографії та інші візуальні ефекти, які підкреслюють концепцію та атмосферу веб-додатку, та враховуються різні розміри екранів та пристроїв, щоб забезпечити оптимальний вигляд та взаємодію на різних пристроях. На основі детального дизайну створюються прототипи сторінок, які можна використовувати для тестування взаємодії та отримання зворотнього зв'язку від користувачів. Цей процес дозволяє розробникам та дизайнерам встановити конкретні візуальні стилі та елементи, які стануть основою для розробки функціональних та привабливих сторінок веб-додатку.

Детальний дизайн для веб-додатку музею “Трипільська Культура” виконувався у Figma. У Figma були створені векторні зображення елементів інтерфейсу, такі як кнопки, поля вводу, меню, заголовки та інші, відповідно до концепції дизайну. За допомогою теми Ozeum визначалися основні та допоміжні кольори, що використовуються на сторінках веб-додатку, з метою забезпечення консистентності у всьому дизайні, та вибиралися шрифти для заголовків, текстів та інших елементів, встановлювалися розміри шрифтів, міжрядкові інтервали, курсиви та жирність для різних типів тексту. Були додані графічні елементи, такі

як ілюстрації, іконки, фотографії та інші візуальні ефекти, що підкреслюють концепцію та атмосферу музею веб-додатку. Цей процес допоміг створити консистентний та привабливий дизайн для веб-додатку музею "Трипільська Культура", який відповідає бренду та забезпечує зручну та естетичну взаємодію з користувачем. На рисунку 2.2 зображено загальний план проекту Tripillia у Figma.



Рисунок 2.2 - загальний план проекту Tripillia у Figma

Після створення прототипів сторінок, вони використовуються для тестування взаємодії та отримання зворотнього зв'язку від користувачів. Процес тестування та зворотного зв'язку прототипів є важливою складовою розробки веб-додатків і включає наступні кроки: створення прототипу, тестування прототипу, збір зворотного зв'язку, аналіз зворотного зв'язку та внесення змін. Створені прототипи піддаються тестуванню, під час якого вони аналізуються з точки зору функціональності, навігації, взаємодії та користувацького досвіду (UX). Тестувальний процес може включати тестування на різних пристроях та браузерах, а також тестування з різних точок зору користувачів. Для перевірки різних аспектів прототипу проводяться різні види тестів. Перш за все, це функціональне тестування, яке перевіряє, чи працюють всі функції прототипу

відповідно до очікувань та специфікацій. Далі, тестування навігації оцінює зручність та ефективність навігації по сторінках прототипу, включаючи меню, посилання та інші елементи навігації. Тестування взаємодії визначає, як користувачі взаємодіють з різними елементами прототипу, такими як кнопки, поля вводу, випадаючі списки тощо. Тестування користувацького досвіду (UX) оцінює задоволення користувачів взаємодією з прототипом та загальним досвідом використання. Тестування на різних пристроях та браузерах визначає, як прототип відображається та працює на різних пристроях та в різних браузерах. Нарешті, тестування з точок зору різних користувачів включає тестування прототипу з різних категорій користувачів, які можуть мати різні потреби, умови та очікування. Після тестування користувачі та інші зацікавлені сторони надають зворотний зв'язок щодо прототипу. Цей зворотний зв'язок може містити відгуки, коментарі, пропозиції щодо покращення та іншу інформацію, що допоможе у вдосконаленні прототипу. Після тестування користувачі та інші зацікавлені сторони надають зворотний зв'язок щодо прототипу. Цей зворотний зв'язок може містити відгуки, коментарі, пропозиції щодо покращення та іншу інформацію, що допоможе у вдосконаленні прототипу. Збір зворотного зв'язку може проходити через різні канали комунікації. Один з найпоширеніших способів - це анкетування. Користувачам надсилають анкети або опитувальники, де вони можуть висловити свої враження та думки щодо прототипу. Також можуть використовуватися фокус-групи або інтерв'ю, де представники цільової аудиторії обговорюють прототип і надають свої коментарі та пропозиції. Крім того, можливість залишити коментарі прямо на прототипі або в електронній формі також дозволяє користувачам зручно висловлювати свої думки. Команді розробників важливо активно слухати та аналізувати зібраний зворотний зв'язок для вдосконалення прототипу та відповіді на потреби користувачів. Коли команда отримує достатньо зворотного зв'язку, розробники аналізують зібрану інформацію для виявлення патернів, тенденцій та проблем, які можуть потребувати внесення змін у прототип. Під час аналізу зворотного зв'язку розробники оцінюють інформацію, що надійшла від користувачів, для виявлення

ключових паттернів, тенденцій та проблем. Процес може включати категоризацію коментарів за темами, порівняння різних думок та пропозицій, ідентифікацію найбільш поширених проблем тощо. Для успішного аналізу важливо враховувати ці проблеми та використовувати стратегії для їх вирішення, такі як систематичний підхід до обробки даних, регулярна звітність та співпраця з різними членами команди.

На основі аналізу зворотного зв'язку розробники вносять необхідні зміни у прототип, з метою поліпшення його функціональності та користувацького досвіду. Цей процес тестування та зворотного зв'язку допомагає забезпечити високу якість веб-додатку шляхом виявлення та виправлення проблем ще на стадії розробки, що в результаті призводить до задоволення потреб користувачів та досягнення більшого успіху проекту.

Процес тестування прототипів сторінок веб-додатку музею "Трипільська Культура" починався з функціонального тестування каталогу артефактів, зокрема перевірка можливості додавання, видалення та редагування артефактів, і перевірка фільтрації артефактів за різними категоріями та характеристиками. Далі перевірялись зручності навігації по сторінках каталогу артефактів та інших розділів веб-додатку, а також робота кнопок, полів вводу та інших елементів інтерфейсу. Перевірка відображення та роботи прототипу перевірялася на різних пристроях та в різних браузерах, для забезпечення якості користувацького досвіду. Ці етапи тестування допомогли виявити та виправити проблеми у веб-додатку ще на етапі розробки дизайну, забезпечивши його оптимальну функціональність та зручність використання для користувачів.

Після успішного завершення процесу тестування прототипів сторінок веб-додатку музею "Трипільська Культура", наступним етапом було узгодження дизайну з замовником. На цьому етапі представники музею мали можливість переглянути та оцінити розроблені прототипи, висловити свої зауваження та пропозиції щодо дизайну та функціональності веб-додатку. Після узгодження дизайну з замовником розпочалася фаза реалізації проекту.

2.2 Розробка функціоналу сторінок веб-додатку музею “Трипільська Культура”

Головна сторінка веб-додатку музею "Трипільська Культура" - це віртуальна візитна картка музею, яка вітає відвідувачів та надає короткий огляд його основних можливостей. Сторінка презентує відвідувачу інформацію про структуру та роботу музею, його діяльність та різноманітні активності та події. Звідси користувач може відвідати інші, більш інформативні частини сайту, такі як: “Каталог артефактів”, “Про заповідник” та Блог. На рисунку 2.3 зображено кінцевий дизайн головної сторінки веб-додатку, розроблений у Figma.

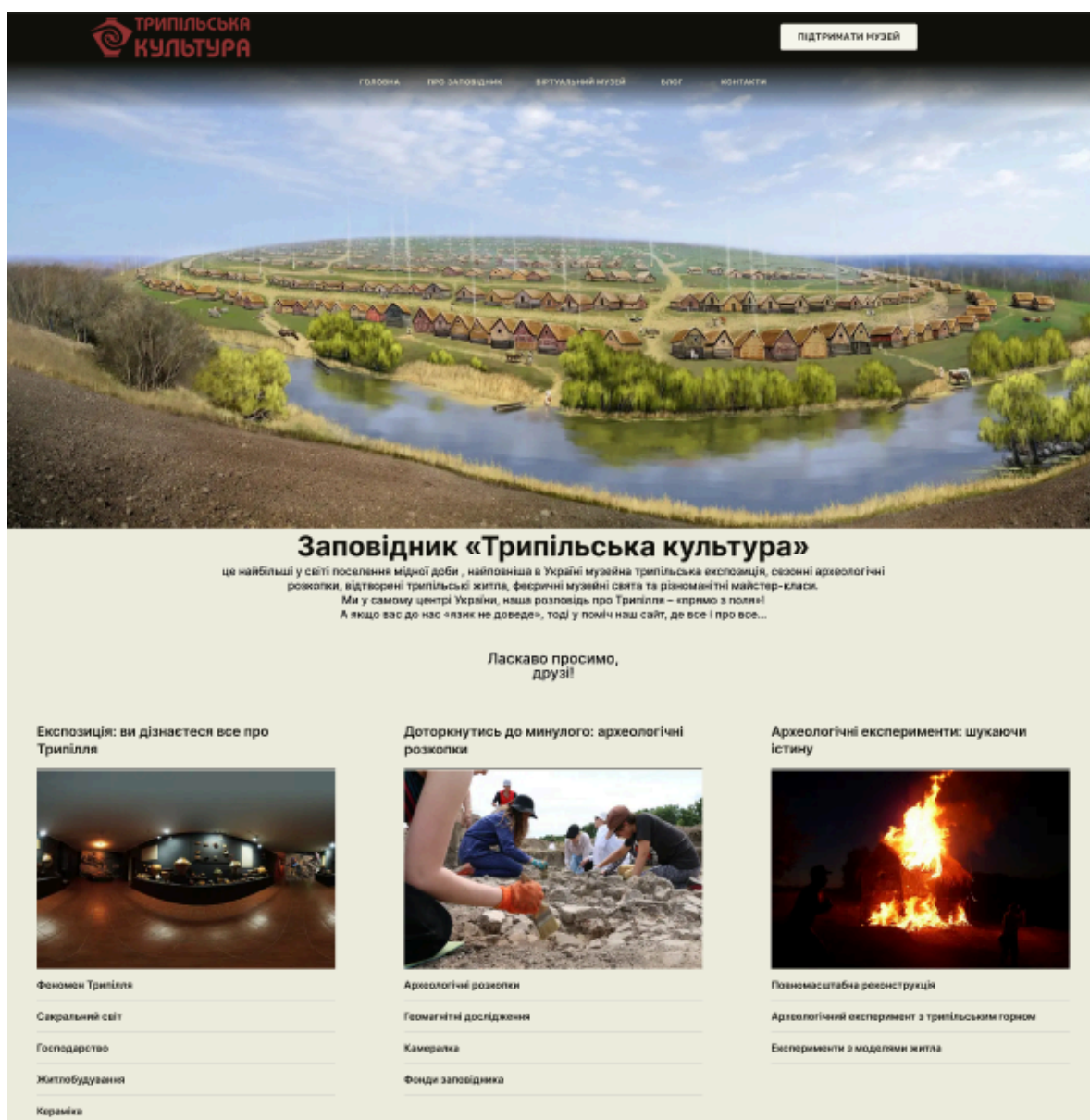


Рисунок 2.3 - кінцевий дизайн головної сторінки

На сторінці "Про заповідник" представлена важлива інформація про заповідник "Трипільська Культура". Описується історія створення заповідника за постановою Кабінету Міністрів України та його основні відомості. Розповідається про унікальність музею як найбільшого за площею поселення світового енеоліту, яке є єдиним в Європі заповідником однієї археологічної культури. Описується співпраця з археологічними експедиціями та розкопками на території мега-поселень, включаючи залучення студентів та обробку знахідок. Описується робота над створенням музейного комплексу в центрі села Легедзине, його експозиція та реконструйовані об'єкти. Описуються традиційні музейні фестивалі та свята, що проводяться заповідником, які спрямовані на популяризацію трипільської культури. Згадується робота по відцифруванню археологічної колекції та введенню її в науковий обіг. Ця сторінка має на меті розкрити відвідувачам унікальність та значення заповідника "Трипільська Культура", а також ознайомити їх з різними аспектами його роботи та діяльності.

Сторінка "Каталог артефактів" у веб-додатку музею "Трипільська Культура" представляє інформацію про артефакти, зібрані та зберігаються в музейному фонді. Вона надає можливість відвідувачам переглядати артефакти різних часів та епох, ознайомлюватися з їхніми описами, історією та значенням. Каталог допомагає підняти свідомість про багату культурну спадщину трипільської цивілізації та привертає увагу до цінностей археологічного дослідження.

На сторінці "Блог" у веб-додатку музею "Трипільська Культура" представлені статті, які розподілені за чотирма категоріями: Експозиція, Експерименти, Розкопки, Активності. Кожна категорія створена для групування статей з відповідними тематиками та розділення інформації для відвідувачів. Сторінка "Блог" надає можливість користувачам ознайомитися з цікавими матеріалами, які стосуються діяльності музею, його експозиції, проведення розкопок, експериментів та активностей. Кожна стаття має власний заголовок, зображення та опис, що дозволяє відвідувачам з легкістю знайти інформацію, яка

їх цікавить. На рисунку 2.4 зображено кінцевий дизайн сторінки блогу веб-додатку, розроблений у Figma.

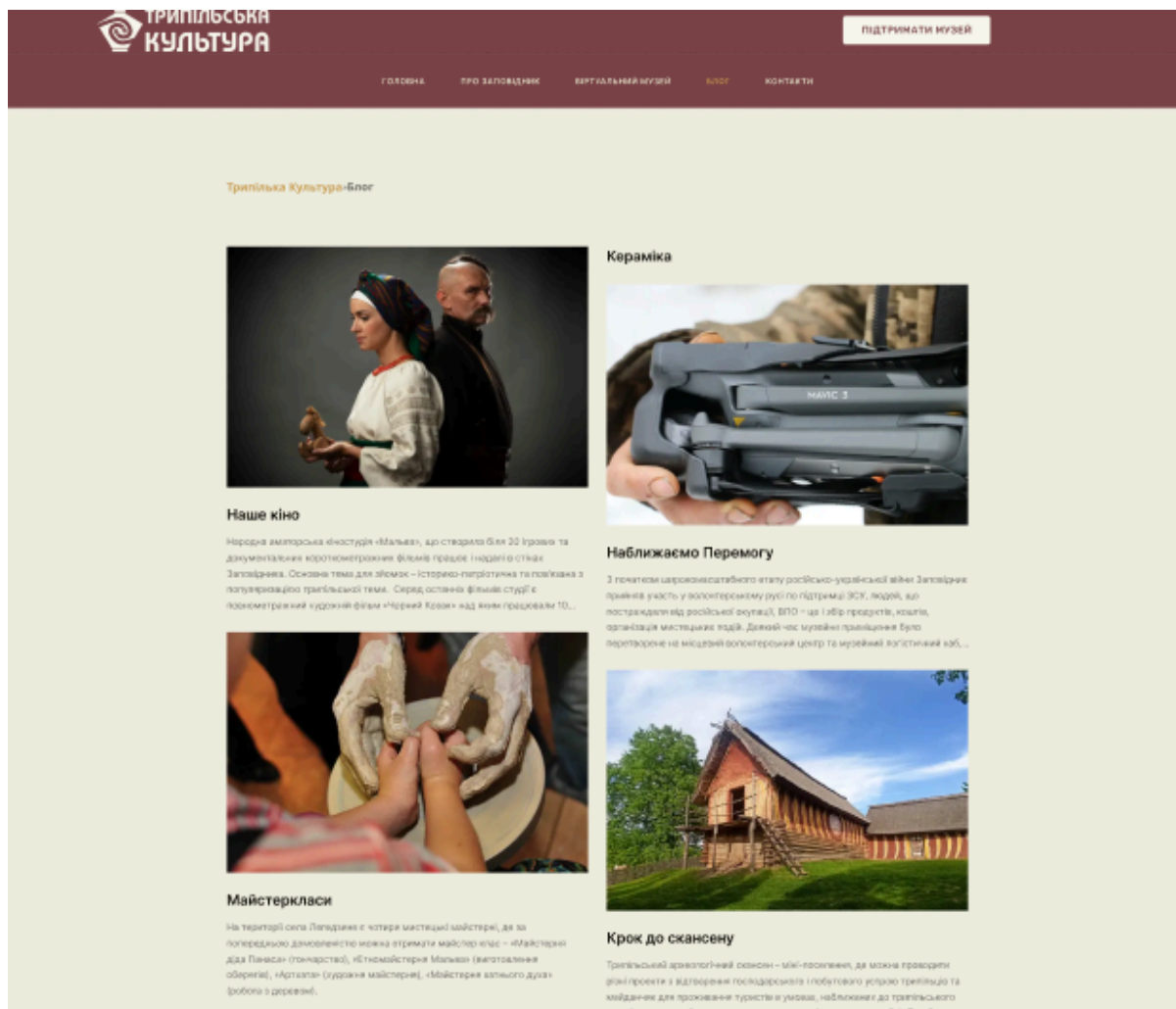


Рисунок 2.4 - кінцевий дизайн сторінки "Блог"

На сторінці "Контакти" веб-додатку музею "Трипільська Культура" розміщена інформація про контактні дані та режим роботи закладу. Користувачі можуть знайти адресу музею, номери телефонів для зв'язку, а також інші способи зв'язку, наприклад, електронну пошту або форму зворотного зв'язку. Також на цій сторінці може бути наведена інформація про графік роботи музею, часи проведення екскурсій або інших подій. Крім того, можуть бути вказані інструкції, як доїхати до музею за допомогою громадського транспорту або автомобілем. В цілому, сторінка "Контакти" надає відвідувачам всю необхідну інформацію для зручного зв'язку та відвідування музею.

На сторінці "Події" веб-додатку музею "Трипільська Культура" розміщена інформація про поточні та майбутні події та івенти, які відбуваються або плануються в музеї. Користувачі можуть дізнатися про проведення виставок, концертів, лекцій, майстер-класів та інших заходів, які організовані в музеї. Кожна подія може мати свій заголовок, опис, дату та час проведення, а також місце проведення. Користувачі можуть також мати можливість зареєструватися на подію або дізнатися додаткову інформацію, натиснувши на посилання для переходу до детальної сторінки події. В цілому, сторінка "Події" надає відвідувачам актуальну інформацію про заходи, які відбуваються в музеї, та дозволяє їм бути в курсі всіх цікавих подій.

2.3 Налаштування теми та головних плагінів

Першим кроком реалізації проекту було завантаження та встановлення системи Wordpress на хостинг. Замовник на той момент вже мав працюючий домен хостингу The Host, тому залишалося перевірити наявність підтримки PHP версії 7.4+, для коректної роботи необхідних плагінів, та створити нову базу даних для веб-додатку WordPress у панелі управління хостингу. Після закінчення роботи та здачі проекту, всі дані (паролі та інше) передаються замовнику. Далі, використовуючи файловий менеджер хостингу, файли WordPress були переміщені на сервер у каталог, який є URL-адресою веб-додатку. Відкривши веб-браузер і перейдіть на URL-адресу, де були розміщені файли WordPress, було відкрито сторінку встановлення WordPress. На сторінці встановлення біло введено дані про базу даних, яка була створена раніше. На цьому етапі ви можна вказати назву сайту, ім'я користувача та пароль для доступу до панелі адміністратора. Після заповнення всіх необхідних полів, було отримано посилання для входу в адмінпанель веб-додатку.

Наступним кроком після встановлення WordPress на хостинг буде встановлення та налаштування теми Ozeum. Після отримання теми від замовника, файли теми було розпаковано та встановлено через адмінпанель WordPress. Після успішної установки я активував тему та встановив та активував плагін ThemeREX Addons,

ввівши код покупки та особисту інформацію для активації теми. Я також обрав потрібний дизайн теми (сітку) та активував його. Після активації теми та встановлення плагіну ThemeREX Addons, мені було запропоновано встановити додаткові плагіни, які входять до складу теми. Щоб це зробити, я перейшов у розділ Appearance > Install Plugins. Там я вибрав необхідні плагіни зі списку та встановив їх, обираючи опцію Install та натиснув кнопку Apply. Після встановлення плагінів я активував їх, вибравши їх у списку і обравши опцію Activate, після чого натиснув кнопку Apply. Оскільки розробка веб-додатку музею "Трипільська Культура" вимагала значних змін оригінальної теми, було вирішено використовувати Child Theme для внесення власних змін до теми без ризику втрати цих змін при оновленні батьківської теми. Child Theme - це тема в WordPress, яка успадковує функціональність та стилі від батьківської теми, але дозволяє змінювати або доповнювати їх безпечним способом. Воно використовується для внесення власних змін до теми без ризику втрати цих змін при оновленні батьківської теми. Зміни, внесені до Child Theme, не будуть втрачені при оновленні батьківської теми. Це дозволяє вам вносити зміни в код та стилі безпечно. Child Theme дозволяє легко керувати змінами, оскільки всі вони знаходяться в окремих файлах, які можна керувати незалежно від батьківської теми, та змінювати або доповнювати CSS стилі та PHP функціонал батьківської теми, не змінюючи саму тему. В цілому, використання Child Theme - це рекомендований підхід для будь-яких значних змін у дизайні або функціональності теми в WordPress. Коли я розробляв веб-додаток для музею "Трипільська Культура", вирішив скористатися Child Theme для безпечних змін до оригінальної теми. Почав я з того, що створив нову теку у директорії тем під назвою "tripillian-culture-child". В цій текі створив файли, що мали бути модифіковані, включаючи functions.php. У functions.php моєї дитячої теми я відтворив або зареєстрував нові функції, які були змінені або додані до оригінальної теми. Я скопіював та зареєстрував необхідні функції з functions.php батьківської теми, щоб забезпечити правильне функціонування веб-додатку. Крім того, я скопіював всі файли з батьківської теми у дитячу, за винятком файлів, які

містяться у теках `/includes`, `/plugins`, `/theme-options`, `/theme-specifics`. Для файлів у цих теках я реалізував зміни через додавання або зареєстрування необхідних функцій у `functions.php` дитячої теми. Таким чином, застосування Child Theme дозволило мені безпечно та ефективно вносити зміни до веб-додатку, не ризикуючи втратою цих змін при оновленні батьківської теми.

Коли я приступив до кастомізації теми для веб-додатку музею "Трипільська Культура", я спочатку відкрив панель керування за допомогою Appearance (WP Dashboard) > Customize. Це дозволило мені гнучко керувати зовнішнім виглядом та поведінкою всього веб-сайту, а також окремих типів записів, сторінок та категорій. Найбільш важливою метою кастомізації було забезпечення відповідності вигляду продукту до розроблених раніше прототипів у Figma. Важливою особливістю нашої теми є система спадкоємності для налаштувань "Theme Options". Це означає, що індивідуальні налаштування кожної сторінки можуть перевизначити глобальні налаштування, встановлені через Customizer Panel. Налаштування "Blog", "WooCommerce", "Plugins settings" і інші групи також можуть перевизначати типові/загальні налаштування, доступні через Customizer. Крім того, я міг бачити зміни на фронтенді в реальному часі після внесення змін - коли потрібний параметр був вибраний, без необхідності постійного перемикання між адмін панеллю та фронтендом. Я завершив кастомізацію, натиснувши на кнопку "Publish" у верхньому меню бічної панелі Customizer. Цей процес кастомізації дозволив мені швидко та ефективно адаптувати вигляд та функціональність теми Ozeum для потреб музею "Трипільська Культура". На рисунку 2.5 зображено панель кастомізації веб-додатку "Трипільська Культура" у Wordpress



Рисунок 2.5 - панель кастомізації веб-додатку “Трипільська Культура”

Після завершення загальних налаштувань кастомізації веб-додатку музею "Трипільська Культура", я розпочав процес додавання інформації, віджетів та іншого контенту на сторінках додатку згідно розробленим раніше дизайном. Для цього процесу я використовував плагін Elementor, а також Панель кастомізації для детального налаштування. Elementor - це потужний інструмент для розробки веб-сайтів на платформі WordPress. Він дозволяє створювати стильні та функціональні сторінки без необхідності в програмуванні. За допомогою Elementor можна легко розміщувати на сторінках текст, зображення, кнопки, відео, форми, віджети і багато іншого, просто перетягуючи їх у зручне місце. Цей інструмент дозволяє швидко створювати сторінки в режимі реального часу, бачачи результати негайно. Використання Elementor у сполученні з Панеллю кастомізації дозволило мені точно відтворити розроблені раніше дизайни та надати їм життя. Я міг легко розмістити кожен елемент контенту на відповідній сторінці, використовуючи інтуїтивний інтерфейс Elementor. Після цього я використовував Панель кастомізації для налаштування загальних параметрів теми, колірної схеми, шрифтів та інших деталей, щоб забезпечити відповідність дизайну та бренду музею. Такий підхід дозволяв ефективно працювати над проектом та швидко досягати бажаних результатів. Після завершення кастомізації та змін зовнішнього

вигляду веб-додатку, я повернувся до налаштування плагінів, щоб забезпечити їх правильне функціонування та інтеграцію з сайтом.

Плагіни - це невеликі програмні додатки, які розширюють функціональність сайту на платформі WordPress. Вони дозволяють додавати різноманітні функції та можливості до сайту, такі як контактні форми, слайдшоу, статистика відвідуваності, інтеграція з соціальними мережами та багато іншого.

Одним з перших встановлених плагінів був LightStart – Maintenance Mode, Coming Soon and Landing Page Builder. Цей плагін був одним з перших, які я встановив для веб-додатку "Трипільська Культура". Він надавав можливість створити сторінку обслуговування для веб-додатку, щоб повідомити відвідувачів про те, що сайт тимчасово недоступний. Це дозволило забезпечити зручність для користувачів, які відвідували сайт під час розробки. Крім того, плагін дозволяв створювати сторінку "Coming Soon" для нового веб-сайту або сторінку "Landing Page" для існуючого сайту, щоб зацікавити відвідувачів майбутнім релізом. Цей плагін був повністю налаштовуваний, дозволяючи змінювати кольори, текст та фон сторінки відповідно до дизайну сайту. Він також пропонував SEO-опції, що допомагали забезпечити оптимальне позиціонування сайту у пошукових системах. Загалом, використання цього плагіна дозволило створити ефективну та зручну сторінку обслуговування для веб-додатку "Трипільська Культура", забезпечивши при цьому зручність для користувачів та ефективний збір контактних даних.

Наступними важливими плагінами були Менеджер файлів WP та WPvivid Backup Plugin. Плагін Менеджер файлів WP був надзвичайно важливим для мене, оскільки він дозволяв керувати файлами в моєму веб-додатку "Трипільська Культура" безпосередньо через панель адміністратора WordPress. Завдяки цьому плагіну я міг легко завантажувати, видаляти, переміщувати та редагувати файли на своєму сервері без необхідності виходити за межі WordPress. Це робило процес керування файлами більш зручним і ефективним, зекономивши час та зменшивши можливість помилок. Щодо WPvivid Backup Plugin, цей плагін виявився надзвичайно важливим для забезпечення безпеки та захисту веб-додатку. Він

надавав можливість регулярного створення резервних копій всього сайту, включаючи базу даних, файли та інші важливі елементи. Це було надзвичайно важливо для запобігання втрати даних у разі виникнення проблем або атак на сайт. Крім того, цей плагін дозволяв легко відновлювати сайт з резервних копій у разі необхідності, що забезпечувало надійність та стабільність веб-додатку.

Іншим важливим для безпеки веб-додатку плагіном був Wordfence Security. Плагін був важливою складовою для забезпечення безпеки та захисту веб-додатку "Трипільська Культура". Він надавав широкий спектр функцій та можливостей для виявлення та запобігання потенційним загрозам для безпеки сайту. Основні функції плагіну Wordfence Security включають виявлення шкідливого коду, захист від вторгнень, відстеження недійсних входів, захист від DDoS-атак та систему оповіщень. Плагін сканує веб-сайт на наявність шкідливого або підозрілого коду, запобігає неповноважному доступу до адміністративних областей, відстежує невдалі спроби входу та блокує потенційно шкідливі IP-адреси. Також надає захист від розподілених атак на відмову в обслуговуванні і надсилає сповіщення про потенційні загрози на електронну пошту або мобільний пристрій, що дозволяє оперативно реагувати на проблеми безпеки. Використання Wordfence Security було важливим для забезпечення безпеки та стабільності мого веб-додатку, дозволяючи запобігати потенційним атакам та зберігати даний веб-сайт у безпечному стані.

Плагін WP Super Cache - це інструмент, який я використовував для підвищення швидкості завантаження веб-сайту. Він дозволяє створювати статичні HTML-файли з динамічних сторінок WordPress, що значно зменшує час заливо корисно для веб-сайтів з великим обсягом відвідувань, оскільки воно рвантаження сторінок для відвідувачів. Коли відвідувач запитує сторінку, плагін перевіряє, чи є вже збережена копія цієї сторінки. Якщо так, то він віддає статичний HTML-файл, що дозволяє значно прискорити завантаження. Це особобить їх більш ефективними і сприяє кращому користувацькому досвіду.

Для підтримки активності соціальних мереж музею "Трипільська Культура" і залучення більшої аудиторії я вирішив використати плагін Revive Old Posts. Цей плагін дозволяє автоматично ділитися старими записами з блогу на різних платформах соціальних мереж, таких як Facebook, Twitter, LinkedIn та інших. Після встановлення та активації плагіну Revive Old Posts я налаштував його, щоб він автоматично публікував посилання на старі записи блогу у наших соціальних мережах. Плагін має дуже зручний інтерфейс, який дозволяє вибрати, які типи постів мають бути розміщені, як часто це має відбуватися, а також додати хештеги та інші параметри для кожного посту. Основна мета використання цього плагіну полягає у тому, щоб постійно підтримувати інтерес до наших матеріалів та залучати нових відвідувачів на сайт. Завдяки Revive Old Posts, я зміг оживити старі, але все ще актуальні записи, які могли залишитися непоміченими новими підписниками в соціальних мережах. Це також допомогло збільшити трафік на веб-додаток, оскільки кожен публікаційний пост містить посилання, що веде до нашого сайту. Плагін дозволяє задавати інтервали між публікаціями, що допомогло уникнути спаму та забезпечило регулярний потік контенту в соціальних мережах. Крім того, я налаштував плагін таким чином, щоб виключити деякі категорії постів, які можуть бути менш цікавими для нашої аудиторії. Використання Revive Old Posts значно полегшило мою роботу з управління соціальними мережами та допомогло забезпечити постійну присутність музею "Трипільська Культура" в інформаційному полі, залучаючи більше відвідувачів і підтримуючи інтерес до нашого контенту.

Плагін Contact Form 7 став невід'ємною частиною функціоналу веб-додатку музею "Трипільська Культура". Його використання дозволило легко створювати та керувати контактними формами, що є важливим для забезпечення ефективного зв'язку з відвідувачами сайту. Після встановлення та активації Contact Form 7, я швидко зрозумів, наскільки гнучким і простим у використанні є цей плагін. Я створив кілька контактних форм, кожна з яких призначена для різних цілей. Наприклад, одна форма використовується для загальних запитів відвідувачів, інша - для бронювання екскурсій, а третя - для зворотного зв'язку щодо виставок і

подій. Процес налаштування форми був дуже інтуїтивно зрозумілим. У редакторі Contact Form 7 я додав необхідні поля, такі як ім'я, електронна пошта, тема повідомлення та саме повідомлення. Я також налаштував перевірку введених даних, щоб переконатися, що всі необхідні поля заповнені правильно. Однією з найкорисніших функцій Contact Form 7 є підтримка Аях-подання форм. Це дозволило забезпечити плавне та безперебійне відправлення форм без необхідності перезавантаження сторінки. Така функціональність значно покращила користувацький досвід, роблячи процес взаємодії з сайтом більш зручним і швидким. Загалом, використання Contact Form 7 значно полегшило процес створення та управління контактними формами на нашому сайті. Це дозволило нам ефективно збирати зворотний зв'язок, обробляти запити відвідувачів та покращити загальну комунікацію між музеєм та його аудиторією. На рисунку 2.6 зображено контактна форма на сторінці “Контакти”.

Зв'яжіться з нами

Ім'я

Електронна пошта

Повідомлення

☐ Я ЗГОДЕН, ЩО МОЇ НАДАНІ ДАНІ ЗБИРАЮТЬСЯ ТА ЗБЕРІГАЮТЬСЯ.

НАДІСЛАТИ ПОВІДОМЛЕННЯ

Рисунок 2.6 - Контактна форма на сторінці “Контакти”

Іншим важливим кроком було забезпечення надійної доставки електронних листів. Для цього я використав плагін WP-Mail-SMTP. Цей плагін дозволяє переналаштувати функцію `wp_mail()` в WordPress для використання SMTP-сервера, що значно покращує доставку листів та зменшує ймовірність

потрапляння їх до спаму. Оскільки адміністрація музею використовує Ukr.net як основну пошту, мені довелося налаштовувати плагін специфічно під цей сервіс. Я ввів необхідні SMTP-налаштування, включаючи SMTP-сервер Ukr.net, порт, метод шифрування, а також облікові дані електронної пошти адміністрації. Після цього я протестував налаштування, щоб впевнитися, що всі листи надсилаються і доставляються коректно.

Це рішення було важливим для забезпечення надійного зв'язку між музеєм та відвідувачами, зокрема для отримання запитів, зворотного зв'язку та інших важливих повідомлень через контактні форми на сайті. Завдяки WP-Mail-SMTP ми змогли уникнути проблем з недоставленими листами та забезпечити безперебійну комунікацію.

Після завершення основних етапів налаштування веб-додатку музею "Трипільська Культура", я вирішив оптимізувати його продуктивність, встановивши плагін Autoptimize. Цей плагін використовується для оптимізації швидкості завантаження сайту шляхом зменшення та об'єднання CSS, JavaScript та HTML файлів. Autoptimize дозволяє мінімізувати розмір файлів та кількість запитів до сервера, що значно покращує швидкість завантаження сторінок. Плагін також включає можливість відкладеного завантаження JavaScript, що означає, що скрипти будуть завантажуватися лише тоді, коли це буде необхідно, тим самим знижуючи навантаження на сервер при першому завантаженні сторінки. Після встановлення та активації плагіну я зайшов у його налаштування та обрав опції для оптимізації CSS, JavaScript та HTML. Також налаштував відкладене завантаження JavaScript і включив злиття CSS та JavaScript файлів. Це дозволило суттєво зменшити час завантаження сторінок та покращити користувацький досвід для відвідувачів сайту. Використання плагіну Autoptimize стало важливим кроком для забезпечення швидкої та ефективної роботи веб-додатку, особливо враховуючи великий обсяг вмісту та багатофункціональні сторінки, які були реалізовані в рамках проекту музею.

В підсумку, під час розробки веб-додатку музею "Трипільська Культура" було використано низку плагінів для покращення функціональності та роботи сайту. Contact Form 7 дозволив створювати гнучкі контактні форми, а Cookie Information забезпечив відповідність сайту вимогам GDPR. Slider Revolution додав динамічні слайдери, покращуючи візуальну привабливість сторінок. WP-Mail-SMTP, налаштований для Ukr.net, забезпечив надійну доставку електронних листів. LightStart створив сторінки технічного обслуговування та "скоро відкриття". WP Super Cache оптимізував швидкість завантаження, створюючи статичні HTML-файли, а Wordfence Security захистив сайт від атак. WPvivid Backup Plugin гарантував збереження даних через регулярні резервні копії. WP File Manager спростив роботу з файлами через адмін-панель, а Elementor став ключовим інструментом для створення сторінок відповідно до розроблених прототипів. Завдяки цим плагінам веб-додаток став зручним, безпечним і функціональним для відвідувачів та адміністрації музею.

2.4Робота з зображеннями

Проблема роботи з зображеннями у розробці веб-додатків часто полягає в балансуванні між якістю візуального контенту і продуктивністю сайту. Великі файли зображень можуть значно сповільнювати завантаження сторінок, особливо на мобільних пристроях з обмеженим швидкісним з'єднанням. Збільшення кількості зображень на сторінці може призвести до збільшення часу завантаження. Це може бути проблемою з точки зору продуктивності, особливо для сторінок з великою кількістю вмісту, таких як галереї або сторінки з продуктами. Неправильне використання форматів файлів або низька якість стискання можуть призвести до великих розмірів файлів, що може впливати на швидкість завантаження та ефективність передачі даних. Зображення повинні бути оптимізовані для різних розмірів екрану, щоб забезпечити адаптивність веб-додатку для різних пристроїв і роздільних здатностей екрану. Використання кешування зображень допомагає зменшити час завантаження сторінок для користувачів, які повторно відвідують сайт.

Використання плагіну оптимізації зображень від Optimole виявилось надзвичайно важливим для веб-додатку музею "Трипільська Культура", особливо через велику кількість зображень на сайті. Плагін оптимізації зображень від Optimole пропонує ряд корисних функцій, які спрощують оптимізацію та управління зображеннями на веб-сайті. Він забезпечує автоматичну оптимізацію зображень без втрати якості, використання Content Delivery Network (CDN) для швидкої доставки зображень користувачам з усього світу, автоматичне змінення розміру зображень відповідно до розмірів екрана користувача, техніку лінійного завантаження зображень для зменшення часу завантаження сторінок та використання пропускну здатності сервера, керування рівнем якості оптимізації та надання детальної аналітики використання зображень. Ці функції дозволяють поліпшити продуктивність та швидкість завантаження веб-сайту, ефективно використовуючи пропускну здатність сервера.

Підсумовуючи, використання Послуги оптимізації зображень від Optimole значно покращило роботу з зображеннями на веб-додатку музею "Трипільська Культура". Завдяки автоматичній оптимізації зображень без втрати якості, використанню CDN для швидкої доставки, адаптивному змінненню розміру зображень, техніці лінійного завантаження та детальній аналітиці, було досягнуто зменшення часу завантаження сторінок та підвищення загальної продуктивності сайту. Це особливо важливо для веб-додатку музею, який містить велику кількість зображень артефактів, що потребують ефективного управління для забезпечення позитивного користувацького досвіду.

2.5 Розробка каталогу для веб-додатку музею "Трипільська Культура"

Під час розробки каталогу для веб-додатку музею "Трипільська Культура" на платформі WordPress, я ретельно дослідив і вибрав технології, які найкраще відповідали потребам проекту. Основна мета полягала в створенні зручного та функціонального каталогу, який міг би ефективно представити колекцію музею. Першим кроком було вивчення різних плагінів та технологій, доступних для

WordPress, які могли б допомогти в реалізації каталогу. Основні критерії вибору включали легкість використання, гнучкість налаштувань, можливості розширення та сумісність з існуючою темою Ozeum.

WooCommerce є одним з найпопулярніших плагінів для створення інтернет-магазинів на WordPress. Хоча цей плагін зазвичай використовується для електронної комерції, його гнучкість дозволяє налаштувати його під каталог музейних експонатів. WooCommerce пропонує широкий спектр розширень і тем, що дозволяє створити індивідуальний дизайн і функціонал для каталогу. Його основний функціонал включає створення категорій, підкатегорій та окремих елементів каталогу, додавання детальних описів, зображень, відео та інших метаданих. WooCommerce також дозволяє налаштувати фільтри та сортування експонатів, що робить процес пошуку та перегляду зручним для користувачів.

WP Catalog є спеціалізованим плагіном для створення каталогів на WordPress. Він пропонує зручні інструменти для створення та управління великою кількістю елементів каталогу. За допомогою цього плагіну можна легко додавати нові категорії та підкатегорії, а також детально описувати кожен експонат. WP Catalog забезпечує можливість додавання зображень, відео та інших метаданих, що робить каталог інформативним і привабливим для користувачів. Цей плагін також підтримує різні типи контенту та дозволяє налаштовувати відображення елементів каталогу відповідно до вимог проекту.

Toolset Types є потужним інструментом для створення власних типів записів і таксономій на WordPress. Цей плагін надає гнучкість у налаштуванні полів і відображенні контенту, що дозволяє створити унікальний каталог відповідно до потреб музею. За допомогою Toolset Types можна легко створювати та налаштовувати нові типи записів, такі як експонати, категорії та підкатегорії, а також додавати індивідуальні поля для кожного елемента каталогу. Цей плагін дозволяє створювати складні структури даних і забезпечує зручний інтерфейс для управління контентом.

Вибір технологій для створення каталогу в веб-додатку музею "Трипільська Культура" забезпечив ефективне представлення колекції музею. Використання WooCommerce дозволило створити зручний і функціональний каталог, який відповідає всім вимогам проекту, забезпечуючи при цьому гнучкість і розширюваність для майбутніх оновлень та вдосконалень.

Після встановлення та активації плагіну WooCommerce автоматично запустився майстер налаштування. Після активації WooCommerce запустився майстер налаштування. Я пропустив кроки, пов'язані з платежами і доставкою, оскільки каталог не є магазином. Замість цього я сконцентрувався на налаштуванні основних параметрів, таких як локація музею і основна валюта. WooCommerce автоматично створює кілька основних сторінок, але я залишив лише ті, які були потрібні для нашого каталогу, а саме "Магазин". Ця сторінка буде використовуватися для відображення всіх експонатів музею. Перейшовши до налаштувань категорій, я створив основні категорії експонатів: Археологічні знахідки, Кераміка, Інструменти та зброя. Для кожної категорії я додав опис і зображення, щоб допомогти користувачам легше орієнтуватися в каталозі. Далі я почав додавати експонати. Для кожного експоната я заповнив такі поля: назва експоната, детальний опис з історичними фактами та інформацією про знахідку, короткий опис для попереднього перегляду, головне зображення експоната, додаткові зображення експоната, категорії та мітки для покращення навігації. Оскільки експонати музею мають унікальні характеристики, я додав спеціальні атрибути в розділі "Властивості продукту", такі як: період знахідки, матеріал, та стан. Щоб забезпечити естетичну відповідність каталогу розробленому раніше дизайну у Figma, я використав спеціальні налаштування теми та плагін Elementor. За допомогою Elementor я налаштував сторінки продуктів так, щоб вони виглядали привабливо та інтуїтивно зрозуміло для користувачів. Я налаштував вигляд сторінок категорій та окремих експонатів, додавши необхідні секції та віджети. Щоб користувачам було легше знайти потрібні експонати, я додав віджети фільтрів. Перейшовши до "Зовнішній вигляд" > "Віджети", я додав фільтр по категоріях та фільтр по властивостях продукту. Щоб експонати музею були

добре індексовані пошуковими системами, я використав плагін Yoast SEO. Для кожного продукту я налаштував метаописи, ключові слова та заголовки сторінок. Це допомогло покращити видимість каталогу в пошукових системах. Після завершення налаштувань я ретельно протестував роботу каталогу, перевіряючи всі посилання, фільтри та сортування працюють коректно. Також я оптимізував швидкість завантаження сторінок за допомогою плагінів, таких як WP Super Cache та Autoptimize.

Налаштувавши WooCommerce для веб-додатку музею "Трипільська Культура", я створив інформативний, структурований і зручний каталог експонатів, що забезпечує користувачам легкий доступ до цінної інформації про артефакти музею.

3 НАЛАШТУВАННЯ SEO ОПТИМІЗАЦІЇ ДЛЯ ВЕБ-ДОДАТКУ МУЗЕЮ “ТРИПІЛЬСЬКА КУЛЬТУРА”

3.1 Призначення та основні методи SEO оптимізації

SEO, або пошукова оптимізація, - це процес оптимізації веб-сайту для підвищення його видимості у пошукових системах, таких як Google, Bing або Yahoo, з метою залучення органічного трафіку. Для ефективної роботи з SEO важливо зосередитися на ключових аспектах: визначенні та використанні відповідних ключових слів у заголовках, метаописах, контенті та URL; створенні якісного, унікального і релевантного контенту, який надає корисну інформацію для користувачів; технічній оптимізації, що включає швидкість завантаження сайту, мобільну дружельність, безпеку (HTTPS), а також створення і оптимізацію файлів robots.txt і sitemap.xml; On-Page SEO, яке передбачає оптимізацію заголовків, мета описів, зображень (включаючи alt-теги), внутрішніх посилань і структури URL; Off-Page SEO, що включає створення зворотних посилань з авторитетних сайтів, взаємодію в соціальних медіа та створення профілю в інтернет-каталогах; а також регулярний аналіз і моніторинг, що дозволяє виявити ефективність стратегій і коригувати їх за потреби. SEO - це постійний процес, що вимагає часу і зусиль, і включає як технічні аспекти, так і креативну роботу з контентом, враховуючи постійні зміни в алгоритмах пошукових систем. На рисунку 3.1 зображено модель роботи пошукових систем.



Рисунок 3.1 - модель роботи пошукових систем

Ключові слова є основою успішної стратегії пошукової оптимізації (SEO). Вони представляють терміни та фрази, які користувачі вводять у пошукові системи, коли шукають інформацію, продукти чи послуги. Визначення та ефективне використання відповідних ключових слів допомагає підвищити видимість веб-сайту в пошукових системах і залучити більше цільового трафіку. Ефективне використання ключових слів сприяє покращенню SEO веб-сайту, підвищуючи його видимість у пошукових системах і залучаючи більше органічного трафіку. Однак важливо не перестаратися і уникати "ключового слова спаму" (overstuffing), оскільки це може призвести до штрафів з боку пошукових систем і негативно вплинути на рейтинг сайту. Таким чином, ключові слова є критичним елементом SEO-стратегії, який потребує ретельного дослідження та стратегічного використання для досягнення найкращих результатів.

On-Page SEO і Off-Page SEO є двома основними аспектами пошукової оптимізації, які допомагають підвищити видимість і рейтинг веб-сайту у пошукових системах. On-Page SEO включає оптимізацію елементів на самому сайті, що допомагає пошуковим системам зрозуміти та правильно індексувати

контент. Ключові компоненти On-Page SEO включають: заголовки, метаописи, вміст, зображення, URL-адреси та внутрішні посилання. Off-Page SEO зосереджується на підвищенні авторитету сайту за межами його власного контенту. Його сновні компоненти включають: зворотні посилання, соціальні медіа, відгуки на різних платформах, тощо. On-Page SEO та Off-Page SEO є взаємодоповнюючими стратегіями, які разом допомагають покращити видимість сайту у пошукових системах та залучити більше цільового трафіку.

WordPress є однією з найпопулярніших платформ для створення веб-сайтів, і він має вбудовані функції, які допомагають з SEO, а також підтримує різноманітні плагіни для подальшої оптимізації. WordPress генерує чистий HTML-код, який легко сканується пошуковими системами. Це допомагає поліпшити індексацію контенту. Він також дозволяє налаштовувати структуру URL-адрес, що робить їх більш зручними для користувачів і пошукових систем. Використання ключових слів у URL-адресах допомагає покращити ранжування. За допомогою плагінів, таких як Yoast SEO або All in One SEO Pack, можна легко налаштовувати мета-заголовки, мета-описи та ключові слова для кожної сторінки та посту. WordPress дозволяє додавати alt-теги до зображень, що допомагає пошуковим системам зрозуміти зміст зображень. Плагіни, такі як Smush або Optimole, автоматично оптимізують зображення для швидшого завантаження. Більшість сучасних тем WordPress є адаптивними, що означає, що вони автоматично налаштовуються під розмір екрану пристрою, забезпечуючи кращий досвід для мобільних користувачів. У WordPress також підтримується кешування та мінімізація файлів через плагіни, такі як W3 Total Cache або WP Super Cache, що допомагає прискорити завантаження сайту. Плагіни для соціальних медіа дозволяють легко додавати кнопки для поширення контенту, що може підвищити видимість і трафік на сайті. Плагіни, такі як Google XML Sitemaps або функціональність Yoast SEO, автоматично генерують карту сайту, яка допомагає пошуковим системам ефективніше індексувати сайт. WordPress дозволяє легко інтегрувати інструменти для аналітики, такі як Google Analytics, для відстеження трафіку і поведінки користувачів на сайті, що допомагає коригувати

SEO-стратегії. Такі плагіни, як Akismet, допомагають захистити сайт від спаму в коментарях, що може вплинути на SEO. Використання цих функцій і плагінів допомагає поліпшити SEO веб-сайту на WordPress, забезпечуючи його кращу видимість і рейтинг у пошукових системах.[18]

Використання SEO, особливо у поєднанні з платформою WordPress, є ключовим інструментом для підвищення видимості та ефективності веб-сайту. WordPress, завдяки своїй гнучкості і широкому вибору плагінів, таких як Yoast SEO і All in One SEO Pack, спрощує процес оптимізації контенту, забезпечуючи зручний доступ до важливих інструментів для роботи з ключовими словами, метаописами, заголовками і зображеннями. Використання On-Page SEO, як оптимізація внутрішніх елементів сайту, і Off-Page SEO, як побудова авторитету сайту за допомогою зворотних посилань і активності в соціальних медіа, допомагає значно підвищити рейтинг сайту в пошукових системах. Загалом, інтеграція SEO стратегій у WordPress дозволяє ефективно керувати видимістю сайту, залучати цільовий трафік і забезпечувати довготривале зростання та успіх веб-ресурсу.

3.2 Налаштування SEO оптимізації для веб-додатку

Робота з SEO при розробці веб-додатку на WordPress починається з декількох ключових кроків. Перш за все, необхідно провести дослідження ключових слів, щоб визначити, які терміни та фрази користувачі вводять у пошукові системи для знаходження контенту. На основі цього дослідження створюється стратегія ключових слів, яка включає їх використання в заголовках, метаописах, URL, контенті та зображеннях.

Після цього важливо встановити та налаштувати плагіни для SEO, такі як Yoast SEO або All in One SEO Pack. Ці інструменти допомагають оптимізувати контент, надаючи підказки та рекомендації для покращення SEO на кожній сторінці та в кожній публікації. Вони також дозволяють налаштовувати

XML-карти сайту, які допомагають пошуковим системам ефективніше індексувати сайт.

Налаштування SEO для веб-додатку музею "Трипільська Культура" було важливим етапом у розробці, адже ми прагнули забезпечити максимально ефективну видимість нашого сайту у пошукових системах. Я розпочав з пошук ключових слів для веб-додатку. Це був одним із найважливіших етапів SEO-оптимізації. Спершу я використовував інструменти для дослідження ключових слів, такі як Google Keyword Planner, Ahrefs, і SEMrush, щоб зібрати інформацію про популярні пошукові запити, пов'язані з нашою тематикою. Під час дослідження ключових слів я звертав увагу на такі аспекти як: кількість запитів за певне ключове слово та його релевантність. Для веб-додатку музею "Трипільська Культура" ключовими словами могли бути:

- "Трипільська культура"
- "історія Трипілля"
- "музей Трипілля"
- "Трипільська археологія"
- "давні цивілізації України"
- "експозиції музею"
- "наукові статті про Трипілля"
- "онлайн екскурсії Трипілля"
- "історичні артефакти Трипілля"

Ці ключові слова були обрані з урахуванням їхнього обсягу пошуку і релевантності для нашої аудиторії. Після збору ключових слів я інтегрував їх у контент веб-додатку, включаючи заголовки сторінок, метатеги, описи зображень та основний текст. Це дозволило нам покращити видимість нашого сайту у пошукових системах та залучити більше відвідувачів, зацікавлених в історії Трипільської культури.

Наступним етапом у налаштуванні SEO веб-додатку було встановлення плагіну Yoast SEO. Це один з найпотужніших інструментів для SEO, який допомагає оптимізувати кожну сторінку і публікацію. Після установки плагіну я перейшов до налаштувань, де ввів ключові слова для нашого контенту, створив XML-карти сайту і налаштував метатеги, зокрема заголовки та метаописи. На рисунку 3.2 зображено панель налаштування плагіну Yoast SEO.

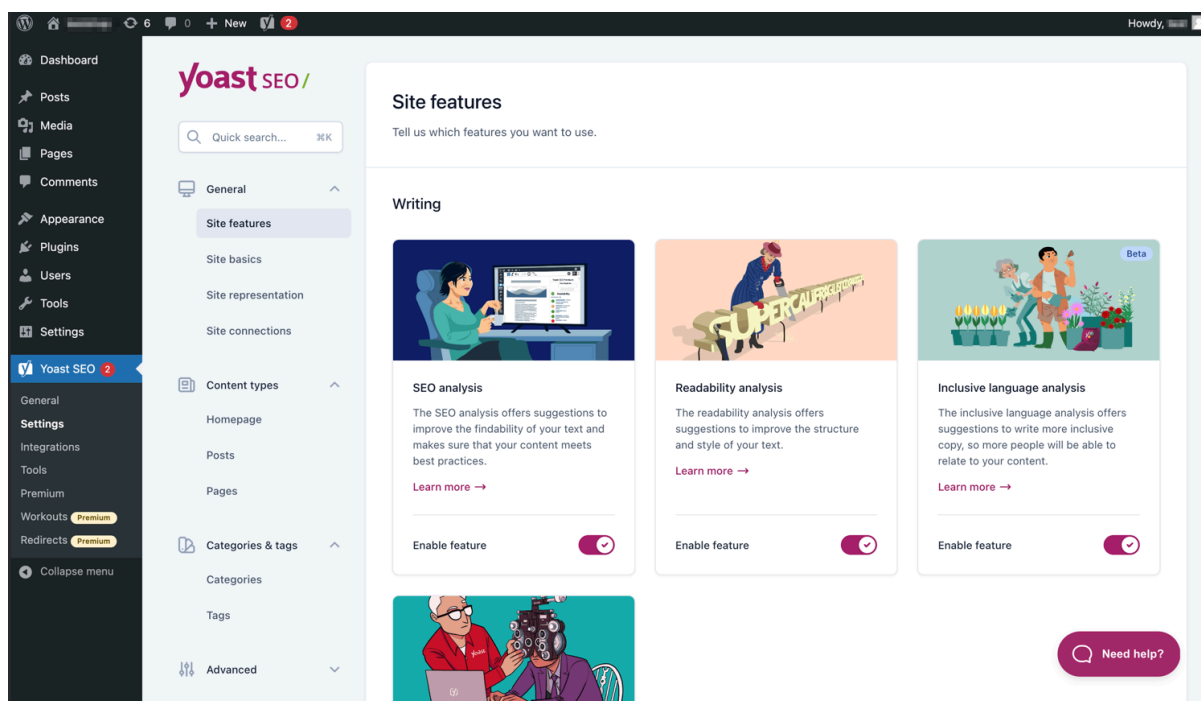


Рисунок 3.2 - панель налаштування плагіну Yoast SEO

Yoast SEO допоміг мені створити XML-карти сайту, які містять інформацію про всі сторінки веб-додатку. Ці карти важливі для пошукових систем, оскільки вони допомагають їм ефективніше індексувати сайт. Плагін автоматично генерує та оновлює XML-карти при додаванні або зміні контенту, забезпечуючи актуальність даних. Після цього я налаштував метатеги. В Yoast SEO є зручний інтерфейс для введення заголовків та метаописів для кожної сторінки сайту. Я намагався включити основні ключові слова в заголовки та метаописи, щоб покращити видимість сторінок у пошукових системах. Кожна сторінка отримала унікальні заголовки та метаописи, що відображали її зміст та відповідали запитам користувачів. Також Yoast SEO надавав корисні рекомендації і підказки для

покращення SEO, такі як оптимальна довжина тексту, використання ключових слів і внутрішніх посилань. [17]

Після цього я налаштував плагін Virtual Robots.txt, який дозволяє управляти файлами robots.txt безпосередньо з панелі адміністратора WordPress. Це дало можливість вказати пошуковим системам, які сторінки індексувати, а які – ні. Завдяки цьому плагіну ми змогли забезпечити, що приватні сторінки, такі як сторінки входу або сторінки адміністрування, не будуть проіндексовані. Також я встановив плагін Breadcrumb NavXT для створення навігаційних ланцюжків (breadcrumbs), які покращують структуру сайту і полегшують навігацію для користувачів. Це також позитивно впливає на SEO, оскільки пошукові системи використовують навігаційні ланцюжки для кращого розуміння структури сайту.

Завдяки цим крокам, веб-додаток музею "Трипільська Культура" отримав оптимізовані сторінки з правильними метатегами, актуальні XML-карти та налаштований файл robots.txt, що забезпечило краще ранжування в пошукових системах і зручність для користувачів.

ВИСНОВКИ

Створення веб-додатку для музею "Трипільська культура" було ретельно спланованим і включало кілька ключових етапів, починаючи від постановки задачі і вибору технологій до налаштування SEO-оптимізації. На етапі постановки задачі було визначено, що замовник не має достатньо ресурсів і технічного досвіду для підтримки сайту на складних front-end фреймворках. Тому було прийнято рішення використовувати систему керування вмістом (CMS), а саме WordPress, через її зручність, функціональність та широкий спектр можливостей. Крім того, було виявлено, що конструктори сайтів мають обмежений функціонал для вимог замовника. Серед декількох тем WordPress, найбільш підходящою виявилася тема Ozeum, яка відповідає потребам музею завдяки своїм численним функціям і можливостям налаштування. Розробка веб-додатку розпочалася з створення UI/UX дизайну за допомогою Figma. Після цього було розроблено функціонал сторінок веб-додатку, який включав налаштування теми та встановлення головних плагінів, таких як Elementor для створення сторінок, Contact Form 7 для обробки контактних форм, а також плагінів для безпеки і оптимізації. Важливою частиною була робота з зображеннями, оскільки веб-додаток включає велику кількість візуального контенту. Для цього використовувався плагін Optimole, який забезпечив оптимізацію зображень і покращення швидкості завантаження сторінок. Особливу увагу було приділено розробці каталогу артефактів музею, який не є інтернет-магазином, а слугує для демонстрації колекції музею. Для цього використовувався плагін WooCommerce, який налаштовувався відповідно до потреб музею. Завершальним етапом була налаштування SEO-оптимізації для підвищення видимості веб-додатку у пошукових системах. Було використано плагіни Yoast SEO, Virtual Robots.txt та Breadcrumb NavXT для налаштування метатегів, створення XML-карт сайту та покращення структури навігації.

В цілому, процес створення веб-додатку для музею "Трипільська культура" був комплексним і вимагав глибокого дослідження та ретельного налаштування різних компонентів. Завдяки використанню WordPress та відповідних плагінів

вдалося створити функціональний, елегантний і зручний у користуванні веб-додаток, який відповідає всім вимогам замовника та забезпечує чудовий онлайн-досвід для відвідувачів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Характеристики Web-технології та Web-дизайн: застосування мови HTML для створення електронних ресурсів // І. Л. Бородкіна, Г. О. Бородкін, Рік видання: 2020.
2. Створюємо динамічні веб-сайти за допомогою PHP, MySQL, JavaScript, CSS і HTML5 (6-те видання) // Робин Никсон, Рік видання: 2023.
3. Fundamentals of Web Development. 3rd Edition // Ренді Конноллі, Пікардо Хоар Рік видання: 2021.
4. Full Stack Web Development: The Comprehensive Guide // Філіп Акерманн Рік видання: 2023.
5. Wordpress official documentation // [Електронний ресурс]:
<https://wordpress.org/documentation/>
6. Joomla official documentation // [Електронний ресурс]:
https://docs.joomla.org/Main_Page
7. Drupal official documentation // [Електронний ресурс]:
<https://www.drupal.org/documentation>
8. React official documentation. Quick Start // [Електронний ресурс]:
<https://react.dev/learn>
9. Angular official documentation. What is Angular? // [Електронний ресурс]:
<https://angular.io/guide/what-is-angular>
10. Vue.js official documentation. Introduction // [Електронний ресурс]:
<https://vuejs.org/guide/introduction.html>
11. Wix Introduction 2024– Ultimate Guide for Beginners // [Електронний ресурс]:
<https://www.beehexa.com/blog/wix-review-2021-beehexa>
12. Weebly: Introduction // [Електронний ресурс]:
<https://www.domain.com/help/article/weebly-introduction>
13. Огляд конструктора сайтів Squarespace // [Електронний ресурс]:
<https://adwservice.com.ua/uk/ogljad-squarespace>

14. Express/Node introduction // [Електронний ресурс]:

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction

15. Django introduction // [Електронний ресурс]:

<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>

16. Figma Resource library. Design basics // [Електронний ресурс]:

<https://www.figma.com/resource-library/design-basics/>

17. SEO basics // [Електронний ресурс]:

<https://yoast.com/tag/seo-basics/>

18. The Beginner's Guide to SEO: How to Optimize Your Website, Rank Higher on Google and Drive More Traffic // Jessica Ainsworth, Рік видання: 2021.

19. Documentation for OZEUM // [Електронний ресурс]:

<https://ozeum.themerex.net/doc/>

ДОДАТОК А

```

<!DOCTYPE html>
<html lang="ru-RU" class="
  scheme_default
  js csstransitions">
  <head> </head>
  <body class="home page-template-default page page-id-69 page-parent wp-custom-logo theme-ozeum ua_chrome preloader frontpage woocommerce-js tri
skin_default body_tag scheme_default blog_mode_front body_style_wide is_stream blog_style_masonry_2 sidebar_hide expand_content remove_margins
ddons_present header_type_custom header_style_header-custom-5371 header_position_over menu_side_hide elementor-default elementor-kit-4719 elem
page elementor-page-69 sc_layouts_row_fixed_initied e--ua-blink e--ua-chrome e--ua-webkit mobile_layout" data-elementor-device-mode="mobile">
    <div id="page_preloader" style="opacity: 0; display: none;"></div>
    <div class="body_wrap">
    <script>
    <a href="#" class="trx_addons_scroll_to_top trx_addons_icon-up initied" title="Scroll to top">
    <script>
    <div class="trx_popup has-cache bottomleft " data-delay="1" data-refresh-interval="1">
    <link href="https://fonts.googleapis.com/css?family=Roboto:400&display=swap" rel="stylesheet" property="stylesheet" media="all" type="text/css">
    <script>
    <link property="stylesheet" rel="stylesheet" id="google-fonts-2-css" href=
    "https://fonts.googleapis.com/css?family=Rubik%3A100%2C100italic%2C200%2_2C800%2C800italic%2C900%2C900italic&display=swap&subset=cyrillic&ver=
    " type="text/css" media="all">
    <script type="text/javascript" id="trx_addons-js-extra">
    <script type="text/javascript" id="ozeum-init-js-extra">
    <script id="rs-initialisation-scripts">
    <script defer src="http://tripillia-reserve.com/wp-content/cache/autoptimize/js/autoptimize_3516798...js">
    <span id="elementor-device-mode" class="elementor-screen-only">
    </body>
  </html>

```

```

<!DOCTYPE html>
<html lang="ru-RU" class="
  scheme_default
  js csstransitions">
  <head> </head>
  <body class="page-template-default page page-id-77 wp-custom-logo theme-ozeum ua_chrome preloader woocommerce-js tribe-js skin_default body_tag scheme_de
fault blog_mode_page body_style_wide is_single sidebar_hide expand_content remove_margins trx_addons_present header_type_custom header_style_header-custo
m-5436 header_position_default menu_side_hide elementor-default elementor-kit-4719 elementor-page elementor-page-77 sc_layouts_row_fixed_initied e--ua-bli
nk e--ua-chrome e--ua-webkit mobile_layout" data-elementor-device-mode="mobile">
    <div id="page_preloader" style="opacity: 0; display: none;"></div>
    <div class="body_wrap">
    <div class="page_wrap">
    <a class="ozeum_skip_link skip_to_content_link initied" href="#content_skip_link_anchor" tabindex="1">Skip to content</a>
    <a class="ozeum_skip_link skip_to_footer_link initied" href="#footer_skip_link_anchor" tabindex="1">Skip to footer</a>
    <header class="top_panel top_panel_custom top_panel_custom_5436 top_panel_custom_new-main-header without_bg_image">
    <div class="menu_mobile_overlay" style="display: none;">
    <div class="menu_mobile menu_mobile_fullscreen scheme_dark">
    <div class="page_content_wrap">
    <a id="footer_skip_link_anchor" class="ozeum_skip_link_anchor" href="#">
    <footer class="footer_wrap footer_custom footer_custom_814 footer_custom_main-footer scheme_dark">
    </footer>
    </div>
    </div>
    <a href="#" class="trx_addons_scroll_to_top trx_addons_icon-up initied" title="Scroll to top">
    <link property="stylesheet" rel="stylesheet" id="google-fonts-2-css" href=
    "https://fonts.googleapis.com/css?family=Rubik%3A100%2C100italic%2C200%2_2C800%2C800italic%2C900%2C900italic&display=swap&subset=cyrillic&ver=6.5.3"
    type="text/css" media="all">
    <script type="text/javascript" id="trx_addons-js-extra">
    <script type="text/javascript" id="ozeum-init-js-extra">
    <script defer src="http://tripillia-reserve.com/wp-content/cache/autoptimize/js/autoptimize_4198e62...js">
    <span id="elementor-device-mode" class="elementor-screen-only">
    </body>
  </html>

```

```

<!DOCTYPE html>
<html lang="ru-RU" class="
scheme_default
js csstransitions">
<head> </head>
<body class="blog wp-custom-logo theme-ozeum ua_chrome preloader woocommerce-js tribe-js skin_default body_tag scheme_default blog_mode_home b
yle_wide is_stream blog_style_masonry_2 sidebar_hide expand_content trx_addons_present header_type_custom header_style_header-custom-5436 head
ition_default menu_side_hide elementor-default elementor-kit-4719 sc_layouts_row_fixed_initiated e--ua-blink e--ua-chrome e--ua-webkit mobile_lay
data-elementor-device-mode="mobile">
<div id="page_preloader" style="opacity: 0; display: none;"></div>
<div class="body_wrap">
<div class="page_wrap">
<a class="ozeum_skip_link skip_to_content_link initied" href="#content_skip_link_anchor" tabindex="1">Skip to content</a>
<a class="ozeum_skip_link skip_to_footer_link initied" href="#footer_skip_link_anchor" tabindex="1">Skip to footer</a>
<header class="top_panel top_panel_custom top_panel_custom_5436 top_panel_custom_new-main-header without_bg_image"></h
<div class="menu_mobile_overlay" style="display: none;"></div>
<div class="menu_mobile menu_mobile_fullscreen scheme_dark"></div>
<div class="page_content_wrap">
<div class="content_wrap"> == $0
<div class="content_wrap">
</div>
<a id="footer_skip_link_anchor" class="ozeum_skip_link_anchor" href="#"></a>
<footer class="footer_wrap footer_custom footer_custom_814 footer_custom_main-footer scheme_dark">
<div data-elementor-type="cpt_layouts" data-elementor-id="814" class="elementor elementor-814">
</footer>
</div>
</div>
<a href="#" class="trx_addons_scroll_to_top trx_addons_icon-up initied" title="Scroll to top"></a>
<link property="stylesheet" rel="stylesheet" id="google-fonts-2-css" href=
"https://fonts.googleapis.com/css?family=Rubik%3A100%2C100italic%2C200%2C800%2C800italic%2C900%2C900italic&display=swap&subset=cyrillic&ver
" type="text/css" media="all">
<script type="text/javascript" id="trx_addons-js-extra">
<script type="text/javascript" id="ozeum-init-js-extra">
<script defer src="http://tripillia-reserve.com/wp-content/cache/autoptimiz/js/autoptimiz_7e1fc03.js"></script>
<span id="elementor-device-mode" class="elementor-screen-only">
</body>
</html>

```


ПРЕЗЕНТАЦІЯ



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

1

ДИПЛОМНА РОБОТА
на ступінь вищої освіти бакалавр
із спеціальності 122 Комп'ютерні технології

РОЗРОБКА ВЕБ-ДОДАТКУ МУЗЕЮ
“ТРИПІЛЬСЬКА КУЛЬТУРА” З
ВБУДОВАНИМ КАТАЛОГОМ

Виконав: студент 4 курсу, групи КНД-42
Мороз Михайло Олексійович

Керівник: доцент кафедри Комп'ютерних наук
к.т.н. Щербина Ірина Сергіївна

Київ - 2024

2

ЗАГАЛЬНА ХАРАКТЕРИСТИКА ДИПЛОМНОЇ РОБОТИ

Тема	Розробка веб-додатку музею "Трипільська культура" з вбудованим каталогом
Мета дослідження	Розбудова веб-додатку музею з вбудованим каталогом під потреби клієнта
Об'єкт дослідження	програмне забезпечення для музеїв
Предмет дослідження	Веб додаток музею

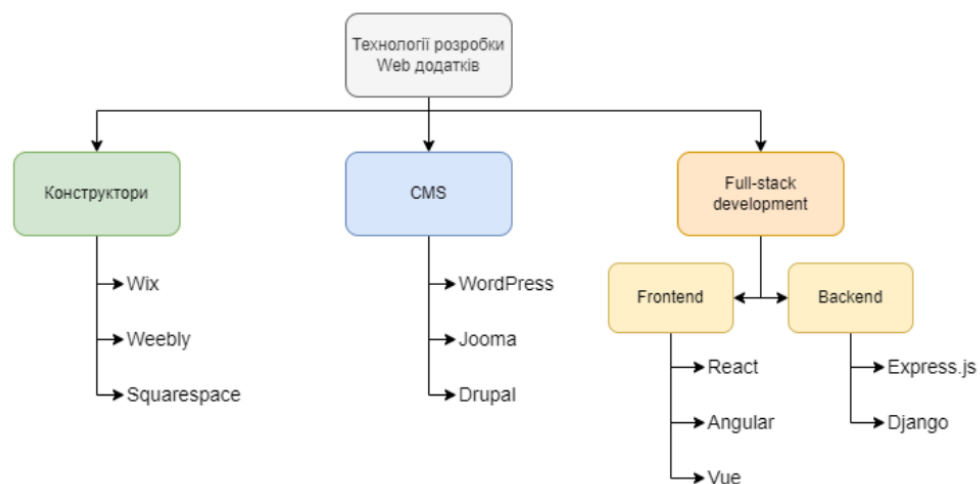
Актуальність

Актуальність теми розробки веб-додатку музею "Трипільська культура" з вбудованим каталогом полягає в необхідності цифровізації культурної спадщини, що забезпечує доступність для широкої аудиторії та сприяє збереженню артефактів. Інтернет є основним джерелом інформації, і веб-додаток дозволяє музею досягти глобальної аудиторії, організувати та представити свою колекцію ефективно.

Вимоги замовника на розробку ВЕБ додатку

1. Зручне управління контентом
2. Мінімальна вартість обслуговування
3. Яскравий та привабливий дизайн, схожий на дизайн інших музеїв світу
4. Можливість додавати нову інформацію
5. Наявність вбудованого каталогу для додавання артефактів музею
6. Наявність SEO оптимізації

ДОСЛІДЖЕННЯ ТА ВИБІР ТЕХНОЛОГІЇ



Порівняння тем WordPress для мистецьких галерей та музеїв

Тема	Основні переваги	Основні недоліки
Unimate	- Легке імпортування демо-контенту. - Динамічні варіанти відображення подій. - Безпечний магазин WooCommerce. - Можливість адаптації під бренд галереї. - Інтеграція з розсилками. - Використання Elementor.	- Менша кількість унікальних сторінок для представлення різних сервісів
Mooseoom	- Повністю адаптивна. - Унікальні сторінки для представлення послуг. - Сумісність з популярними плагінами. - Висока реактивність та мобільна сумісність. - Документація та професійна технічна підтримка.	- Можливо, складніше налаштування для новачків.
Ozeum	- Професійний дизайн. - Одноразова установка демо-контенту. - Сумісність з популярними плагінами - Повністю адаптивна. - Вбудована підтримка WooCommerce. - Зручний пошук Ajax.	- Може потребувати додаткового часу для налаштування всіх функцій.

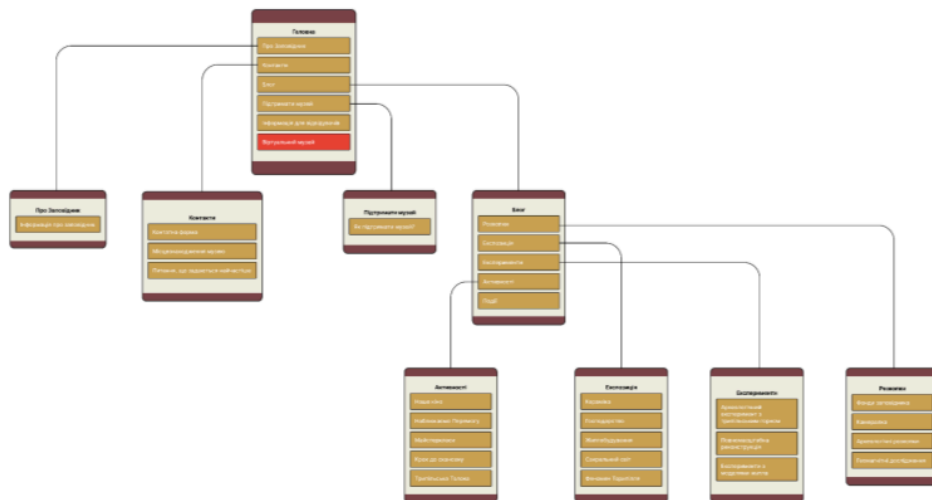
Вибір технологій

- Content Management System (CMS): WordPress
- Тема WordPress: Ozeum
- Інструменти дизайну: Figma
- Плагіни:
 - Elementor
 - Yoast SEO
 - WP Super Cache
 - WP Mail SMTP
 - Optimole.
 - WooCommerce.
 - Contact Form 7

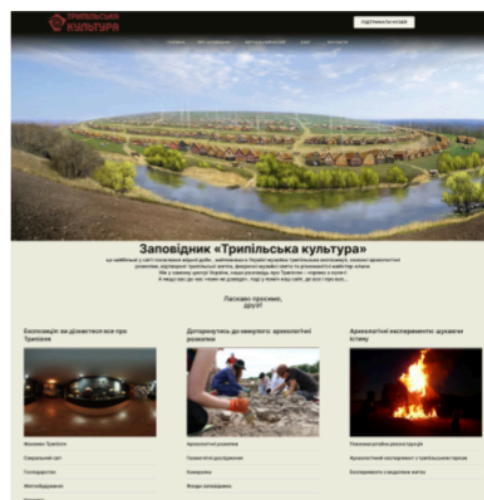
Розробка веб-додатку



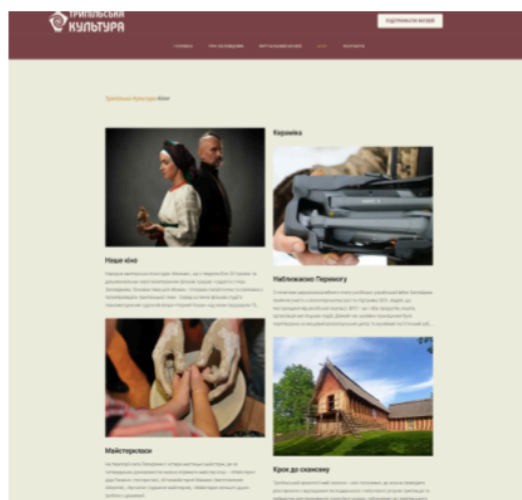
Карта сайту веб-додатку музею
"Трипільська Культура"



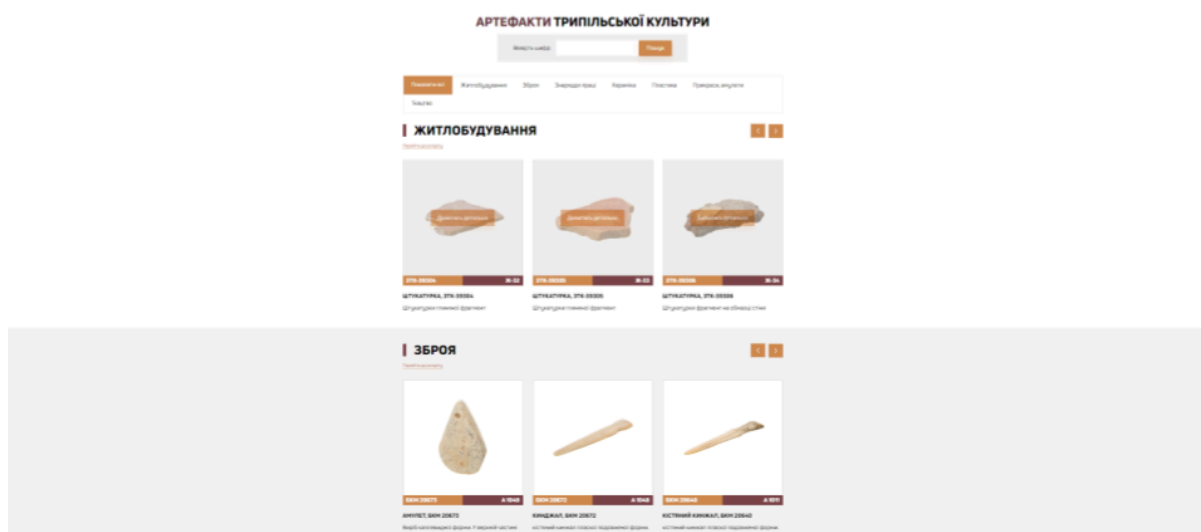
Головна сторінка веб-додатку музею "Трипільська Культура"



Сторінка блогу веб-додатку музею “Трипільська Культура”



Сторінка каталогу веб-додатку музею “Трипільська Культура”



Використані плагіни



Elementor



Contact Form 7



WPvivid backup



Wordfence



WP Super Cache



Optimole



WooCommerce



Yoast SEO

Налаштування SEO оптимізації



Оптимізації сайту під пошукові системи

- Встановлення та налаштування плагіну **Yoast SEO**
- Налаштування файлу **robots.txt** за допомогою плагіну **Virtual Robots.txt**
- Використання **Breadcrumb NavXT**
- Дослідження та вибір ключових слів
- Оптимізація швидкості завантаження сайту

- Проаналізовано вимоги проекту та обрано технологію WordPress для розробки веб-додатку музею "Трипільська Культура", що забезпечує зручність у користуванні та мінімальні витрати на обслуговування. Вибрано та налаштовано тему Ozeum, яка поєднує сучасний дизайн з функціональністю, відповідаючи всім вимогам музею. Створено UI/UX дизайн у Figma, що відображає естетичний вигляд та зручність навігації.
- Налаштовано головні плагіни для забезпечення необхідних функцій сайту, включаючи Contact Form 7 для зв'язку з відвідувачами, WP Super Cache для прискорення завантаження сторінок, та WP-Mail-SMTP для надійного надсилання електронних листів через поштовий сервіс Ukr.net. Впроваджено каталог артефактів на базі WooCommerce, що дозволяє ефективно демонструвати колекцію музею без функціоналу електронної комерції.
- Оптимізовано роботу з зображеннями за допомогою плагіну Optimole, що забезпечує швидке завантаження сторінок навіть при великій кількості високоякісних зображень. Налаштовано SEO за допомогою плагінів Yoast SEO, Virtual Robots.txt та Breadcrumb NavXT, що дозволяє забезпечити високі рейтинги у пошукових системах та покращити видимість сайту.

Апробація

Результати дослідження бакалаврської дипломної роботи апробовані на IV Всеукраїнській науково-практичній конференції «Сучасні інтелектуальні інформаційні технології в науці та освіті», 15 травня 2024 року.

Тема: СТВОРЕННЯ ВЕБ-ДОДАТКУ МУЗЕЮ "ТРИПІЛЬСЬКА КУЛЬТУРА З ВБУДОВАНИМ КАТАЛОГОМ

Виконав: Мороз Михайло Олексійович,
студент 4 курсу, групи КНД-42

Науковий керівник: Щербина Ірина Сергіївна,
доцент кафедри Комп'ютерних наук