

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: **«ВИКОРИСТАННЯ WORDPRESS ДЛЯ
РОЗРОБКИ САЙТУ-ПОРТФОЛІО З ОСОБИСТИМ
КАБІНЕТОМ КОРИСТУВАЧА»**

Виконала: студентка 2 курсу, групи 8.1211-з
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)
освітньої програми інженерія програмного забезпечення
(назва освітньої програми)
А.Б. Алексанян
(ініціали та прізвище)

Керівник старший викладач кафедри програмної інженерії,
к.ф.-м.н. Чопорова О.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри фундаментальної та прикладної
математики, професор, д.т.н. Гребенюк С.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти магістр

Спеціальність 121 інженерія програмного забезпечення

Освітня програма інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

Лісняк А.О.

(підпис)

« _____ » _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Алексанян Анастасії Борисівні

(прізвище, ім'я та по-батькові)

1. Тема роботи (проєкту) Використання WordPress для розробки сайту-портфоліо з особистим кабінетом користувача

керівник роботи (проєкту) Чопорова Оксана Володимирівна, к.ф.-м.н.
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 10 » травня 2022 року № 514-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік питань до розробки.
3. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі, аналіз предметної області.
2. Приклади використання інформаційної системи.
3. Переваги користування інформаційною системою.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	08.05.2022	
2.	Збір вихідних даних.	28.05.2022	
3.	Обробка методичних та теоретичних джерел.	06.09.2022	
4.	Розробка першого та другого розділу.	17.10.2022	
5.	Розробка третього розділу.	21.11.2022	
6.	Оформлення та нормоконтроль кваліфікаційної роботи магіста.	02.12.2022	
7.	Захист кваліфікаційної роботи.	16.12.2022	

Студент _____
(підпис)

А.Б. Алексанян _____
(ініціали та прізвище)

Керівник роботи _____
(підпис)

О.В. Чопорова _____
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

А.В. Столярова _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Використання WordPress для розробки сайту-портфоліо з особистим кабінетом користувача»: 58 с., 47 рис., 21 джерело.

БАЗА ДАНИХ, ПРОЄКТ, САЙТ, ХОСТІНГ, WORDPRESS.

Об'єкт дослідження: сайт портфоліо з особистим кабінетом користувача на основі WordPress.

Мета роботи: навчитися створювати сайти портфоліо за допомогою WordPress, проаналізувати предметні області, створити динамічний контент.

Метод дослідження – аналітичний.

У дипломній роботі викладено теоретичні відомості для виконання поставленого завдання, реалізація поставленої задачі.

У роботі розглянуто деякі основні особливості WordPress, зокрема, ієрархія шаблонів, зв'язок баз даних з ядром CMS. Досліджено основні функції, за допомогою яких побудовано зв'язки між шаблонами. Особливу увагу приділено до зв'язку шаблонів та баз даних. Проведено аналіз способів безпеки на прикладі даної CMS, а також деяких стандартних можливостей. Наведено приклади застосування CMS WordPress.

SUMMARY

Master's qualifying paper «Using Wordpress for Development of the Portfolio Website with the User Cabinet»: 58 pages, 47 figures, 21 references.

DATABASE, PROJECT, SITE, HOSTING, WORDPRESS.

The object of research: a WordPress-based portfolio site with a personal user account.

The purpose of the thesis: learn how to create portfolio sites using WordPress, analyze subject areas, create dynamic content.

The research method is analytical.

In the work the theoretical data for performance of the set task, realization of the set task is stated.

Thesis discusses some key features of WordPress, in particular, the hierarchy of templates, the relationship of databases with the CMS core. The main functions used to build relationships between templates are explored. Particular attention is paid to the connection of templates and databases. The analysis of security methods on the example of this CMS, as well as some standard features. Examples of CMS WordPress application are given.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат.....	4
Summary.....	5
Вступ	8
1 Динамічний контент	10
1.1 Актуальність.....	10
1.2 Динамічний контент і його відмінності від статичного	11
1.3 Основне про CMS	14
1.4 Система управління контентом WordPress	17
1.5 Висновки до першого розділу	18
2 Використовувані технології.....	19
2.1 Огляд основних технологій.....	19
2.1.1 HTML	19
2.1.2 CSS	19
2.1.3 JavaScript.....	20
2.1.4 PHP	22
2.1.5 CMS	23
2.2 Вибрані методи проектування.....	24
2.2.1 Wordpress	24
2.2.2 Advanced Custom Fields.....	26
2.2.3 Open Server	28
2.3 Висновки до другого розділу	31
3 Сайт «ALEXANYAN»	33
3.1 Налаштування середовища розробки	33
3.2 Створення каркаса теми WordPress	37
3.3 Налаштування панелі адміністратора.....	41
3.4 Створення сторінки сайту.....	45

3.5 Створення полів для сторінки	47
3.6 Огляд інтерфейсу сайту	51
Висновки	56
Перелік посилань	57

ВСТУП

Відомо, що у мережі понад 1,8 мільярда активних веб-сайтів, і це число зростає з кожною секундою. Всім цим веб-сайтам потрібна система управління контентом, програмне забезпечення, яке створює і зберігає електронний контент, такий як документи, зображення, веб-сторінки тощо. Наведемо переваги ефективної системи управління контентом у веб-дизайні.

Зрозуміло, що існує безліч причин, за якими використання системи управління контентом в веб-дизайні вигідно для бізнес-сайтів. Основі переваги включають таке:

- забезпечує послідовну публікацію контенту;
- дозволяє швидко виправляти помилки веб-сайту;
- дозволяє швидко створювати веб-сайти;
- дозволяє «підганяти» під бізнес.

Система управління контентом, також відома як CMS, являє собою систему управління веб-сайтом, на якому здійснюється управління і обслуговування веб-контенту. Системи управління контентом контролюють все: від редагування тексту і додавання зображень до створення галерей. CMS – це місце, де публікується новий веб-контент, наприклад локальні сторінки та блоги [3].

Можна стверджувати, що WordPress.org – це таке місце, де знаходиться веб-сайт з системою управління контентом. Зокрема, WordPress дозволяє користувачам підвищити свою актуальність, заощадити час і гроші, а також контролювати свій веб-контент.

Відомо також, що актуальність бізнесу підвищується за рахунок рейтингу в пошукових системах. Зрозуміло, що і WordPress дозволяє компаніям поліпшувати і підтримувати свій рейтинг в пошукових системах. Модулі, такі як Yoast, дозволяють користувачам WordPress оптимізувати свій

веб-сайт і аналізувати свій веб-контент [2].

Можна стверджувати, що WordPress дозволяє користувачам працювати більш ефективно. Так WordPress пропонує тематичні шаблони, щоб позбавити користувачів від необхідності вибирати правильні колірні палітри і макети для веб-дизайну. Відповідні теми зекономлять час і гроші, надаючи креативні шаблони, що включають витрати, пов'язані з запуском з нуля.

Ніяка інша CMS не дає користувачам такого контролю над своїм контентом, як WordPress. Користувачі можуть легко оновлювати свій сайт і додавати новий контент. Вся команда може робити внесок в свій веб-сайт, в той час як адміністратор сайту може обмежувати доступ до певних розділів.

Таким чином, в наш час мати сайт на основі CMS – це легко і просто. Особливо це буде зручно, якщо просто хочете мати можливість керувати своїм власним контентом, або, запропонувати користувачу приємний та особливий досвід користування сайтом.

В даній роботі ми будемо використовувати саме CMS WordPress для реалізації динамічного контенту при створенні сайту.

Для досягнення поставленої мети у першому розділі ми розглянемо WordPress, як найпоширенішу CMS у світі, дослідимо основні вистроєні функції, наведемо приклади їх використання.

Другий розділ роби буде присвячено саме реалізації динамічного контенту на даному сайті. Наведено приклади використання плагінів та вбудованих можливостей, розроблено правильно ієрархію шаблонів, враховано основні пробіли в безпеці.

1 ДИНАМІЧНИЙ КОНТЕНТ

1.1 Актуальність

Незважаючи на те, що Інтернет давно і міцно увійшов в наше життя, багато підприємці і навіть великі фірми не розуміють, що їм дасть створення власного сайту, адже існують і інші, добре зарекомендували себе, перевірені способи самореклами: телебачення, радіо, ЗМІ, банери, фласри тощо.

У кожної сучасної компанії є сайт. Це один із елементів престижу, адже саме в Інтернеті потенційні клієнти в першу чергу шукатимуть інформацію про компанію. І якщо в ньому немає хоча б односторінкового листа з прайс-листом, це здасться підозрілим – наскільки неуспішна ця компанія, якщо вона не може створити навіть невеликий веб-ресурс.

Актуальність створення сайту полягає ще й у тому, що якщо ви хочете якомога швидше донести інформацію до величезної кількості людей, то краще, ніж власний сайт робити це ні в якому разі. Веб-ресурс дозволяє стисло і водночас повно представити інформацію про компанію та її продукти чи послуги. Також сайт може повідомляти про новини компанії, зміни в прайс-листі або режимі роботи, містити відгуки вдячних клієнтів.

Не кожній компанії потрібен великий портал зі складним дизайном і функціоналом. Іноді достатньо невеликого сайту-візитки, який можна зробити самостійно або замовити у професіоналів за невелику плату.

Актуальність розробки сайту пояснюється наступними факторами:

- 1) швидкість надання інформації широкому колу людей;
- 2) покращення іміджу компанії та підвищення її популярності;
- 3) вміння спілкуватися з клієнтами;
- 4) організація маркетингових досліджень;
- 5) реклама та залучення покупців і клієнтів;

б) збільшення трафіку.

Потрібно розуміти, що жоден інший ресурс не принесе стільки переваг, як власний сайт, будь то візитка, інтернет-магазин або будь-який інший веб-ресурс.

Слід зазначити, що одним лише створенням сайту справа не обмежується. Його потрібно буде розвивати і підтримувати, підвищувати конверсію і вчасно поповнювати, а це непросте завдання. Однак всі ці зусилля і витрати повністю окупляться тим прибутком, який принесе сайт.

1.2 Динамічний контент і його відмінності від статичного

Розглянемо контент і його види, оскільки відомо, що HTML – це мова веб-програмування, яку можна використовувати для написання коду і створення веб-сайтів. Кодування можна писати і редагувати практично в будь-якому текстовому редакторі. Однак існують спеціальні програми для редагування тексту, які виділяють синтаксис, щоб полегшити роботу веб-розробника (наприклад, Sublime Text і Notepad++) [4].

При розробці слід використати CMS. CMS (система управління контентом), з іншого боку, в основному безкоштовне програмне забезпечення з відкритим вихідним кодом, яке ми можемо завантажувати, завантажувати на свій сервер, або налаштувати та використовувати з мінімальними знаннями програмування. Ми можете створити повноцінний веб-сайт з системою адміністрування, яке можна використовувати для написання повідомлень, створення сторінок, завантаження файлів, редагування компонентів дизайну і виконання безлічі інших завдань веб-розробки.

Більшість систем управління контентом, таких як WordPress, є інтуїтивно зрозумілі і прості у використанні інструменти, які відмінно підходять для

ведення блогу або веб-сайту і навіть для реєстрації користувачів і коментарів [2].

Зупинимось далі на видах саме контенту. Статичний контент – це контент, який зберігається у файлі. Щоб змінити такий контент знадобиться редагувати на пряму файл в якому він знаходиться.

До переваг статичного контенту можна віднести такі:

1) статичний контент підходить для складних веб-сайтів, які потребують великої кількості функціональності: перевага тут у тому, що можна впливати на все, що захочеться, можна створити свою власну платформу на основі CMS;

2) традиційний HTML – це те, де кожен раз потрібно робити це самому: при написанні коду можна використовувати уяву, щоб створити все, що хочеться реалізувати на своєму веб-сайті.

Але є і недоліки:

1) якщо потрібно щось змінити на «серверній частині» веб-сайту, слід перебувати за своїм комп'ютером в середовищі розробки, щоб внести зміни. Немає, на жаль, гнучкості для доступу і редагування контенту з будь-якого комп'ютера;

2) можна створити власне програмне забезпечення на основі CMS зі статичним HTML, а потім використовувати свої пропріетарні системи управління контентом для редагування контенту сайту з будь-якого місця. Однак такий крок вимагатиме значного часу і роботи, не говорячи вже про значні технічні знання в області кодування.

На відміну від попереднього динамічний контент – це контент, який знаходиться в базі даних. З бази даних він напряму виводиться користувачеві на сторінку браузера. Для роботи з динамічним контентом часто використовують CMS.

До переваг динамічного контенту з CMS віднесемо наступне:

1) можна часто міняти контент;

- 2) установка виконується дуже швидко і легко;
- 3) особливих технічних знань в області кодування не потрібно: всі інструменти для додавання тексту, зображень або відео, а також для створення нових сторінок або підлеглих надано всередині самої платформи;
- 4) більшість платформ мають вбудовану систему безпеки;
- 5) доступні різні шаблони, які можна змінити відповідно до своїх потреб, наприклад, Theme Forest містить понад 1000 шаблонів;
- 6) більшість шаблонів вже адаптивні;
- 7) у систему легко додати сторонні плагіни, такі як слайдери банерів, галереї зображень, інструменти SEO і системи електронної комерції;
- 8) можна управляти своїм контентом з будь-якого комп'ютера в будь-якому місці, де є інтернет;
- 9) для завантаження вмісту в Інтернет не потрібне програмне забезпечення FTP;
- 10) гарний зв'язок з пошуковими мережами;
- 11) блоги вже вбудовані і готові до роботи.

До недоліків слід віднести:

- 1) CMS вимагає більше пам'яті, потужності процесора і обслуговування програмного забезпечення, ніж її конкуренти HTML, невеликі зміни налаштувань на сервері можуть призвести до збою всього сайту;
- 2) деякі шаблони погано закодовані;
- 3) створювати величезний і складний веб-сайт з системою управління контентом – не найкраща ідея: наприклад, такий сайт, як amazon.com, має складний, великий дизайн, який неможливо було б створити за допомогою платформ на основі CMS. Краще його створювати з нуля, використовуючи власний HTML і інші мови програмування.

1.3 Основне про CMS

У цьому розділі робот наведемо основні поняття CMS та їх можливості, що дозволить у подальшому мати явлення що, до роботи та можливостей CMS. Відомо, що система управління контентом (CMS) – це програмний додаток, який дозволяє користувачам створювати, редагувати, спільно використовувати, публікувати і зберігати цифровий контент. CMS зазвичай використовуються для управління корпоративним контентом (ECM) і управління веб-контентом (WCM) [1].

Зокрема, CMS надає графічний користувальницький інтерфейс з інструментами для створення, редагування та публікації веб-контенту без необхідності писати код з нуля. CMS складається з двох компонентів: програми управління контентом (CMA) і додатків доставки контенту (CDA). CMA – це графічний користувальницький інтерфейс, який дозволяє користувачам розробляти, створювати, змінювати і видаляти контент з веб-сайту без знання HTML. Компонент CDA надає внутрішні служби, які підтримують управління та доставку контенту після того, як користувач створює його в CMA.

Наведемо далі основні функції CMS, які дозволяють:

- 1) зручні для пошукової оптимізації URL-адреси;
- 2) інтегрована і онлайн-довідка, включаючи дошки обговорень;
- 3) групові системи дозволів і безпеки;
- 4) повна підтримка шаблонів і настроюються шаблони;
- 5) прості процедури установки і управління версіями за допомогою майстра;
- 6) панель адміністратора з підтримкою декількох мов;
- 7) мінімальні вимоги до сервера;
- 8) інтегровані файлові менеджери; а також інтегровані журнали аудиту.

Переваги використання системи управління контентом:

1) зручність використання: завдяки графічному інтерфейсу програмне забезпечення можуть використовувати навіть люди з обмеженими технічними знаннями;

2) легко шукати інформацію: вбудована функція пошуку дозволяє користувачам вводити те, що вони шукають, і отримувати список елементів, що повертаються їм – так само, як пошукова система Google;

3) легко управляти контентом: легко не тільки створювати контент, але і видаляти його, CMS дозволяє легко відмінити публікацію контенту, щоб підтримувати веб-сайти в актуальному стані;

4) доступний звідки завгодно: CMS може бути хмарної або локальної, але користувачі можуть отримувати доступ до контенту з будь-якого місця за допомогою пристрою, підключеного до Інтернету;

5) дозволяє декільком користувачам: CMS спрощує управління дозволами на публікацію;

6) миттєве оновлення контенту: CMS дозволяє користувачам управляти контентом і оновлювати його в режимі реального часу - без необхідності чекати розробника;

7) легко масштабується: CMS дозволяє підприємствам легко додавати нові веб-сторінки;

8) легко оновлюється: команди розробників можуть випускати оновлення всього декількома клацаннями миші.

Отже, існує безліч безкоштовних CMS і CMS за передплатою, доступних для особистого і корпоративного використання. Наведемо приклади найбільш популярних постачальників систем управління контентом.

Joomla – це безкоштовна система управління веб-контентом з відкритим вихідним кодом, побудована на платформі MVC. Відомо, що Joomla написана на мові сценаріїв PHP і пропонує такі функції, як кешування, RSS-канали, повідомлення в блогах, пошук і підтримку мовного перекладу [14].

WordPress – це також ще одна безкоштовна WCMS з відкритим вихідним кодом, яка заснована на PHP і MySQL. Компанії можуть використовувати WordPress в хмарі або розгорнути його на локальному комп'ютері в якості власного веб-сервера. Це програмне забезпечення легко налаштовується, є безліч тем і плагінів WordPress. Це також популярна платформа для ведення блогів [1].

Backdrop CMS – це безкоштовна CMS з відкритим вихідним кодом, яка є частиною проекту Drupal і надає доступну CMS для малих і середніх організацій. Зрозуміло, що сам по собі Backdrop пропонує тільки самі основні функції управління веб-контентом, але його можна розширити за допомогою різних модулів [15].

Також **Wix** – це так система, яка пропонує як безкоштовні, так і платні плани. Можливості включають інструменти для спільної роботи в команді, інтеграцію зі сторонніми виробниками, безпеку корпоративного рівня, сотні шаблонів дизайну і можливість додавання власного коду. Wix також надає цілодобову підтримку клієнтів без вихідних [16].

HubSpot CMS – це платна CMS має конструктор сторінок з перетягуванням, рекомендації по пошукової оптимізації та теми веб-сайтів. Безкоштовна CRM-платформа HubSpot також включена в кожен пакет CMS, що дозволяє користувачам відстежувати клієнтів і контент в одному місці [17].

Magnolia – це автономна CMS з відкритим вихідним кодом, яка забезпечує інтеграцію з численними маркетинговими технологіями, включаючи CRM-системи, інструменти аналітики і автоматизації маркетингу. Можливості включають персоналізацію та оптимізацію, а також управління цифровими активами [18].

Таким чином, CMS в наш час дуже багато, у кожного розробника є можливість вибрати з цієї величезної кількості. Головне щоб вибір CMS був обдуманим з усіх боків, а створення сайту стало захоплюючим досвідом.

1.4 Система управління контентом WordPress

Розглянемо одну із систем управління контентом (CMS) з відкритим вихідним кодом, яка спрощує створення і управління веб-сайтом. Це WordPress, яка є однією з найпопулярніших CMS в світі. Люди щодня використовують WordPress для підтримки своєї присутності в Інтернеті.

WordPress має однією з переваг безкоштовне використання: немає ніяких зборів за установку або річних витрат, крім хостингу.

Це програмне забезпечення доступне за нульовою ціною, оскільки воно запускається добровольцями, які беруть участь в проекті з відкритим вихідним кодом. Програмне забезпечення з відкритим вихідним кодом не схоже на інші матеріальні продукти, оскільки жодна компанія або приватна особа не володіє WordPress. З цієї причини його підтримка, підтримка і розвиток повністю залежать від волонтерів.

Спільнота WordPress існує тому, що всі, від початківців розробників до дизайнерів, витрачають свій час і сили, пропонуючи свої послуги безкоштовно усім користувачам, тому що вони вірять в цінні послуги, які надає WordPress.

WordPress надає власникам повну можливість створювати, публікувати і управляти своїм контентом. Завдяки простому інтерфейсу WordPress ідеально підходить для користувачів, яким зручніше працювати з текстовими редакторами, ніж з програмами FTP та написаними вручну веб-сторінками. Розробники опублікували тисячі плагінів і віджетів WordPress, які дозволяють нам створити дійсно якісний сайт.

WordPress дає велику кількість можливостей розробнику. В основі роботи WordPress лежить добре налагоджена зв'язок між ядром і базою. Ми легко можемо використовувати це в свою користь. Також WordPress редактор Gutenberg - плагін, який в останніх версіях WordPress вже вбудований в нього, дозволяє візуально і без застосування коду змінювати вміст сторінок і постів.

У даній роботі ми скористаємося такою можливістю. WordPress має дуже

велику базу плагінів, створених звичайними людьми, частина з них платні, але є і безкоштовні. Ми використовували безкоштовний плагін з можливістю розширення функціоналу після оплати. Даний плагін називається АСF (Advanced Custom Fields) – плагін дозволяє нам створювати динамічний контент по всьому сайту. АСF тісно взаємодіє з тією самою базою даних. Він створює поля, які можна привласнити майже до будь-якого розділу в панелі адміністратора, дані яких записуються в поля таблиць бази даних. Найчастіше це використовується для виведення динамічного контенту на сайті, а саме: виведення контенту сторінок і постів, виведення шапки і підвалу сайту. Саме так більшість користувачів використовують даний плагін, але можливостей як організувати його роботу з боку бекенду набагато більше. У даній роботі ми наочно побачимо застосування даного плагіна в самих нестандартних місцях.

1.5 Висновки до першого розділу

Виходячи з викладеного в даному розділі можна з упевненістю сказати, що розробка сайтів з динамічним контентом в даний час як ніколи актуальна.

Найпопулярнішим і найшвидшим шляхом є розробка сайту на CMS і найкращим рішенням для цього може стати зв'язка WordPress і плагіна АСF. Так як вони ідеально взаємодіють між собою і дають великий запас можливостей, реалізувати який не під силу навіть багатьом досвідченим розробникам, а отже, завжди буде місце для розвитку та лікування навичок.

2 ВИКОРИСТОВУВАНІ ТЕХНОЛОГІЇ

2.1 Огляд основних технологій

2.1.1 HTML

HTML – це інструмент розмітки тексту. Мова HTML містить достатню кількість елементів, які дозволяють оформити документ на свій смак.

2.1.2 CSS

CSS – це мова каскадних таблиць стилів. Він призначений для покращення дизайну веб-сторінок. CSS використовується веб-розробниками для налаштування зовнішнього вигляду (шрифтів, кольорів, відступів, компонування тощо) веб-сторінок. CSS призначений для відділення основного вмісту документа (написаного на мові розмітки, наприклад HTML) від стилю цього вмісту (написаного на CSS). Це розділення дає веб-розробникам більше гнучкості, полегшуючи визначення зовнішнього вигляду документів і створення повторюваних елементів розмітки [19].

Використовуючи HTML і CSS, важливо розуміти наступне:

- код HTML створює текст логічно, тобто, задає структуру Веб-сторінки: розташування та порядок абзаців, графіки, рядків і комірок у таблиці та спеціальне значення окремих текстових фрагментів;
- таблиці стилів CSS формують тексти фізично, тобто, встановити вигляд веб-сторінки: яким шрифтом буде набраний звичайний текст абзаців, яким кольором виділяти заголовки, чи матиме таблиця рамку тощо.

Етикет веб-дизайну вимагає, щоб презентація веб-сторінки була відокремлена від її структури. Тому професійні веб-дизайнери по можливості поміщають визначення стилів CSS в окремі файли (таблиці стилів).

Крім того, HTML-код, який не захищений визначеннями стилів, стає більш читабельним. Справа в тому, що в HTML одночасно існують теги логічного і фізичного форматування веб-сторінок.

Каскадні таблиці стилів за своєю суттю є нединамічними. Вони дозволяють визначити, як буде виглядати документ після завантаження, і нічого більше. Але властивості веб-сторінок, створених за допомогою CSS, можна динамічно змінювати за допомогою JavaScript.

2.1.3 JavaScript

Популярність мови JavaScript пояснюється широкими можливостями взаємодії з елементами веб-сторінки без її перезавантаження. Це дозволяє приховувати і показувати фрагменти дизайну, переміщати їх і змінювати зовнішній вигляд. За допомогою цих дій ви можете створювати ефекти презентації, меню, невеликі ігри, обробляти дані форми та керувати вмістом.

Підтримка шарів. Шари – це частини HTML, які можна розмістити на веб-сторінці, наклавши їх один на одного з точністю до пікселів. Шари набувають все більшого визнання як засіб компонування веб-сторінок і створення різних ефектів. Частково це пов'язано з тим, що їх параметри легко динамічно змінюються за допомогою скриптів, що дає можливість створювати меню та інші виразні інструменти дизайну.

Робота з формами. Через скрипти зручно отримувати та обробляти будь-які дані форми, це дозволяє перевірити інформацію на правильність введення перед відправкою на сервер. Ви можете створити «охорону дурнів», щоб забезпечити введення правильних повідомлень в елементи форми. Отже, якщо

вас попросять ввести число в текстовому полі, вам слід негайно припинити можливість введення нецифрових символів.

Робота з зображеннями. За допомогою сценаріїв можна попередньо завантажувати зображення. Навіть перед прямим доступом до 14 зображень браузер розміщує їх у своїй пам'яті, щоб за потреби швидше відображати в документі.

Робота з файлами cookie. JavaScript повністю підтримує файли cookie, які є невеликими текстовими файлами на локальному комп'ютері, які зберігають технічну інформацію. Файли cookie можуть використовуватися для збереження дати останнього відвідування читачем, паролів, а також будь-якої інформації про дії відвідувача на сайті. Цей додаток дозволяє персоналізувати сайт і зробити його більш зручним для відвідувачів.

Створення нових вікон. Сценарії дозволяють створювати нові вікна, встановлювати тип відображення цих вікон і формувати їх вміст. Крім того, для створених таким чином вікон можна вимкнути адресний рядок, смуги прокрутки, меню та контролювати їх розмір і положення на екрані.

Перевірка браузера. Підхід різних браузерів до об'єктної моделі, хоча і стандартизований, ще не повністю прийнятий усіма. Тому, щоб створити універсальний документ, який однаково працює скрізь, ви повинні перевірити, який браузер зараз використовується, і надати йому відповідний код.

Математичні функції. JavaScript містить усі необхідні арифметичні операції, підтримує всі стандартні математичні функції, як з цілими числами, так і з плаваючою комою.

Розробляючи програму JavaScript, важливо пам'ятати про різницю між клієнтською та серверною платформами [20].

2.1.4 PHP

PHP – це мова програмування для створення веб-сайтів. PHP дозволяє автоматизувати роботу з сайтом. Це мова сценаріїв, вбудована в HTML, яка інтерпретується та виконується на сервері. Різниця між PHP і JavaScript полягає в тому, що скрипт PHP виконується на сервері, а результат роботи передається клієнту, а код JavaScript повністю передається на машину клієнта і тільки там виконується.

PHP – це вбудована мова сценаріїв. Мета мови – дозволити веб-розробникам швидко створювати динамічно генеровані сторінки. Крім того, розробникам веб-додатків не потрібно говорити, що веб-сторінки – це не лише текст і зображення. Вартий уваги сайт повинен підтримувати певний рівень взаємодії з користувачем: пошук інформації, продаж продуктів, конференції тощо. PHP працює як частина веб-сервера. У цій мові немає сильної типізації даних і немає потреби в діях виділення/звільнення пам'яті. Програми, написані на PHP, досить легко читаються. Написаний PHP-код легко читати та розуміти візуально.

Функції PHP. На PHP можна зробити все:

- обробляти дані з форм;
- генерувати динамічні сторінки;
- прийом і відправка сесій – файли cookie (cookies).

Крім того, PHP включає підтримку багатьох баз даних, що робить написання веб-додатків з підтримкою баз даних неймовірно легким. Крім того, PHP розуміє протоколи IMAP, SNMP, NNTP, POP3 і навіть HTTP, а також має можливість працювати з сокетом (сокетами) і спілкуватися за допомогою інших протоколів.

Можливості мови значно розширюються за допомогою додаткових модулів, що містять різні функції. Тому, якщо ви хочете використовувати

функції будь-якого модуля (і самого PHP), ви повинні спочатку перевірити, чи підтримує їх веб-сервер.

Недоліки PHP. За своєю ідеологією PHP спочатку був орієнтований на написання невеликих скриптів. PHP є інтерпретованою мовою, а тому не може зрівнятися за швидкістю зі скомпільованим C. Однак при написанні невеликих програм, що, загалом, притаманне проектам на PHP, йому просто немає рівних [21].

PHP спочатку був аббревіатурою від Personal Home Page, тепер це означає Personal Hypertext Preprocessor.

2.1.5 CMS

Система керування вмістом (CMS) – це інформаційна система або комп'ютерна програма, яка використовується для забезпечення та організації спільного процесу створення, редагування та керування вмістом (рис. 2.1) [2].



Рисунок 2.1 – Найпопулярніший CMS

Основні функції CMS:

1) надання інструментів для створення контенту, організація співпраці з контентом;

- 2) управління контентом: зберігання, контроль версій, дотримання режиму доступу, контроль документообігу тощо;
- 3) публікація контенту;
- 4) подання інформації у формі, зручній для навігації, пошуку.

Система керування контентом може містити різноманітні дані: документи, фільми, фотографії, номери телефонів, наукові дані тощо. Така система часто використовується для зберігання, керування, перегляду та публікації документації. Контроль версій є однією з його головних переваг, коли вміст змінюється групою людей.

2.2 Вибрані методи проєктування

Для реалізації поставленого завдання потрібно наступне програмне забезпечення:

- 1) Wordpress;
- 2) Advanced Custom Fields;
- 3) Open Server.

Далі описуються їх основні можливості, переваги, недоліки та характеристики.

2.2.1 Wordpress

WordPress – це система керування вмістом з відкритим кодом; написаний на PHP; сервер бази даних – MySQL; випущений під GNU GPL версії 2. Сфера застосування – від блогів до досить складних новинних ресурсів і інтернет-магазинів (рис. 2.2).

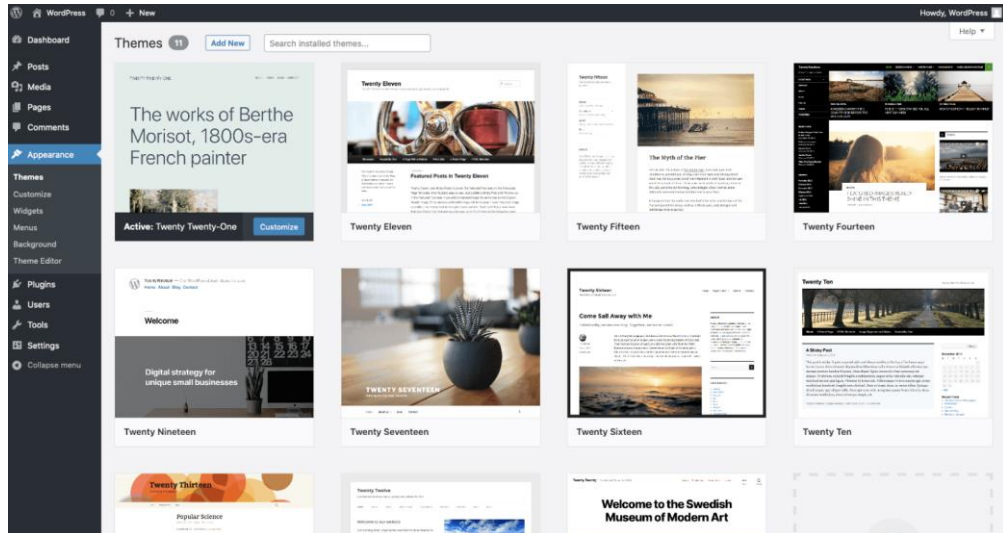


Рисунок 2.2 – Адміністративна панель Wordpress

Перша версія WordPress була створена в 2003 році. Її, як відгалуження зовсім іншого проекту, написав Метт Малленвег, програміст з Х'юстона, штат Техас. Спочатку WordPress був зосереджений на створенні персональних блогів.

Особливості Wordpress. WordPress – це безкоштовне програмне забезпечення з відкритим кодом, яке можна завантажити та встановити за лічені хвилини. Крім того, плагіни та теми розширюють можливості платформи.

Оскільки платформа WordPress є відкритою, це дозволяє змінювати вихідний код цієї системи на ваш розсуд. Якщо зміни будуть успішними, їх можна буде запропонувати спільноті для включення в наступний офіційний випуск цієї системи. Але лише п'ять основних розробників можуть додавати новий код до основної платформи цих випусків. Крім того, додаткові функції можна реалізувати за допомогою модулів розширення або плагінів.

Плагін – це розширення WordPress, яке додає функцію чи функціональність, які спочатку не були вбудовані в нього.

Платформа WordPress існує у двох формах: WordPress.org і WordPress.com. Веб-сайт WordPress.org містить останні версії цієї cms, плагіни, теми та локалізації. WordPress.com – це служба від Automattic, яка була

заснована для надання безкоштовного хостингу WordPress. WordPress.com побудовано на форку WordPress під назвою WordPress MU, який дозволяє вам керувати декількома сайтами одночасно за допомогою однієї встановленої копії цієї cms.

WordPress.com належить і управляється Automattic, яка також підтримує та обслуговує WordPress.org. Ця компанія надає інфраструктуру для підтримки CMS WordPress і пропонує роботу багатьом веб-розробникам цієї системи.

2.2.2 Advanced Custom Fields

Існує багато різних способів створити веб-сайт, особливо якщо ваш сайт працює на базі WordPress. Для веб-розробників поширеним інструментом є плагін під назвою Advanced Custom Fields. Advanced Custom Fields – один із найнадійніших інструментів, які розробники використовують для створення WordPress.

Advanced Custom Fields – це інструмент, який використовується для створення макетів для вашого сайту WordPress. Загальні функції такі: вибір полів на панелі адміністратора WordPress, вибір місця відображення цих полів у серверній частині (щоб користувачі могли вводити вміст), а потім реалізація візуального рівня для цього вмісту (див. рис. 2.3).

Нижче наведено список деяких типів полів, які пропонує ACF (та багато інших):

- текст;
- текстова область;
- повзунок діапазону;
- група кнопок;
- прапорець;
- перемикач;

- вибирати;
- правда-неправда;
- файл;
- галерея;
- зображення;
- вставити;
- вибір кольору;
- гугл-карта;
- час;
- акордеон;
- розмістити об'єкт;
- посилання.

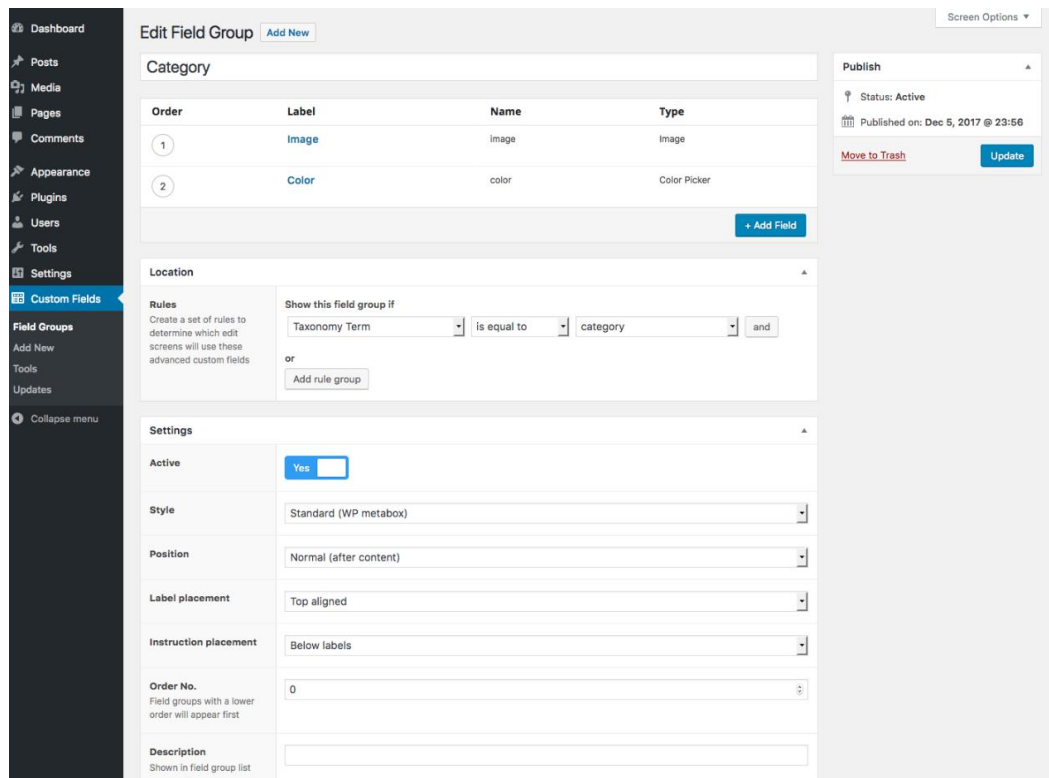


Рисунок 2.3 – Сторінка налаштувань Advanced Custom Fields

Щоб почати використовувати АСF, у є кілька варіантів: завантажити та встановити безкоштовну версію зі сховища WordPress або придбати версію Pro

та встановити її на своєму веб-сайті.

У безкоштовній версії є доступ до понад 30 типів полів.

У версії Pro можна призначати поля до сторінок користувальницьких параметрів, використовувати поля повторювачів, щоб дозволяти повторювані рядки підполів, отримувати доступ до блоків ACF у редакторі блоків тощо.

2.2.3 Open Server

Open Server – портативний локальний WAMP/WNMP-сервер з багатофункціональною програмою керування та великим вибором плагінів. Представлений пакет програм разом «на коліні» – це не чергова аматорська збірка, це перший повний професійний інструмент, створений спеціально для веб-розробників з урахуванням їх пропозицій і побажань (рис. 2.4).

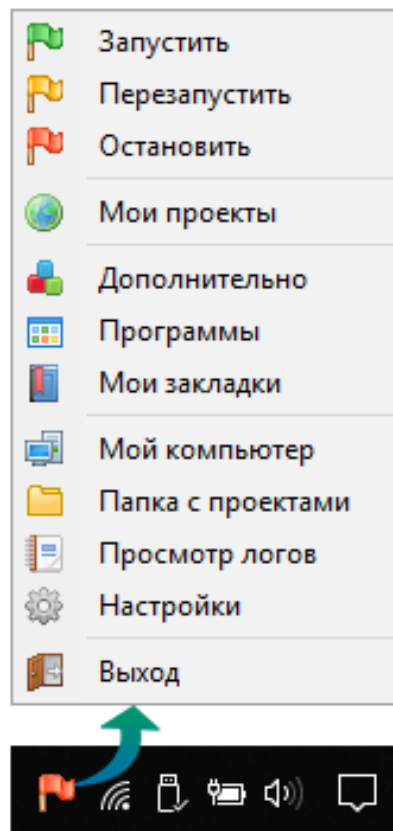


Рисунок 2.4 – Open Server Panel

Для налагодження скриптів у різних середовищах Open Server пропонує на вибір два типи HTTP-серверів одночасно, різні версії модулів PHP і СУБД, а також можливість швидкого перемикання між ними.

Модулі HTTP: Apache 2.2.21 і Nginx 1.0.11.

Модулі СУБД: MySQL 5.1.61, MySQL 5.5.20 і PostgreSQL 9.1.1.

Модулі PHP: PHP 5.2.17 (IMagick 2.2.1, Zend Optimizer 3.3.3, IonCube Loader 4.0.7, Memcache 2.2.4) і PHP 5.3.9 (Imagick 2.3.0, Xdebug 2.1.3, IonCube Loader 4.0). 0. 10, Memcache 2.2.6) (рис. 2.5).

PHP	5.2.17	5.3.29	5.4.45	5.5.38	5.6.40	7.0.33	7.1.33	7.2.34	7.3.33	7.4.30	8.0.22	8.1.9
Amqp		1.4.0	1.4.0	1.4.0	1.4.0	1.9.3	1.9.4	1.10.2	1.11.0	1.11.0	1.11.0	1.11.0
Apcu		4.0.10	4.0.10	4.0.10	4.0.10	5.1.15	5.1.18	5.1.19	5.1.21	5.1.21	5.1.21	5.1.21
Hprose		1.6.5	1.6.5	1.6.6	1.6.6	1.6.6	1.6.8	1.8.0	1.8.0	1.8.0		
Hrtime				0.6.0	0.6.0	0.6.0	0.6.0	0.6.0	0.6.0	0.6.0		
Igbinary		1.2.1	1.2.1	2.0.1	2.0.7	2.0.8	3.0.1	3.2.1	3.2.6	3.2.7	3.2.7	3.2.7
Imagick		3.3.0	3.4.1	3.4.3	3.4.3	3.4.3	3.4.4	3.4.4	3.6.0	3.7.0	3.7.0	3.7.0
IonCube	12.0.1	12.0.1	12.0.1	12.0.1	12.0.1	12.0.1	12.0.1	12.0.1	12.0.1	12.0.1		12.0.1
Mailparse		2.1.6	2.1.6	2.1.6	2.1.6	3.0.2	3.0.3	3.1.1	3.1.2	3.1.3	3.1.3	3.1.3
Memcache	2.2.4	3.0.8	3.0.8	3.0.8	3.0.8	3.0.9	3.0.9	4.0.5.2	4.0.5.2	4.0.5.2	8.1	8.1
Memcached						3.1.3	3.1.4	3.1.5	3.1.5	3.2.0	3.2.0	3.2.0
Mongo		1.6.14	1.6.14	1.6.14	1.6.16							
MongoDB				1.2.9	1.5.3	1.5.3	1.6.0	1.9.0	1.12.0	1.13.0	1.13.0	1.13.0
PDFlib		9.1.1	9.2.0	9.2.0	9.2.0	9.2.0	9.2.0	9.2.0	9.3.1	10.0.0	10.0.0	10.0.0
Rar		3.0.2	3.0.2	3.0.2	3.0.2	4.0.0	4.0.0	4.2.0	4.2.0	4.2.0	4.2.0	4.2.0
Redis		2.2.7	2.2.7	2.2.7	2.2.7	4.2.0	5.1.1	5.3.2	5.3.5	5.3.7	5.3.7	5.3.7
Ssh2							1.2	1.2	1.3.1	1.3.1	1.3.1	1.3.1
Timezonedb		2016.9	2016.9	2017.2	2018.7	2018.7	2019.2	2020.4	2021.5	2021.5	2021.5	2021.5
Uploadprogress		1.0.3.1	1.0.3.1	1.0.3.1	1.0.3.1			1.1.3	2.0.2	2.0.2	2.0.2	2.0.2
Xdebug		2.2.7	2.4.1	2.5.5	2.5.5	2.6.1	2.9.4	3.1.4	3.1.4	3.1.4	3.1.4	3.1.4
Xhprof								2.2.3	2.3.5	2.3.5	2.3.5	2.3.5
Yaml		1.2.0	1.2.0	1.2.0	1.2.0	2.0.4	2.0.4	2.2.1	2.2.2	2.2.2	2.2.2	2.2.2
Zend Optimizer	3.3.3											

Рисунок 2.5 – PHP модулі та додаткові розширення

Чудовий набір інструментів: HeidiSQL, Admin, PHPMyAdmin, PHPPgAdmin, PgAdmin.

Пакет також включає такі елементи, як: Perl, FTP-сервер, Sendmail, Memcached-сервер!

Open Server – це єдиний відомий мені проект, який включає Nginx! Крім того, тут реалізовані зручні правила перезапису посилань через файли .htaccess в корені домену, а PHP працює в режимі True FastCGI (indestructible).

Всі компоненти взяті з офіційних репозиторіїв і завжди оновлюються до останніх версій з кожним оновленням пакета.

Перш за все, хотілося б відзначити, що Open Server – це повністю і повністю портативний сервер. Немає системних служб, багато сміття в реєстрі та system32. Ви можете носити його всюди на флешці (бажано високошвидкісної), запускати на робочому / домашньому комп'ютері, не боячись, що у вас щось не запрацює.

Якщо необхідні системні компоненти відсутні на комп'ютері, Open Server встановить їх самостійно, виберіть у меню [Інструменти – Перший запуск], якщо сервер запускається на комп'ютері вперше.

Я не хочу описувати основні можливості програми, тому що здебільшого вони стандартні для даного типу програмного забезпечення. Зайве говорити, що за допомогою Open Server ви можете запустити / зупинити сервер або відкрити потрібний домен. Вам буде набагато цікавіше дізнатися про конкретні «функції», які роблять Open Server особливим, дійсно особливим:

- 1) детальний перегляд журналів всіх компонентів в реальному часі;
- 2) вибір модулів HTTP, СУБД і PHP у будь-якій комбінації;
- 3) підтримка SSL і кирилических доменів з коробки;
- 4) підтримка псевдонімів або інших покажчиків на домен, а також зручна форма для їх встановлення (привіт шанувальникам мультисайтів у Drupal!);
- 5) створити локальний субдомен без втрати видимості основного домену в Інтернеті;
- 6) доступ до доменів (в один клік) і швидкий доступ до шаблонів конфігурації модулів;
- 7) багатомовний інтерфейс (російська, українська, білоруська, англійська) (див. рис. 2.6).

Програма постійно вдосконалюється, всі достатні запити від користувачів Open Server детально вивчаються і більшість з них реалізуються!

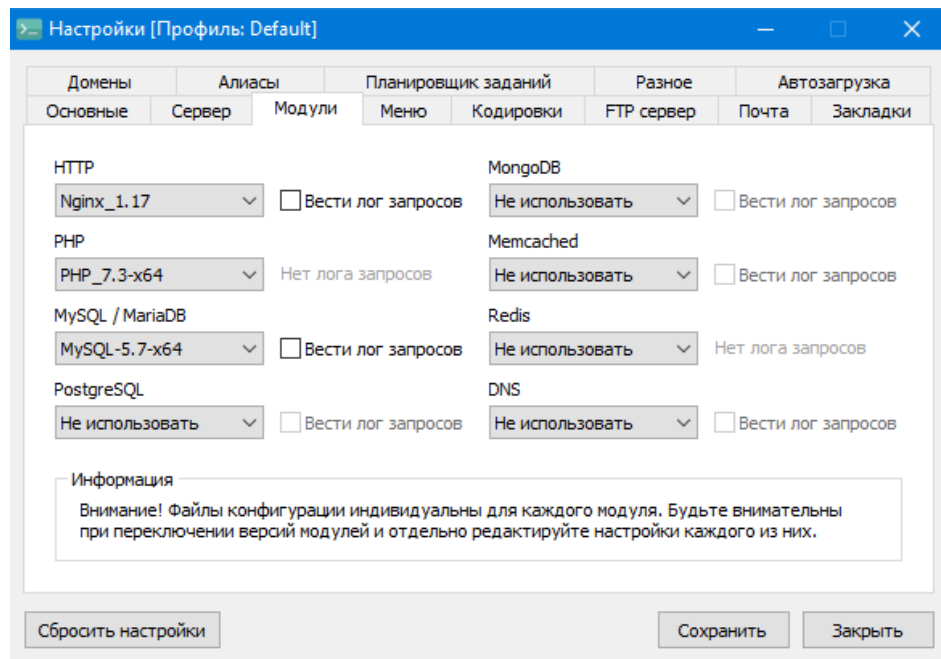


Рисунок 2.6 – Меню налаштувань

2.3 Висновки до другого розділу

Виходячи з основних переваг і недоліків проаналізованої CMS, згаданих вище, а також використовуючи дані досліджень, можна зробити висновок, що система управління контентом WordPress – це дуже ефективна система керування вмістом.

Гнучка практика показує, що дана платформа може бути використана для вирішення більшості завдань, пов'язаних зі створенням веб-ресурсів різного призначення, хоча спочатку ця CMS створювалася як система управління блогами. Завдяки простоті використання та функціональним можливостям WordPress є найкращим рішенням для розробки сайту-портфоліо.

Також виходячи з усього перерахованого вище, Advanced Custom Fields буде кращим рішенням для створення динамічного контенту і подальшої роботи

з ним. Так як цей плагін має кращу підтримку від розробника та ідеальну сумісність з WordPress.

Серед безлічі локальних серверів дійсно важко вибрати свій, найкращим варіантом буде пробувати їх все щоб зупинитися на зручному для себе сервері. На мою думку, найзручніший локальний сервер – це OpenServer, тому що він має великий спектр налаштувань і постійно оновлюється.

3 САЙТ «ALEXANYAN»

3.1 Налаштування середовища розробки

Першим етапом на початку створення сайту є налагодження середовища розробки.

Щоб реалізувати сайт потрібно мати сервер і базу даних. На локальному сервері можна розробити код сайт і налаштувати підключення його до бази даних, але є також є безкоштовні хостинги з уже налаштованими серверами і базами даних.

У цій роботі вибрали хостинг InfinityFree – відмінний безкоштовний хостинг зі зручною панеллю адміністратора і великим функціоналом. Але не дивлячись на всі переваги в цьому хостингу, є також один мінус – це слабка швидкість передачі даних. Через це фактора розробка даного сайту була трохи довше, ніж вона могла б бути на локальному сервері, але це того варто так, як все-таки мені не довелося налаштовувати локальний сервер і підключення до бази даних, а також в кінцевому результаті не довелося робити перенос на хостинг, щоб мій сайт виявився у загальному доступі.

В першу чергу, щоб почати використовувати весь функціонал даного хостингу, потрібно пройти реєстрацію в ньому. Реєстрація досить проста, від мене потрібні були тільки пошта, придуманий мною пароль і пройти верифікацію через лист, відправлений на пошту хостингом (рис. 3.1).

Після реєстрації відразу потрапив в панель користувача, де почався процес створення сайту з першого кроку – вибору доменного імені (рис. 3.2).

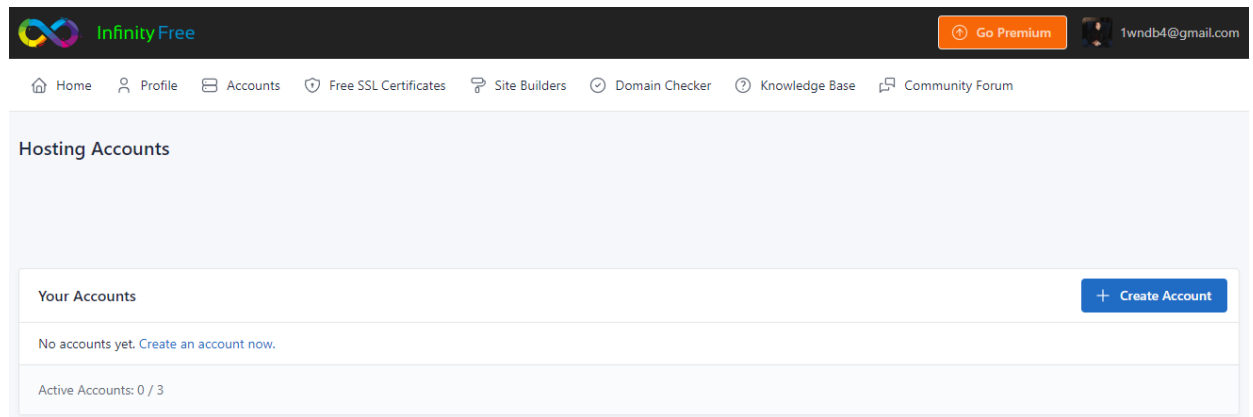


Рисунок 3.1 – Панель користувача

Рисунок 3.2 – Перший крок

Другим кроком було заповнення короткої інформації про сайт та створення облікового запису (рис. 3.3).

Рисунок 3.3 – Другий крок

У завершальному третьому етапі був вибір, перейти в панель користувача або в панель управління. Так як сайт на цьому хостингу ми створювали перший раз, зайшовши в панель користувача, в якій знаходилася вся необхідна інформація по сайту. У цій панелі виявили дані для віддалених підключень до хостингу по FTP з'єднання і дані для підключення до бази MySQL, а також всю іншу не надто важливу інформацію на акаунту і сайту (рис. 3.4).

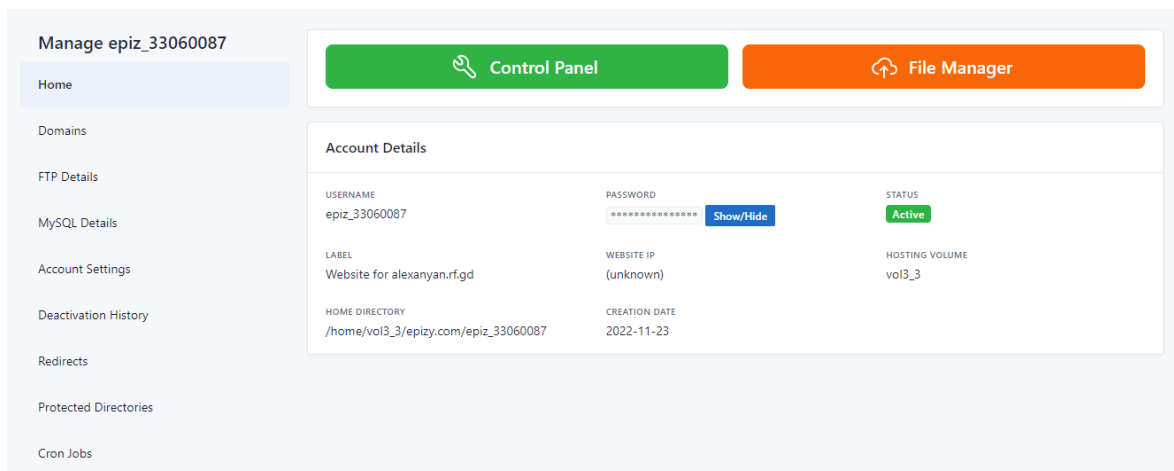


Рисунок 3.4 – Внутрішня панель користувача

Після, хостинг запропонував мені відразу перейти в панель управління, що і зробив. Панель управління в даному сервісі звичайна і поширена в інших сервісах, що не робить її менш зручною для роботи (рис. 3.5).

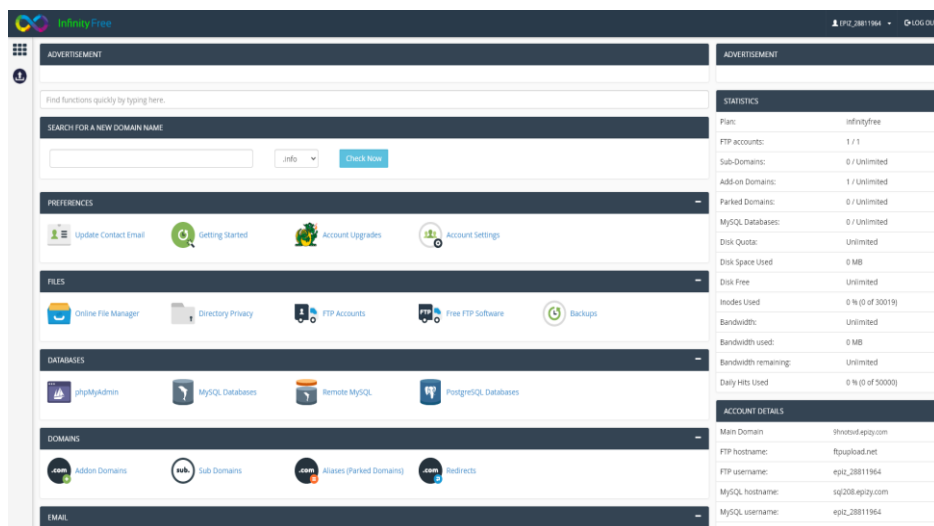


Рисунок 3.5 – Панель управління

Вивчивши панель управління, перейшов до установки CMS. Дану можливість знайшов у вкладці Software. Для мого вибору було запропоновано безліч популярних систем управління вмістом таких як: WordPress, Joomla, AbanteCart і т.п. (рис. 3.6).

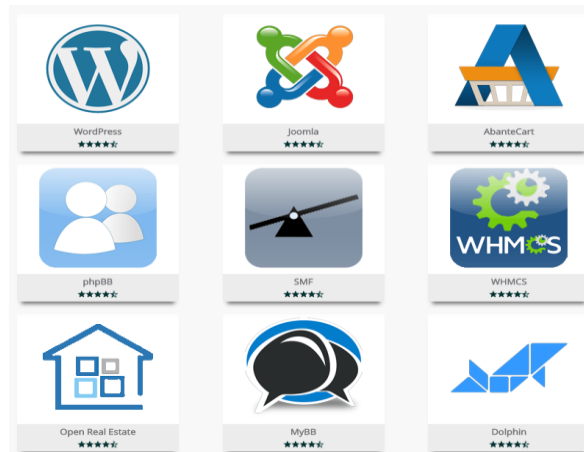


Рисунок 3.6 – Системи управління вмістом

Вибравши в списку WordPress, мене перевело на сторінку заповнення основної інформації про сайт для даної CMS. Мені треба було вибрати версію WordPress яку треба встановити, написати назву сайту і зробити його короткий опис, також заповнив дані для входу в панель адміністратора на сайті (рис. 3.7).

 The screenshot shows the WordPress installation page. At the top, it says 'WordPress' with a five-star rating and lists various versions. Below this is a navigation bar with 'Install' selected. The main content area is titled 'Software Setup' and includes a 'Quick Install' button. Under 'Software Setup', there are two main sections: 'Site Settings' and 'Admin Account'. The 'Site Settings' section has fields for 'Site Name' (filled with 'Alexanyan') and 'Site Description' (filled with 'Alexanyan'), and checkboxes for 'Enable Multisite (WPMU)' and 'Disable WordPress Cron'. The 'Admin Account' section has fields for 'Admin Username' (filled with 'admin'), 'Admin Password' (filled with 'admin'), 'Bad (23/100)', and 'Admin Email' (filled with 'hwndb4@gmail.com').

Рисунок 3.7 – Сторінка встановлення WordPress

Після вдалої установки CMS на хостинг, мені було виведено на екран повідомлення з посиланням на мій сайт. За замовчуванням на сайті була встановлена стандартна тема Twenty Twenty-One, розроблена командою WordPress.

Стандартна тема зручна і буде корисна для більшості користувачів і розробників сайтів на основі WordPress, але вибрав шлях реалізації власної теми. Отже вирішили створити власну тему, щоб повною мірою відчувати всю міць цієї системи керування вмістом і позбудеться від зайвого функціоналу, який мені не потрібно, але який є у всіх комерційних темах (рис. 3.8).

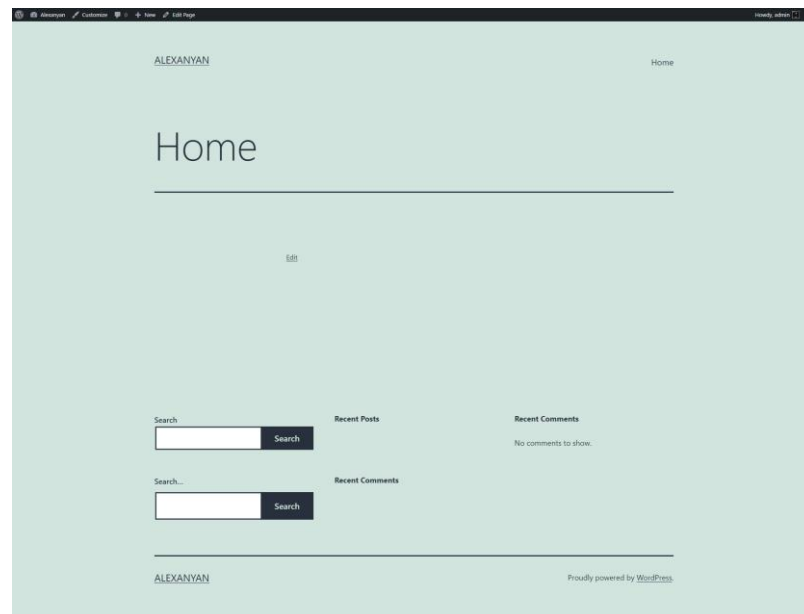


Рисунок 3.8 – Головна сторінка сайту з темою Twenty Twenty-One

3.2 Створення каркаса теми WordPress

Створення власної теми на WordPress також захоплює, як і створення чогось своїми руками в будь-якій сфері життєдіяльності людини. Але такий шлях створення сайту не для новачків. Щоб створити свою тему, потрібно відчувати всю суть даної системи і зрозуміти всі принципи роботи з нею. Маючи досвід роботи з даною системою, приступив до створення теми.

Насамперед потрібно підключитися в редакторі до хостингу через FTP. У редакторі, яким ми користуємося, є окрема вкладка для віддалених підключень на сервер, в якій даний етап грамотно і зручно реалізовано. Ввівши потрібні дані в правильні поля, підключився до сервера віддалено (рис. 3.9).

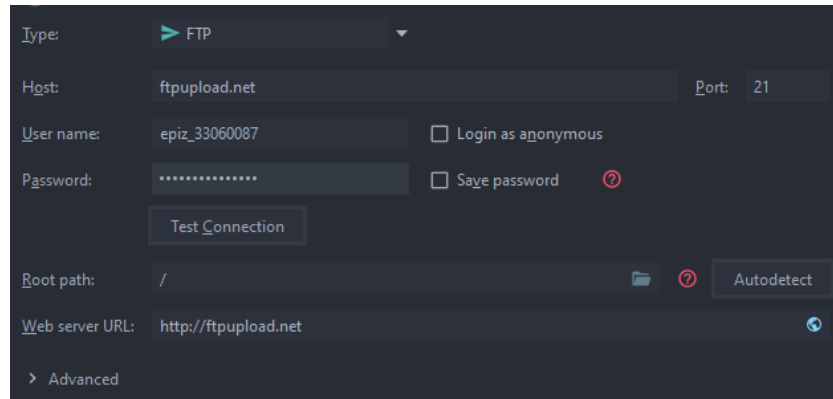


Рисунок 3.9 – Вікно налаштування FTP

Даний сервер має свою структуру папок і файлів, але головне, що дана структура не змінюється в WordPress з роками. Зручно, коли з кожним проектом не доводиться продумувати ієрархію файлів і зв'язку між ними, а користуватися готовим рішенням.

Теми в даній системі лежать завжди по одному шляху – wp-content / themes /. Спочатку в цій папці лежить і встановлена тема, але ми її видалили, з метою економії місця.

Насамперед потрібно створити папку з ім'ям теми, бажано, щоб папка називалася, як і сама тема, але можна і не дотримати свого слова. Після того, як папка створена її потрібно наповнити вмістом, а саме файлами без яких WordPress не знайде і не визначить нашу тему (рис. 3.10).

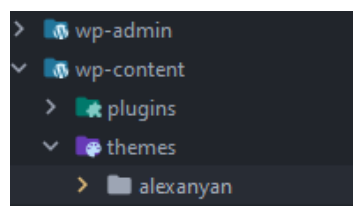


Рисунок 3.10 – Папка створеної теми

Для того щоб система побачила нашу тему потрібно створити два файли, `index.php` і `style.css`. Така особливість закладена розробниками даної CMS.

У файлі `style.css` вказується вся інформація про тему, яка потім буде відображатися в панелі адміністратора. Також в даному файлі можна писати стилі сайту, але робити це неправильно. Для написання стилів краще виділити окрему папку і файл (рис. 3.11).

```
1  /*
2  Theme Name: Alexanqan
3  Author: Nastua Alexanqan
4  Author URI:
5  Description: Alexanqan - Resume & Personal Portfolio WordPress Theme
6  Version: 1.0
7  Text Domain: alexanqan
8  */
```

Рисунок 3.11 – Зміст файлу `style.css`

У файлі `index.php` вказується каркас шаблону, який буде відповідати за показ усіх сторінок сайту. Зазвичай в цьому файлі прописуються виведення всіх записів сайту на сторінку. Коли створили два головних файлу, система побачила тему, але користі від неї зараз немає і потрібно продовжити заповнювати тему файлами (рис. 3.12).

```
1  <?php
2  /** The main template file ... */
16
17  get_header(); ?>
18  <div id="primary">
19      <div id="main">
20          <?php if (have_posts()) {
21              // Load posts loop.
22              while (have_posts()) {
23                  the_post();
24                  the_content();
25              }
26          } ?>
27      </div>
28  </div>
29
30  <?php
31  get_footer();
32
```

Рисунок 3.12 – Зміст файлу `index.php`

Так як вже працював з WordPress, заповнити тему файлами не склало труднощів. Кожен може оформляти тему за власним смаком, немає ніяких рамок і обмежень в цьому плані. В першу чергу створили папку assets в якій будуть зберігається файли css і javascript, картинки і шрифти. Всередині цієї папки теж краще розділити весь вміст внутрішніми папками, а саме розділити стилі, скрипти, картинки і шрифти на окремі папки (рис. 3.13).

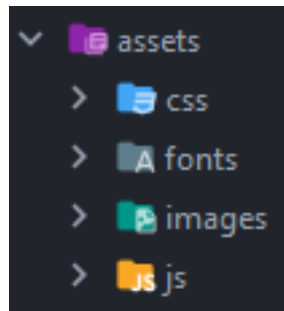


Рисунок 3.13 – Папка assets

Наступні файли є обов'язковими при створення нової теми – це файли footer.php, header.php, front page.php (home.php, page.php), 404.php.

Файл front page.php або в деяких випадках home.php, page.php. Такі назви не випадкові так, як це файл належить головній (домашньої) сторінки на сайті. Назва файлу з перерахованих вибирає сам розробник так, як чітких вказівок і обмежень в цьому плані немає.

Файл 404.php відповідає за відображення сторінки помилки при будь-яку помилку користувача на сайті пов'язаної з адресним рядком. Наприклад, якщо користувач введе в адресному рядку назву сторінки якої немає на сайті, то сайт йому виведе саме сторінку до якої прив'язаний файл 404.php.

Файли header.php і footer.php відповідають за шапку і підвал сайту відповідно. У WordPress як і всюди в сучасному програмуванні прийнято розділяти код на якомога більші частини з метою подальшого повторного їх використання. Саме з цього файли відповідають за шапку і підвал сайту виносять в окремі файли.

Наступним кроком потрібно створити папку в якій будуть знаходитися всі файли сторінок сайту, окрім файлу головної сторінки який вже створив в корені теми. Назвати таку папку можна просто `templates`.

Я вже писав, що в WordPress прийнято розділяти код на частини. Для цього потрібно створити папку, в якій будуть лежати файли з частинами коду, які будуть використовуватися на різних сторінках. Ми створили папку і назвали її `template-parts`.

В результаті цих дій у мене зібрався каркас теми з файлів і папок створених мною (рис. 3.14).

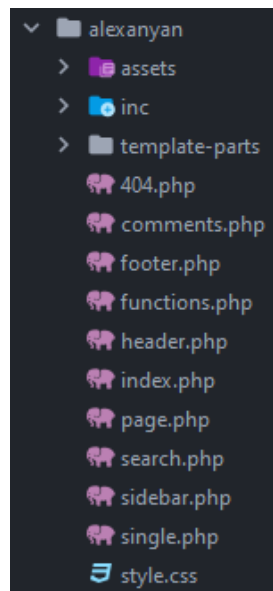


Рисунок 3.14 – Заповнена тема alexanyan

3.3 Налаштування панелі адміністратора

Щоб потрапити в панель адміністратора потрібно в адресному рядку сайту дописати `/wp-admin`. Після того як дописали в адресному рядку `/wp-admin`, мене перевело на сторінку з формою для авторизації. У цю форму ввів доступи які створював на етапі налаштування хостингу, коли встановлював WordPress (рис. 3.15).

Рисунок 3.15 – Сторінка авторизації

Насамперед перевірили чи працює тема створена мною і бачить її WordPress. Для цього перейшли у вкладку Зовнішній вигляд / Теми, тут виводиться список тем встановлених на сайті. На моєму сайті є тільки одна тема alexanyan створена мій. У ній ми якраз бачимо всю інформацію, що писав у файлі style.css (рис. 3.16).

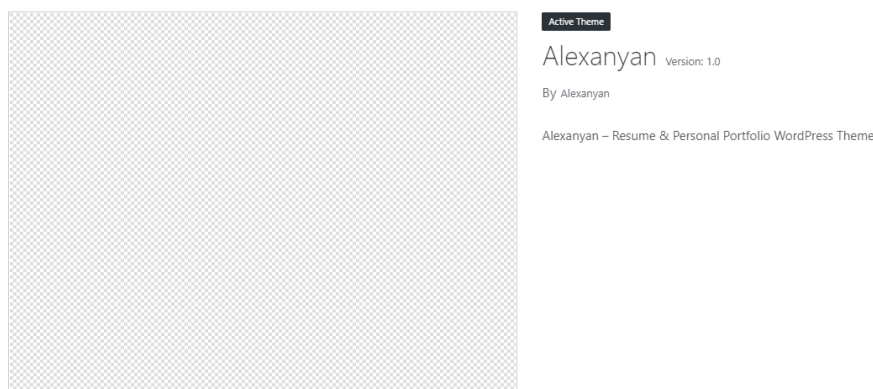


Рисунок 3.16 – Тема сайту

Наступним кроком у мене була установка і настройка плагінів на сайт. Модулі в WordPress грають роль помічників, як для розробника, так і для користувача. За допомогою плагінів звичайний користувач-власник сайту може

розширити його функціонал не вдаючись до допомоги розробника. Але є ряд плагінів, які буде складні для розуміння звичайному користувачеві, адже вони більшою мірою спрямовані на розробників.

Щоб потрапити в меню плагінів ми перейшли у вкладку Модулі. Встановили кілька плагінів які допоможуть мені розширити функціонал сайту і зробити його більш оптимізованим і захищеним (рис. 3.17).

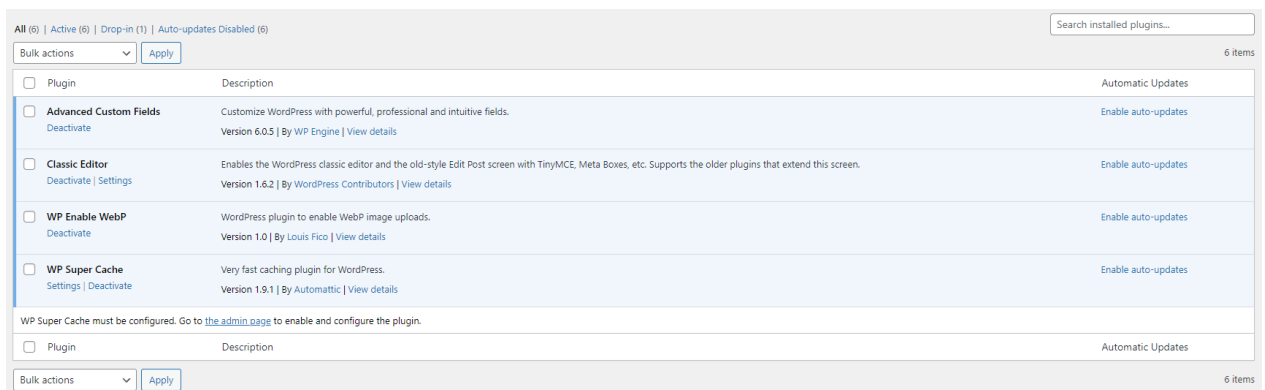


Рисунок 3.17 – Вкладка плагіни

Перший і найголовніший плагін з моєму проєкті – Advanced Custom Fields PRO. Цей плагін дає можливість розробнику створювати поля на сторінках для виведення динамічного контенту. Видів полів є дуже велика кількість і всі вони потрібні в певний момент, але використали лише невелику частину з них.

Другий плагін – Classic Editor. Він включає класичний редактор WordPress і екран редагування повідомлення в старому стилі з TinyMCE, Meta Boxes і т. Д. Підтримує старі плагіни, що розширюють цей екран. Даний плагін зручний і підходить тим розробникам, хто вже давно займається розробкою сайтів на WordPress, щоб повернути звичний CMS.

Третій плагін – WP Enable WebP. Цей плагін включає підтримку зображень нового формату WebP в панелі адміністратора, щоб вони відображалися як звичайні картинки, а не у вигляді невідомого файлу. Даний формат зображень оптимізує сайт не навантажуючи сторінку при відображенні

її користувачеві.

Четвертий плагін – WP Super Cache, він дозволяє оптимізувати, швидко і просто налаштувати кешування файлів сайту.

Наступним кроком потрібно було створити вкладку в меню адміністратора, щоб додавати в неї поля які не будуть прив'язані до певної сторінці, а будуть належать частинам сайту які можуть дублюватися, такі як шапка і підвал сайту. Для цього зайшли в файл functions.php. У цьому файлі пишуть код, який розширює базовий функціонал WordPress. Ми написали шматок коду, в якому створюю нову вкладку з назвою ThemeSettings для плагіна ACF (рис. 3.18).

```

139 if (function_exists( function: 'acf_add_options_page')) {
140     acf_add_options_page([
141         'page_title' => 'Theme General Settings',
142         'menu_title' => 'Theme Settings',
143         'menu_slug' => 'theme-general-settings',
144         'capability' => 'edit_posts',
145         'redirect' => false,
146     ]);
147 }

```

Рисунок 3.18 – Функція створення вкладки ThemeSettings

У підсумку після збереження файлу, в меню адміністратора з'явилася нова вкладка ThemeSettings (рис. 3.19).

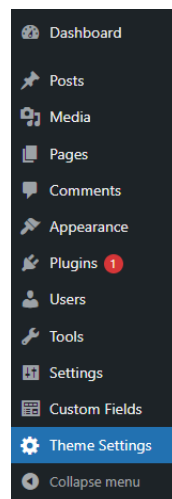


Рисунок 3.19 – Вкладка Theme Settings

3.4 Створення сторінки сайту

Сторінки в WordPress можна створити у внеску сторінки. Створили необхідну мені сторінку – Home. Кожна сторінка в панелі адміністратора виглядає однаково. У ній є назва сторінки і дрібні настройки (рис. 3.20).

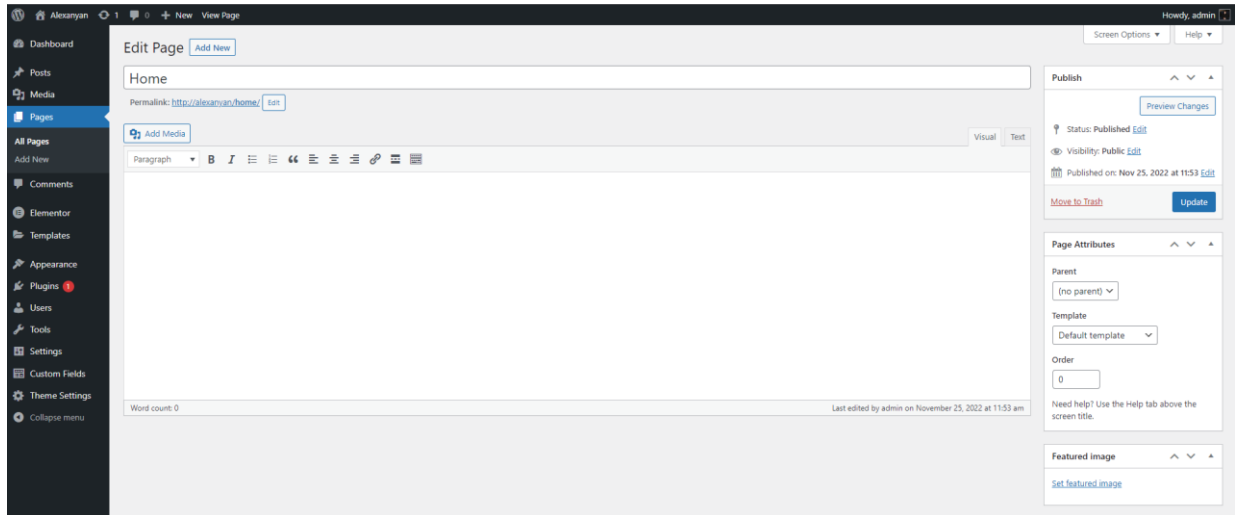


Рисунок 3.20 – Головна сторінка у панелі адміністратора

Найголовніша настройка якій торкалися постійно атрибути сторінки, в ній можна вибрати шаблон (файл) до якого прикріпити цю сторінку (рис. 3.21).

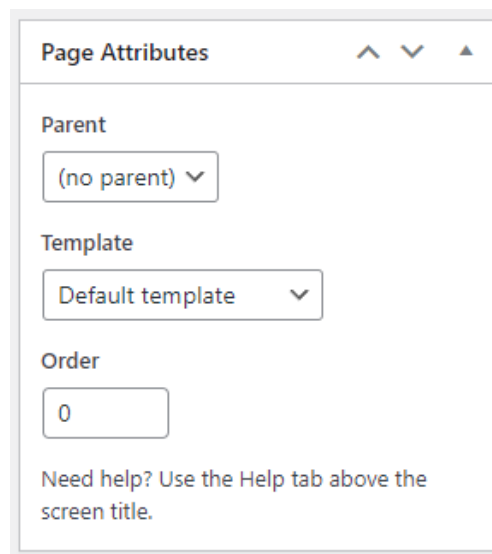


Рисунок 3.21 – Атрибути сторінки

Щоб назва файлу з'явилася в списку шаблонів, потрібно в файлі сторінки написати на початку певний текст за яким WordPress визначає файл. Після чого можна побачити назву в списку шаблонів (рис. 3.22).

```
<?php
/*
 * Template Name: Home
 */
?>
```

Рисунок 3.22 – Приклад визначення сторінки у файлі

Вибравши шаблоном сторінки якийсь файл, на сторінці буде відображатися тільки те, що написано в файлі.

Після створення сторінок ми відразу вирішили зробити меню в панелі адміністратора, щоб потім просто вивести його в файл. Для того, щоб зробити меню потрібно перейти у вкладку Зовнішній вигляд / Меню. У цій вкладці можна створити і налаштувати меню. У меню можна додавати сторінки, записи і власні пункти. Створили три меню, для кожної шапки сайту (рис. 3.23).

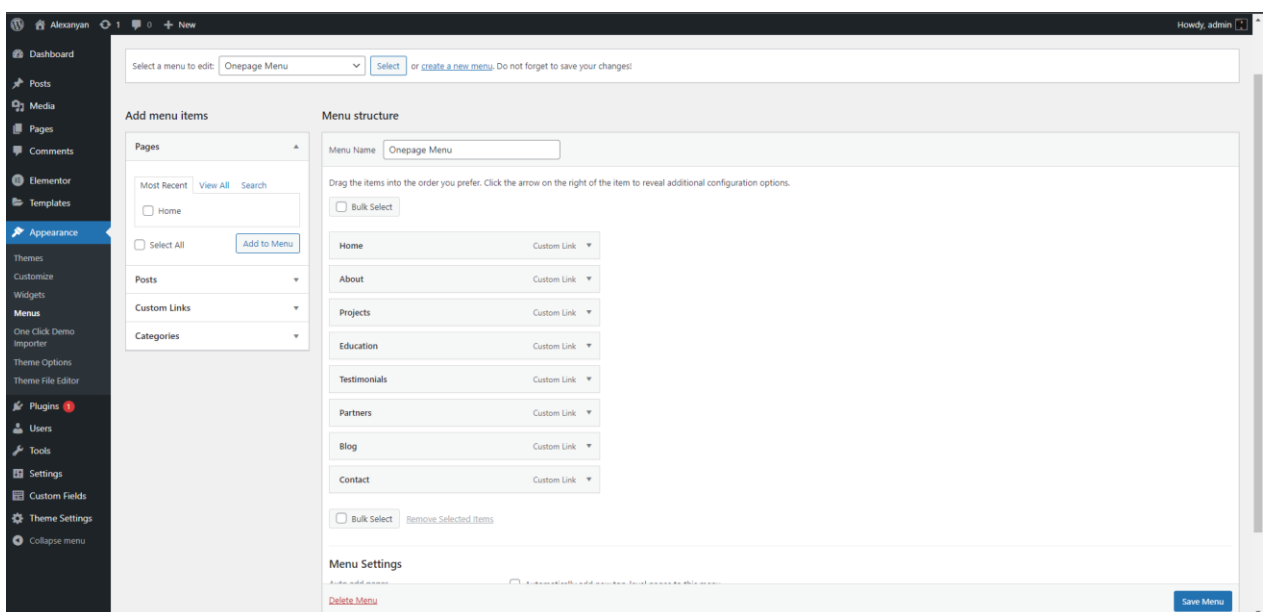


Рисунок 3.23 – Вкладка налаштування меню сайту

Для виведення меню в файлі потрібно написати одну сходинку коду. У коді меню виводиться за назвою (рис. 3.24).

```
<?php wp_nav_menu(['menu' => 'Menu Main']); ?>
```

Рисунок 3.24 – Код виведення меню на сторінку

3.5 Створення полів для сторінки

Після створення сторінок, наступним кроком стає створення полів для сторінок їх висновок в файл і заповнення в панелі адміністратора.

Щоб створювати поля потрібно перейти у вкладку Групи полів. Тут потрібно створити групи полів для кожної сторінки окремо і для створеної мною вкладки Theme Settings. При створенні груп полів потрібно вказати назву групи, бажано, щоб назва була схоже на те до чого ці поля будуть створені. А також потрібно вказати до чого дані поля будуть прикріплені, можна вказати як один варіант, так і кілька (рис. 3.25).

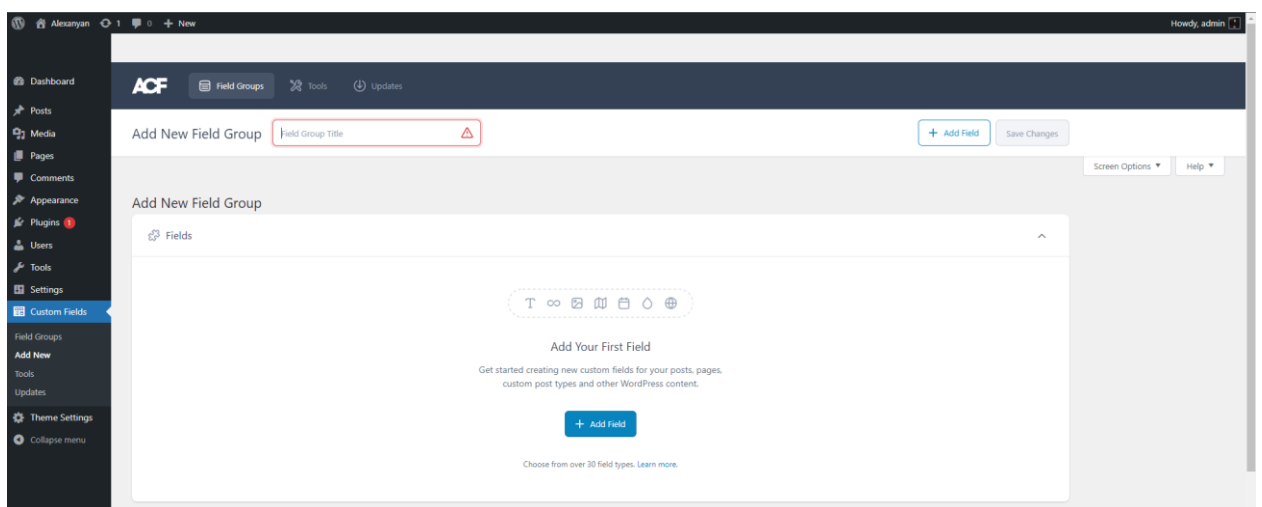


Рисунок 3.25 – Групи полів

Прикріпили кожну групу полів до своїх сторінок, для кожної групи по одній. І одну групу до вкладки ThemeSettings. Вийшло чотири групи полів (рис. 3.26).

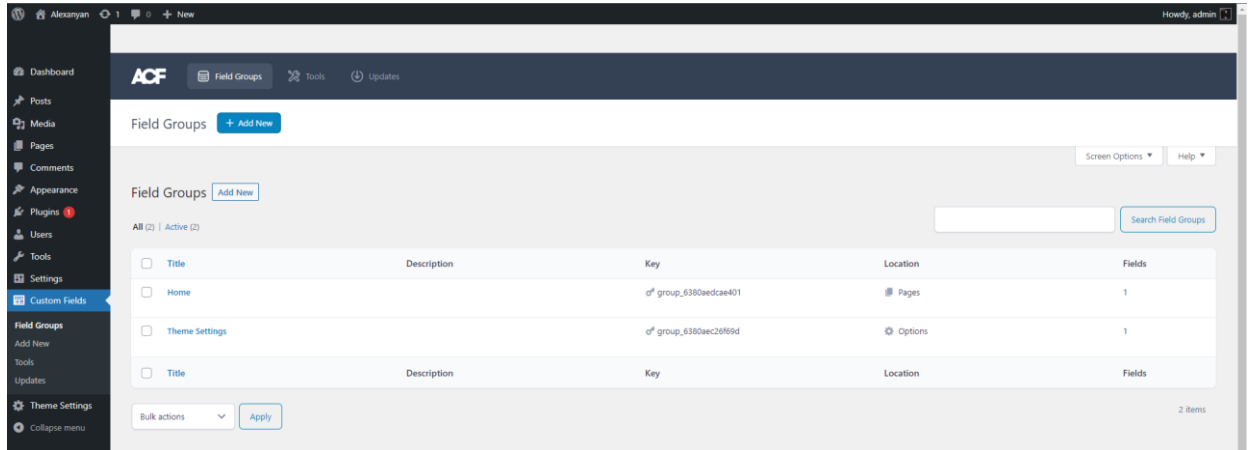


Рисунок 3.26 – Групи полів

Створивши групу полів в них потрібно створити поля які потрібні будуть на сторінці. Є велика кількість різних видів полів (рис. 3.27).

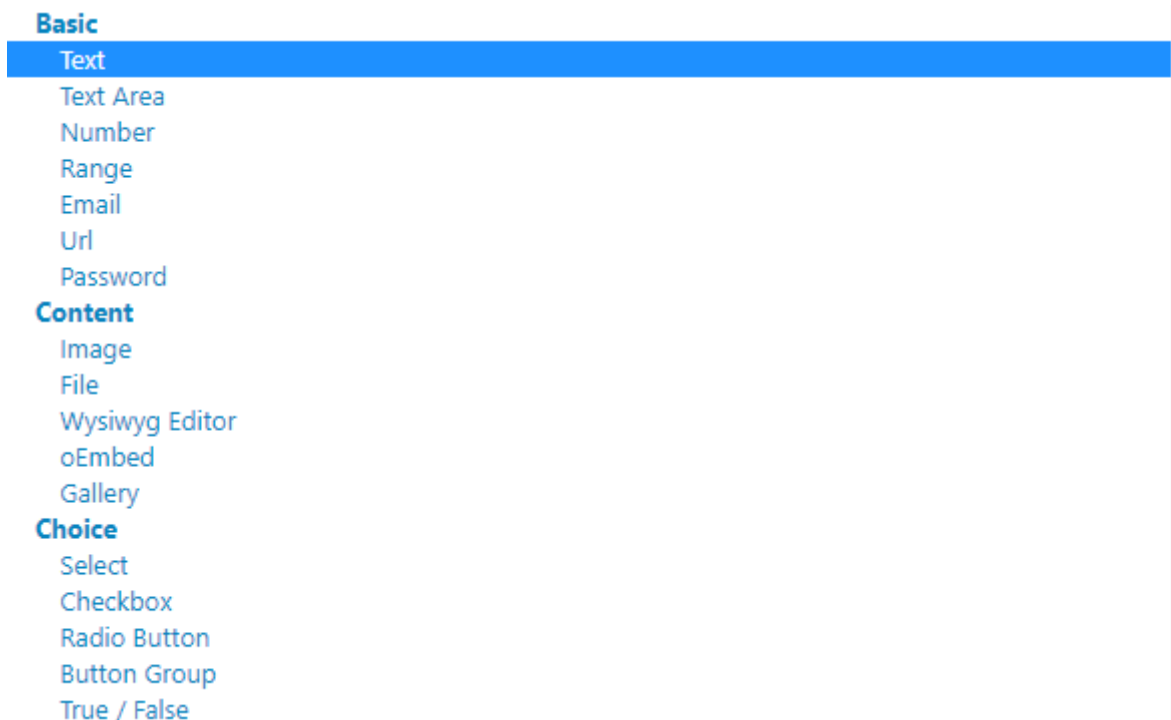


Рисунок 3.27 – Перелік видів груп полів

Використано деякі види полів такі, як: текст, область тексту, зображення, редактор WordPress, посилання, вкладка, повторювач. Всі поля мають свій індивідуальний функціонал. Пояснення до того як працює кожної поле можна знайти в документації на сайті <https://www.advancedcustomfields.com/resources/>. Головне що потрібно знати для роботи з полями – це те, що у поля є ім'я, яке відображається на сторінці в панелі адміністрації, а також є ключ, він потрібен щоб у файлі звертатися до вмісту поля через нього (рис. 3.28).

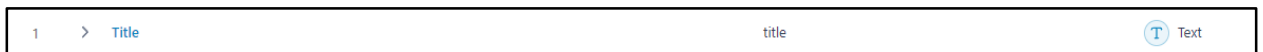


Рисунок 3.28 – Приклад поля

Створили поля для всіх сторінок і приступили до їх заповнення контентом. Кожна група полів відображається на сторінці в панелі адміністратора, як окремий блок з полями (рис. 3.29).

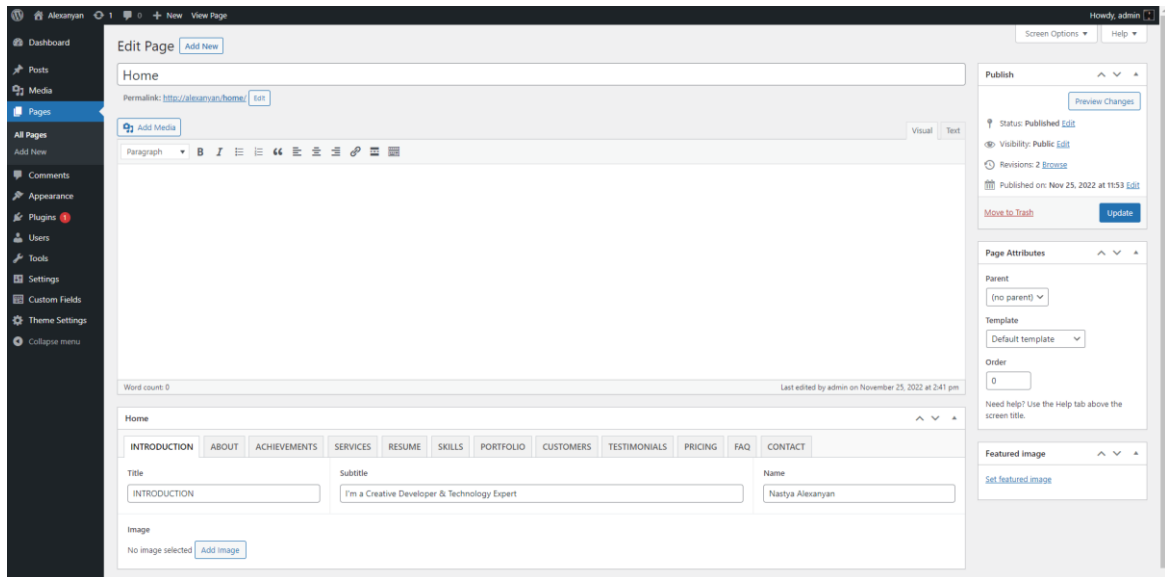


Рисунок 3.29 – Групи полів у головній сторінці

Залежно від типу поля в нього можна вводити: текст, картинку, відео, посилання, файл і навіть HTML код.

Далі після заповнення полів потрібно їх вивести в файли. Щоб виводить поля в файли потрібні знання мови PHP. Головна команда за допомогою якої звертаються до полів показана на малюнку (рис. 3.30).

```
<?php echo get_field( selector: 'introduction_title') ?>
```

Рисунок 3.30 – Код виведення полів на сторінку

Щоб виводити поля, які підключені до вкладки ThemeSettings в панелі адміністратора, потрібно вказати додатковим параметром звідки ми беремо поле, а саме з вкладки options (рис. 3.31).

```
<?php echo get_field( selector: 'header_logo', post_id: 'option') ?>
```

Рисунок 3.31 – Приклад виведення поля з вкладки ThemeSettings

Вводить команду потрібно в дужки визначають мову PHP, таким чином можна вбудовувати поля в звичайний HTML. У команді пишеться ключ поля. Для кожного поля є своя область доступності, обмежена вона тим файлом або сторінкою до якого підключена. Якщо висновок у файлі правильно зроблений, то на сторінці в браузері буде виведено вміст полів (рис. 3.32).

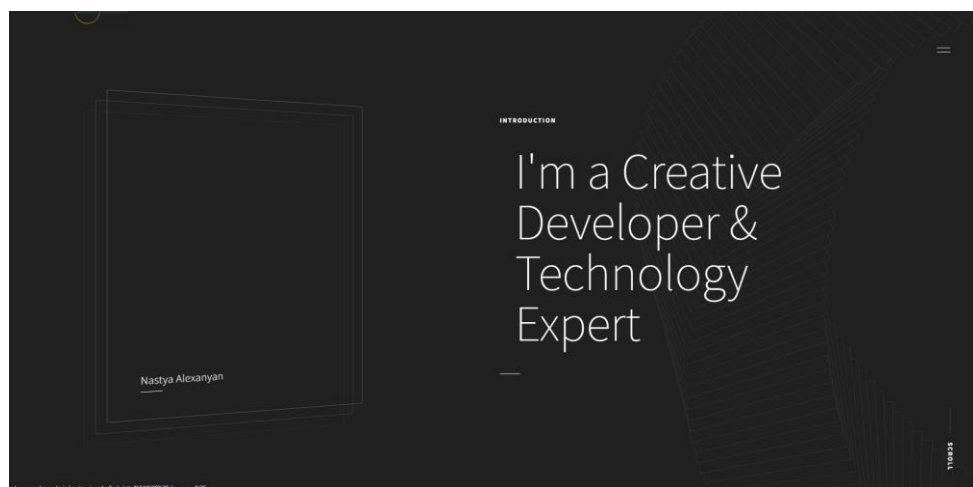


Рисунок 3.32 – Інтерфейс головної сторінки

Таким чином вивели все поля в усі створені мною файли. Після цього написали стилі і скрипти для зовнішнього вигляду сайту та анімацій.

3.6 Огляд інтерфейсу сайту

Для створення інтерфейсу сайту ми використовували css і javascript. Тематика сайту – портфоліо для креативних людей, щоб допомогти їм донести свої роботи до суспільства у всій красі і бездоганності.

Перша головна сторінка зустрічає користувача коли він потрапляє на сайт. На головній сторінці зібрана вся важлива інформація про людину, його навички, приклади робіт і контактні дані.

У першій частині зроблено висновок псевдоніма і короткої цитати яка описує сферу діяльності майстра. А також його фото. Для більшого ефекту занурення, перша секція повністю займає весь екран користувача (рис. 3.33).

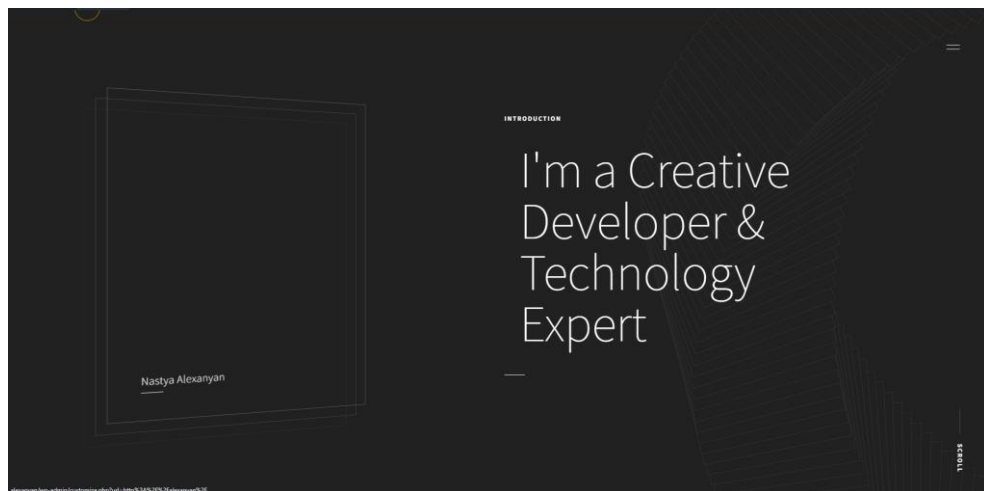


Рисунок 3.33 – Перша секція головної сторінки

Друга секція показує навички майстра і відсотковість володіння ними, а також кількість виконаних замовлень і клієнтів. Також коли користувач потрапляє в зону видимості цієї секції, відбувається анімація руху, шкала навичок рухається від нуля до певної точки (рис. 3.34).

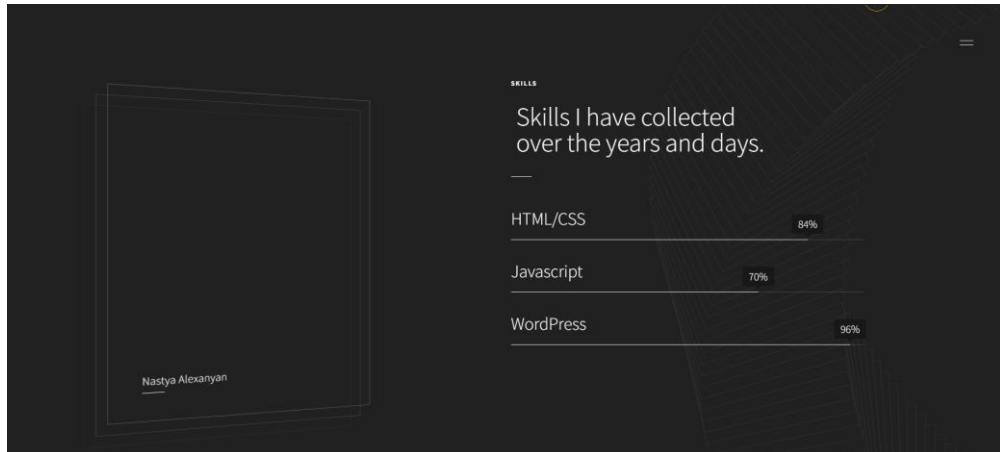


Рисунок 3.34 – Друга секція головної сторінки

У третій секції користувач бачить блоки з декількома прикладами робіт. Також користувач може відкрити зображення на весь екран, щоб детально побачити приклад роботи. Ця секція займає більше місця ніж один екран, тому користувачеві потрібно переглянути вниз, щоб побачити всі приклади роботи, які майстер розмістив на сайті. Також тут реалізована можливість вибору іншої картинки при натисканні на блок, таким чином користувач зможе побачити дві і більше різних картинок відразу (рис. 3.35).

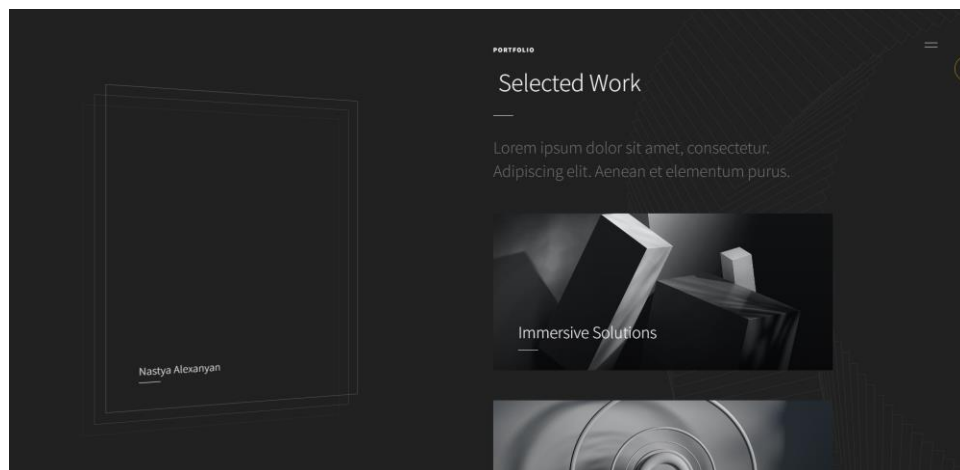


Рисунок 3.35 – Третя секція головної сторінки

У четвертій секції розташовується коротка інформація про майстра, його навички та життєвий шлях. Ця секція також як і більшість інших розрахована на те, щоб займати весь екран користувача інформацією, але також можна за

бажанням додати набагато більше, щоб вмістити весь необхідний текст про себе (рис. 3.36).

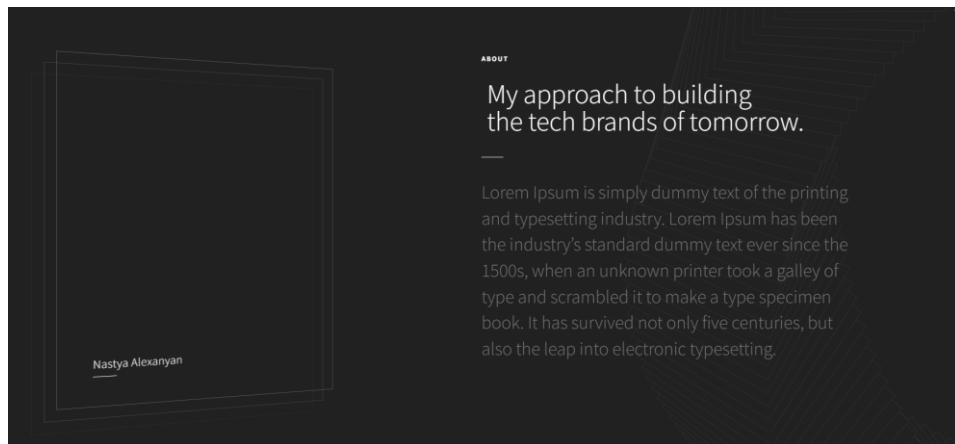


Рисунок 3.36 – Четверта секція головної сторінки

П'ята секція наповнена інформацією про досягнення автора, кількість його проектів, нагород і так далі. Також у секції є другий заголовок та короткий опис для надання користувачеві інформації про секцію (рис. 3.37).

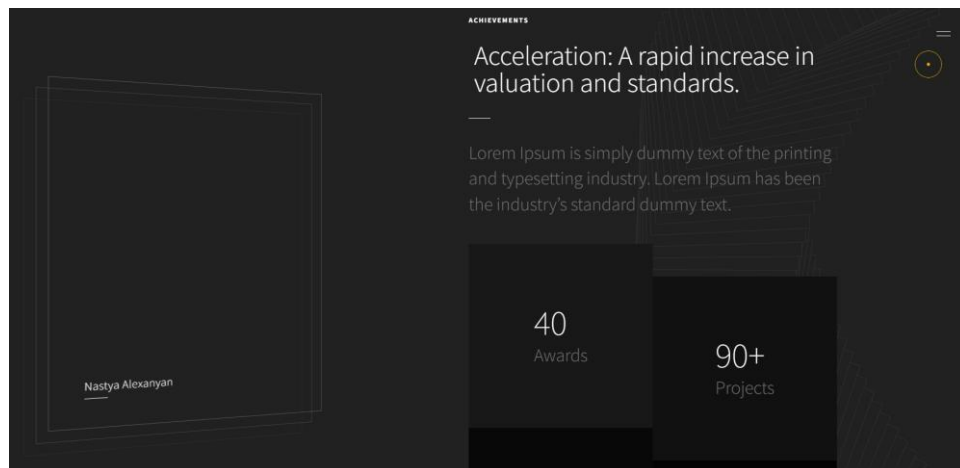


Рисунок 3.37 – П'ята секція головної сторінки

Шоста секція складається із заголовка, короткого опису та дропдаунів. У дропдауни можна розмістити більш розширену інформацію про сферу діяльності майстра та його роботи. Можна створити велику кількість дропдаун для вміщення всієї потрібної інформації (рис. 3.38).

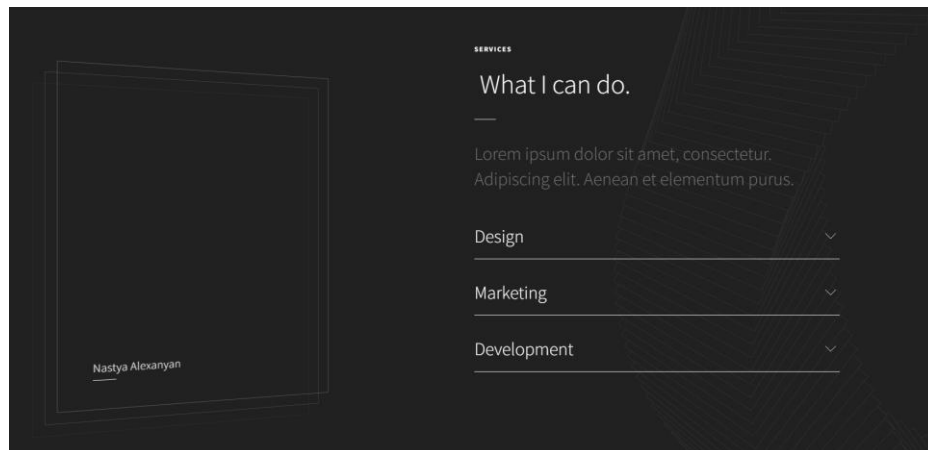


Рисунок 3.38 – Шоста секція головної сторінки

У сьомій секції знаходяться відгуки клієнтів про майстра. Адмін сайту сам розміщує всі відгуки за згодою клієнта, оскільки клієнт може вільно побачити свій відгук, коли прийде на сайт знову (рис. 3.39).

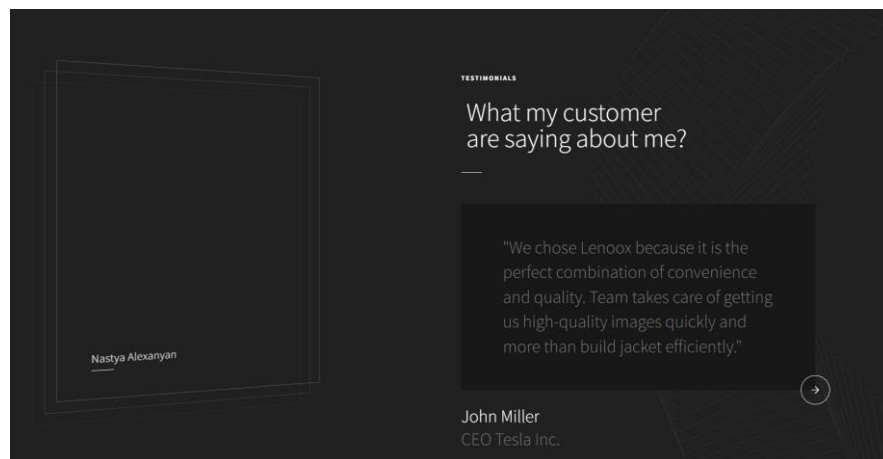


Рисунок 3.39 – Сьома секція головної сторінки

В останній секції знаходиться форма зворотного зв'язку. Реалізовано її за допомогою плагіна WPForms. Валідація та відправлення відбувається без перезавантаження сторінки, оскільки плагін дозволяє це зробити з коробки. Також налаштований антиспам із міццю плагіна Akismet Spam Protection (рис. 3.40).

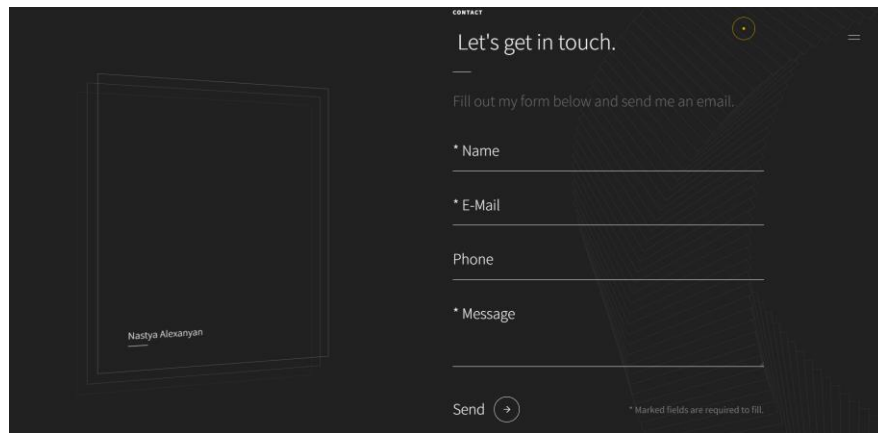


Рисунок 3.40 – Восьма секція головної сторінки

Відкривши спливаюче меню, користувач може побачити посилання, які будуть його переміщати по секціях головної сторінки. Також тут розташовані посилання для переходу до соціальних мереж. (рис. 3.41).

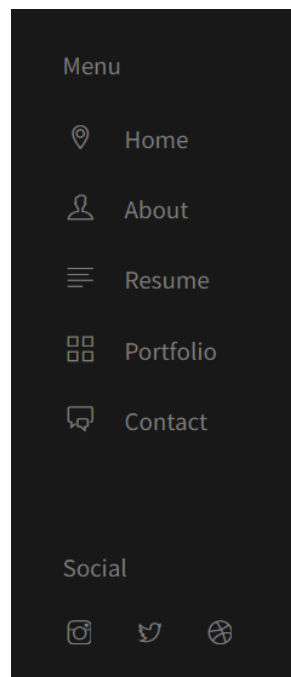


Рисунок 3.41 – Меню головної сторінки

ВИСНОВКИ

У роботі був створений сайт Alexanyan з динамічним контентом, для розміщення своїх навичок і прикладів робіт.

У першому розділі було проаналізовано CMS WordPress, її переваги та особливостями перед іншими системами керування вмістом. Обрана CMS має такі переваги, як зручність використання і швидкість.

У другому розділі були розглянуті етапи створення сайту, від настройки CMS до створення і наповнення АСФ.

На мою думку, сайт можна поліпшити в деяких деталях:

- додати просунуті анімації на сторінки;
- поліпшити продуктивність сайту, перенісши його на інший хостинг

і зменшивши код.

Даний сайт може стати відмінним рішенням для людей, хто хоче показати свої навички і приклади робіт користувачам мережі, якщо продовжити вдосконалювати його і вводити нові функціональні рішення.

ПЕРЕЛІК ПОСИЛАНЬ

1. WordPress. URL: <https://uk.wikipedia.org/wiki/WordPress> (дата звернення: 29.09.2022).
2. Система керування вмістом. URL: https://uk.wikipedia.org/wiki/Система_керування_вмістом (дата звернення: 21.11.2022).
3. Редактори коду. URL: <https://techrocks.ru/2018/09/10/top-6-popular-free-source-code-editors/> (дата звернення: 26.10.2022).
4. Total number of Websites. URL: <https://www.internetlivestats.com/total-number-of-websites/> (дата звернення: 10.11.2022).
5. Plumley G. WordPress 24-Hour Trainer. Indianapolis: Wiley, 2011. 373 p.
6. Hussey T. Using WordPress. Indianapolis: Que, 2010. 351 p.
7. Coyier C., Starr J. Digging Into WordPress. Perishable Press, 2011. 442 p.
8. Blakeley Silver T. WordPress Theme Design. Packt Publishing, 2008. 224 p.
9. Мейер Э. CSS – каскадные таблицы стилей. Подробное руководство: 3-е изд. / пер. с англ. Н. Шатохиной. СПб: «Символ Плюс», 2008. 576 с.
10. Каслдайн Э., Шарки К. Изучаем JQUERY / пер. с англ. В. Черник. СПб.: «Питер», 2012. 400 с.
11. Веллинг Л., Томсон Л. Разработка Web-приложений с помощью PHP и MySQL / пер. с англ. А. Моргунов. М.: Издательский дом «Вильямс», 2017. 768 с.
12. Крейн Д., Паскарелло Э., Даррен Д. Аjax в действии / пер. с англ. В. В. Вейтмшш, А. В. Назаренко. М.: Издательский дом «Вильямс», 2006. 640 с.
13. Дари К., Бринзаре Б., Черchez-Тоза Ф., Бусика М. Аjax и PHP: разработка динамических веб-приложений / пер. с англ. А. Киселева. СПб: «Символ Плюс», 2006. 336 с.

14. Joomla. URL: https://docs.joomla.org/Portal:Learn_More/ru (дата звернення: 09.10.2022).
15. Backdrop CMS. URL: https://en.wikipedia.org/wiki/Backdrop_CMS (дата звернення: 12.10.2022).
16. Wix. URL: <https://uk.wix.com/> (дата звернення: 07.10.2022).
17. HubSpot. URL: <https://www.hubspot.com/> (дата звернення: 07.10.2022).
18. Magnolia. URL: <https://www.magnolia-cms.com/> (дата звернення: 07.10.2022).
19. CSS. URL: <https://uk.wikipedia.org/wiki/CSS> (дата звернення: 07.12.2022).
20. JavaScript. URL: <https://uk.wikipedia.org/wiki/JavaScript> (дата звернення: 22.10.2022).
21. Основне про PHP. URL: <http://www.makewebsites.narod.ru/php.html> (дата звернення: 20.10.2022).