

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет _____ Магістерської та
_____ аспірантської підготовки

Кафедра _____ інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Створення інтернет-магазину на MVC фреймворку
WP Emegre

Виконав студент 2 року групи МК- 2
спеціальності 122 Комп'ютерні
науки та інформаційні технології

_____ Копельчук Сергій Володимирович

Керівник _____ д.х.н., професор
Кругляк Юрій Олексійович

Консультант _____

Рецензент _____

Одеса 2018

АНОТАЦІЯ

Дипломна робота на тему: «Створення інтернет-магазину на MVC фреймворку WP Emegre», автор – студент групи МК2 Копельчук Сергій Володимирович

Актуальність: актуальність дипломної роботи зумовило ріст популярності інтернет комерції.

Мета: розробка та побудова структури інтернет-магазину за допомогою інструментів WordPress.

Задачі: вивчити літературу, що стосується теми дослідження, розглянути існуючі аналоги, розробити інтернет магазин та побудувати структуру MVC.

Об'єкт дослідження: інтернет-комерція.

Предмет: інтернет-магазин.

Методи: аналіз веб-інструментів, порівняння аналогів, розробка діючого проекту.

Результат: створення масштабованої моделі діючого проекту для роботи інтернет магазину, функціональність якої можна розширювати.

Рекомендації: проект призначений для реалізації інтернет-комерції.

Робота включає: 78 сторінок, 26 рисунків, 2 додатки, використаних джерел – 13.

Ключові слова: ІНТЕРНЕТ-МАГАЗИН, WORDPRESS, WP EMEGRE, WOOCOMMERCE, MVC, SEO, ІНТЕРНЕТ-КОМЕРЦІЯ.

SUMMARY

Graduation work on the topic: "Creating an online store on the MVC frameworks WP Emegre", author - student of the group MK2 Kopelchuk Sergey Vladimirovich
Actuality: the relevance of the thesis led to the growth of the popularity of Internet commerce.

Aim: Developing and building an online store with WordPress tools.

Objectives: to study literature related to the topic of research, to consider existing analogues, to develop an online store and to build the structure of MVC.

Object of research: Internet commerce.

Subject: online store.

Methods: analysis of web tools, comparisons of analogs, development of an active project.

The result: creating a scalable model of an active project for the online store, whose functionality can be expanded.

Recommendations: The project is intended for the implementation of Internet commerce.

The work includes: 78 pages, 26 images, 2 additions, used sources - 13.

Keywords: ONLINE STORE, WORDPRESS, WP EMEGRE, WOOCOMMERCE, MVC, SEO, INTERNET COMMERCE.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 ЗАГАЛЬНИЙ РОЗДІЛ.....	11
1. Аналітичний огляд існуючих рішень	11
1.1 Основні відомості про веб-розробку	11
1.1.2 Комерціалізація Інтернету	12
1.1.3 Інтернет-магазин.....	13
1.1.4 Основні цілі у розробці веб-сайту електронної комерції	13
2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ	16
2.1 Технічне завдання.....	16
2.1.1 Призначення розробки	16
2.1.2 Вимоги до програмної документації	16
2.1.3 Стадії й етапи розробки програмного проекту.....	17
2.1.4 Порядок контролю і прийому.....	17
2.2 Веб – моделювання.....	18
2.3 Опис програмної мови PHP.....	19
2.3.1 Опис ліцензування PHP	21
2.3.2 Опис використання PHP	21
2.3.3 Опис безпеки PHP	22
2.4 Опис системи управління контентом	24
2.4.1 CMS WordPress	25
2.5 Шаблон проектування MVC.....	32
2.5.1 Історія розвитку MVC.....	32
2.5.2 Відмінності опису концепції шаблону	33
2.5.3 Призначення MVC.....	33
2.5.4 Концепція MVC	34
2.5.5 Умовно-обов'язкові модифікації	34
2.6 UML діаграма WordPress	35
3 ОПИС ПРОЕКТНОГО І ТЕХНІЧНОГО РІШЕННЯ	37
3.1 Мікрофреймворк WP Emerge	37
3.2 Хостінг та сервер	38
3.3 Встановлення WordPress та WP Emerge.....	40
3.4 Структура проекту.....	42
3.5 Застосування основного функціоналу.....	45
4 ПРОСУВАННЯ САЙТУ (SEO).....	50
4.1 Просування сайту на WordPress.....	50

4.2 Наявність WWW у просуванні сайту	51
4.3 Встановлення плагіну для SEO на WordPress	52
4.4 Налаштування перманентних посилань.....	54
4.5 Використовування ключових слів в текстах.....	55
4.6 Теги Title.....	56
4.7 Збільшення CTR за допомогою Meta Description.....	57
4.8 Meta Keywords	59
4.9 Теги H1, H2, H3	60
4.10 Кількість слів	61
4.11 Використання коротких URL-адрес	62
4.12 Використання розумних назв для зображень	63
4.13 Використання переваг HTTPS для розкрутки	64
4.14 Опис Google Knowledge Graph.....	64
4.15 Опис Google Analytics	65
5 УПРАВЛІННЯ ПРОЕКТОМ	67
ВИСНОВКИ.....	73
ДОДАТОК А КОД YOAST SEO ДЛЯ SOCIAL MARKUP.....	77
ДОДАТОК Б UML ДІАГРАМА БАЗИ ДАНИХ WORDPRESS	78

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Скорочення

API – Application Programming Interface
BSD – Berkeley Software Distribution
CDA – Content Delivery Application
CMA – Content Management Application
CMS – Content Management System
CSS – Cascading Style Sheets
DVB – Digital Video Broadcasting
FTP – File Transfer Protocol
GPL – General Public License
HTML – HyperText Markup Language
HTTP – HyperText Transfer Protocol
ISN – Initial Sequence Number (початковий порядковий номер)
LAMP – Linux Apache MySQL PHP
MVC – Model-View-Controller
PHP – PHP: Hypertext Preprocessor
SEO – Search Engine Optimization
TCP – Transmission Control Protocol
UDP – User Datagram Protocol
UML – Unified Modeling Language
VPN – Virtual Private Network
WAN – Wide Area Network
WCM(S) – Web Content Management (System)
WYSIWYG – What You See Is What You Get

Терміни

MySQL – система управління реляційною базою даних із відкритим кодом
WordPress – це безкоштовна та відкрита CMS на основі PHP та MySQL

ВСТУП

Інтернет-бізнес починає привертати все більше значення в сучасному світі, і в нашій країні відповідно, у зв'язку з тенденцією до загальної глобалізації економіки. Інтернет і електронна торгівля грають в цьому процесі одну з найважливіших ролей. Поява абсолютного нового виду зв'язку зумовило перегляд підприємців процесу організації бізнесу.

Якщо 15 років тому компанія виходила в мережу виключно для підтримки свого престижу, то сьогодні – це як мінімум ще один крупний канал продаж. В кризисні 2008-2009 рр. підприємства традиційного сектора торгівлі були змушені скоротити штат і обсяги продажів, а інтернет-сектор навпаки збільшив торговий оборот за цей період на 46%.

Інтернет відкриває широкі можливості не тільки для великих підприємств. Багато початківців бізнесменів відразу відкривають свою справу в Інтернеті, так як для цього потрібна менша сума первинних інвестицій, ніж у сфері традиційної торгівлі. А компанії, які працюють за принципом прямих поставки, вкладають кошти виключно в веб-сайт і рекламу.

Західні покупці, а в слід за ними і українські, все частіше купують товари, не виходячи з дому, посилаючись на нестачу часу, невеликий асортимент товарів у традиційних магазинах і високі ціни. В Україні в початку 2000-х тільки інтузіасты зробили електронні покупки, сьогодні, за даними статистики, хоча б раз це зробили 4 з 5 чоловік. Розвиток бізнесу в інтернеті на сьогоднішній день крайне перспективне напрямом для підприємців, що і обумовлює актуальність даної роботи.

Розробка веб-сайтів електронної комерції – це нелегке завдання. У нього є свої проблеми. Оскільки велика кількість веб-користувачів відвідують сайт електронної комерції щодня, тому потрібно його підтримувати. Для цього може знадобитися залучити експертів веб-розробників електронної комерції від надійної та відомий компанії з розробки електронної комерції веб-сайтів, що забезпечує інтуїтивний процес веб-розробки електронної комерції з легкістю.

Ціль випускної роботи полягає в моделюванні та створенні інтернет-магазину, вивченні успішного зарубіжного та вітчизняної досвіду та розробки на цій основі інтернет-магазину.

Ця мета досягається вирішенням наступних завдань:

- виявляти генезис електронної комерції, її основні передумови та тенденції розвитку;

- охарактеризувати загальні принципи побудови інтернет-магазинів;
- виділити ключові фактори успіху;
- розробка концепції інтернет-магазину;
- моделювання архітектури веб-додатку;
- вибір інструментів для розробки інтернет магазину.

При написанні даної роботи використані CMS WordPress, мікрофреймворк WP Emerge для побудови MVC структури проекту і плагін WooCommerce для побудови електронної комерції сайту.

1 ЗАГАЛЬНИЙ РОЗДІЛ

1. Аналітичний огляд існуючих рішень

1.1 Основні відомості про веб-розробку

Веб-розробка – це робота, пов'язана з розробкою веб-сайту для Інтернету (всесвітньої павутини) або інтрамережі (приватна мережа). Веб-розробка може варіюватися від розробки простої єдиної статичної сторінки простого тексту до складних Інтернет-додатків (веб-додатків) для електронного бізнесу та послуг соціальної мережі. Більш повний перелік завдань, до яких часто відноситься веб-розробка, може включати також веб-інжиніринг, веб-дизайн, розробку веб-контенту, взаємодію з клієнтами, сценарії на стороні клієнта / сервер, веб-сервер та конфігурація безпеки мережі та розвиток електронної комерції. Серед веб-професіоналів, "веб-розробка", як правило, стосується основних аспектів, не пов'язаних із розробкою веб-сайтів: написання розмітки та кодування. Останнім часом веб-розробка мала на увазі створення систем управління контентом (CMS).

Ці CMS можуть бути створені з нуля, фірмового або з відкритим вихідним кодом. Загалом, CMS діє як проміжне програмне забезпечення між базою даних та користувачем через браузер. Основним достоїнством CMS є те, що це дозволяє нетехнічним людям внести зміни до свого веб-сайту без технічних знань.

Для великих організацій та підприємств, команди веб-розробників можуть складатися із сотень людей (веб-розробників) і виконувати стандартні методи, такі як Agile методології, при розробці веб-сайтів. Менші організації можуть вимагати лише одного постійного або контрагента розробника або вторинного розпорядження пов'язаними робочими місцями, такими як графічний дизайнер або технік інформаційних систем. Веб-розробка може бути спільним зусиллям між відділами, а не доменом призначеного відділу. На даний час існує три види спеціалізації, які характеризують веб-розробників: інтерфейсний розробник, розробник програмного забезпечення та розробник повного стека. Розробники інтерфейсу мають справу з макетом і візуальними зображеннями веб-сайту, тоді як розробники заднього виду вирішують функціональність веб-сайту. Задні розробники будуть програмувати за допомогою функцій веб-сайту, який збиратиме дані[1].

1.1.2 Комерціалізація Інтернету

З комерціалізації Інтернету веб-розробка стала зростаючою галуззю. Зростання цієї галузі керується бізнесом, який бажає використовувати свій веб-сайт, щоб продавати продукти та послуги клієнтам.

Є багато інструментів із відкритим кодом для розробки веб-сторінок, таких як BerkeleyDB, GlassFish, LAMP (Linux, Apache, MySQL, PHP) і Perl / Plack. Це дозволило мінімально знизити вартість навчання веб-розробки. Ще одним фактором зростання промисловості стало зростання простого у використанні програмного забезпечення для веб-розробки WYSIWYG, таке як Adobe Dreamweaver, BlueGriffon та Microsoft Visual Studio. Знання мови розмітки HyperText (HTML) або мов програмування все ще потрібні для використання такого програмного забезпечення, але основи можна швидко вивчати та впроваджувати.

Постійно зростаючий набір інструментів і технологій допомогли розробникам створювати більш динамічні та інтерактивні веб-сайти. Крім того, веб-розробники тепер допомагають надавати програми як веб-сервіси, які традиційно доступні лише як додатки на настільному комп'ютері. Це дало багато можливостей децентралізувати інформацію та розповсюдження ЗМІ. Приклади можна побачити з появою хмарних сервісів, таких як Adobe Creative Cloud, Dropbox та Google Drive. Ці веб-сервери дозволяють користувачам взаємодіяти з додатками з багатьох розташувань, замість того, щоб бути прив'язаними до певної робочої станції для середовища їх застосування.

Приклади драматичних перетворень у сфері комунікації та торгівлі, керовані веб-розробкою, включають електронну комерцію. Інтернет-аукціонні сайти, такі як eBay, змінювали спосіб, який споживачі шукають та купують товари та послуги. Інтернет-магазини, такі як Amazon.com і Buy.com (серед багатьох інших), перетворили досвід для покупок і покупок для багатьох споживачів.

Іншим прикладом трансформаційного спілкування під керівництвом веб-розробки є блог. Веб-застосунки, такі як WordPress та Movable Type, створили блог-середовища для сайтів. Зростання використання систем керування контентом з відкритим кодом та CMS підприємства покращило вплив веб-розробки при взаємодії та спілкуванні в Інтернеті.

1.1.3 Інтернет-магазин

Розробка веб-сайтів також вплинула на особисту мережу та маркетинг. Веб-сайти більше не просто інструменти для роботи або для комерції, але служать більш широко для спілкування та соціальних мереж. Веб-сайти, такі як Facebook і Twitter, надають користувачам платформу для спілкування та організації з більш особистим та інтерактивним способом залучення громадськості.

У цю цифрову епоху люди більше залежні від розумних гаджетів, ніж будь-коли. У той же час промисловість електронної комерції сьогодні швидко розвивається. Як повідомляє Statista, приблизно 60% людей сьогодні витрачають свої гроші на інтернет-магазини. Існує велика частина інтернет-покупців серед веб-користувачів у всьому світі. Саме тому підприємства вважають за краще робити інвестиції в розробку веб-сайтів електронної комерції. Таким чином, вони розробляють свій власний веб-сайт електронної комерції для збільшення продажів.

1.1.4 Основні цілі у розробці веб-сайту електронної комерції

У цьому розділі представлено 5 речей, які слід знати перед розробкою веб-сайту електронної комерції.

Правильна платформа електронної комерції та гарна тема. Це один з найважливіших факторів, які слід враховувати перед розвитком веб-сайту електронної комерції. Спершу веб-розробникам електронної комерції потрібно вибрати правильну платформу електронної комерції для цілей розвитку. Справді, вже існує низка платформ електронної комерції, а саме. WooCommerce, Drupal, Magento, Shopify, OSCommerce, Joomla і т.д., з яких є можливість вибору. Гарний вибір платформи електронної комерції в кінцевому підсумку заощадить багато часу та грошей.

Крім того, право вибору теми також відіграє важливу роль у розвитку електронної комерції. Це насправді важливо для успіху будь-якого бізнесу електронної комерції. Відповідно до статистики W3Tech, 1/4 всіх веб-сайтів електронної комерції не виконуються завдяки поганій темі дизайну для свого веб-сайту.

Фактично, на ряді веб-сайтах електронної комерції, що поєднували блог чи портфоліо функціональності, для збільшення взаємодії з користувачами. Це також допомагає компаніям електронної комерції підтримувати довгострокові

відносини зі своїми клієнтами. Крім того, Google першими розміщує ті сайти, які містять блоги. Таким чином, ваш рейтинг електронної комерції веб-сайту також покращиться.

SEO (пошукова оптимізація). SEO відіграє важливу роль у розробці веб-сайтів електронної комерції. Щоб запустити успішний бізнес електронної комерції, спочатку продажі. SEO може допомогти збільшити продажі вашої електронної комерції веб-сайт і, таким чином, дозволяє вам високо стояти на цій конкурентній арені. Крім того, якщо будь-який користувач шукає ваш веб-сайт електронної комерції, він з'явиться у верхній частині пошукових запитів за допомогою хорошого SEO. Таким чином, це збільшує видимість та конверсії Вашого сайту електронної комерції.

Таким чином, все більше і більше клієнтів автоматично потраплять до вас, коли веб-сайт електронної комерції розміститься на найвищих веб-сторінках. За даними обстеження Статисти, було встановлено, що прибіл. 35 найбільших веб-сайтів електронної комерції не відображаються на перших 10 веб-сторінках. Таким чином, користувачі ніколи не будуть прагнути перейти на ці веб-сайти.

Responsive Design. Мобільні користувачі ростуть з кожним днем, і в той же час, веб-сайти на електронній комерції мають великий обсяг трафіку. Це показує велику кількість людей, які використовували для онлайн-магазину через мобільні пристрої. За статистикою Mediabistro, 2/4 продажі веб-сайту електронної комерції поставляються з мобільних телефонів. Таким чином, стає необхідним оптимізувати веб-сайт електронної комерції як для мобільних, так і настільних систем[2].

Відповідний веб-дизайн може допомогти вам у цьому. Це робить ваш веб-сайт електронної комерції доступним для кожного типу пристрою. Завдяки цьому розробники електронної комерції знаходять процес розробки легко. Крім того, чуйний веб-сайт електронної комерції є найважливішим інструментом успіху вашого веб-сайту електронної комерції.

Хороша маркетингова стратегія. Після завершення розробки веб-сайту електронної комерції вам необхідно продати свій веб-сайт електронної комерції. Для цього ви повинні дотримуватися деяких правил маркетингу, щоб залучити все більше і більше клієнтів. Насправді, існує багато способів виходу на ринок вашої електронної комерції. Завдяки надійному агентству з маркетингу ви можете продати свій сайт ідеально.

Проте кожен маркетинговий метод має свої плюси і мінуси. Але з гарною маркетинговою стратегією та гарним дизайном, ваш бізнес електронної

комерції може досягти успіху.

Безпека. Кожен розробник електронної комерції гарантує, що вся інформація на своєму сайті електронної комерції безпечна для створення довіри серед своїх клієнтів. Існують різні методи захисту. Ваш веб-сайт електронної комерції повинен підтримувати SSL, щоб шифрувати інформацію, яка потребує захисту. Це дійсно важливо для користувачів, які використовують дебетові та кредитні картки, і, таким чином, вводять інформацію про платіж, як адресу, ім'я, номер телефону, електронна пошта тощо.

Відповідність PCI є ще одним важливим методом безпеки для підприємств електронної комерції, які приймають цифрові платежі. Фактично, за даними опитування Делойт, ок. 40% веб-сайтів електронної комерції відповідає стандартам PCI.

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ

2.1 Технічне завдання

Даний розділ представляє собою технічні вимоги до створення веб-сайту електронної комерції. Дані вимоги обов'язкові для врахування під час розроблення технічного завдання на програмний виріб.

При розробці технічних вимог було:

- 1) розробка та створення моделі веб-додатку;
- 2) розробка веб-сайту електронної комерції.

2.1.1 Призначення розробки

Повне найменування системи: «Створення інтернет-магазину на MVC фреймворку WP Emerge».

Областю використання проекту є розробка веб-сайту електронної комерції з метою використовувати в якості інтернет магазину, торгівля продуктами обраними клієнтом.

2.1.2 Вимоги до програмної документації

Кожна стадія проектування завершується складанням відповідних документів. Тому важливим елементом проектування програмних додатків є оформлення програмної документації. Виняток може становити нескладних програм із коротким життєвим циклом і невеликою трудомісткістю.

Зміст програмних документів:

- специфікація – перелік і призначення всіх файлів програмного виробу, включаючи файли документації;
- текст програми – запис кодів програми та коментарі до них;
- програма і методика випробувань – перелік і опис вимог, які повинні бути перевірені в ході випробування програми, методи контролю;
- технічне завдання – документ, в якому викладаються призначення і область застосування програми, вимоги до програмного виробу, стадії і терміни розробки, види випробувань;
- пояснювальна записка – обґрунтування прийнятих і застосованих технічних і техніко-економічних рішень, схеми та опис алгоритмів, загальний опис роботи програмного виробу.

До програмних документів віднесено також документи, що забезпечують функціонування та експлуатацію програм – експлуатаційні документи:

- опис мови – містить синтаксис і семантику мови;
- керівництво з технічного обслуговування – містить відомості для застосування;
- тестових і діагностичних програм при обслуговуванні технічних засобів.

2.1.3 Стадії й етапи розробки програмного проекту

Розробка повинна бути виконана в три стадії:

- 1) розробка технічного завдання;
- 2) робочі проектування;
- 3) впровадження.

На першій стадії повинен бути виконаний етап розробки: погодження та затвердження цього технічного завдання.

На стадії робочого проектування повинні бути виконані перераховані нижче етапи робіт:

- 1) розробка програми;
- 2) розробка програмної документації;
- 3) випробування програми.

Зміст робіт по етапах.

На етапі розробки технічного завдання виконуються такі роботи:

- 1) постановка завдань;
- 2) визначення та уточнення вимог до технічних засобів;
- 3) визначення вимог до програми;
- 4) визначення стадій, етапів і термінів розробки програми та документації на неї;
- 5) вибір мов програмування;
- 6) узгодження та затвердження технічного завдання.

2.1.4 Порядок контролю і прийому

Перевірка документації програми здійснюється самим замовником із залученням сторонніх експертів, здатних засвідчити факт відповідності створеного програмного продукту всім пунктам технічної документації, включаючи технічне завдання і технічний проект.

Іспити і тестування програми повинні проводитися в процесі створення програми самим розроблювачем:

- 1) з використанням контрольних тестів, що дозволяють домогтися перевірки працездатності;
- 2) із залученням сторонніх неофіційних бета-тестерів, що у процесі тестування програмного продукту повинні повідомляти розроблювачеві всі знайдені помилки і похибки в роботі програми.

Іспити і тестування відбуваються після завершення створення програми замовником:

- 1) з використанням перевірочних тестів;
- 2) в організації замовника із залученням сторонніх експертів.

2.2 Веб – моделювання

Веб-моделювання (як правило, модемне веб-розробка) – це галузь веб-розробки, яка вирішує конкретні питання, пов'язані з проектуванням та розробкою великомасштабних веб-додатків. Зокрема, він зосереджений на дизайнерських нотаціях та візуальних мовах, які можуть бути використані для реалізації надійних, добре структурованих, зручних та підтримуваних веб-додатків. Розробка веб-сайту, що потребує даних, має визначати його характеристики за різними ортогональними абстракціями. Основними ортогональними моделями, що беруть участь у розробці складних веб-застосунків, є структура даних, склад вмісту, шляхи навігації та модель презентації.

На початку розробки веб-сторінок було нормальним для доступу до веб-додатків, створюючи щось без уваги до стадії розробки. В останні роки фірмам веб-дизайну було багато проблем з управління своїми веб-сайтами, оскільки процес розвитку розвивався та ускладнював інші програми. Інструменти веб-розробки допомогли спростити веб-додатки, що потребують даних, за допомогою генераторів сторінок. Microsoft Active Server Pages та Java Server Pages Java Server допомогли вивести вміст і користуватися запрограмованими шаблонами.

Для моделювання веб-додатків розроблено декілька мов та позначень. Серед них можна навести:

- HDM (Hypertext Design Model) – W2000;
- RMM (Remote monitoring and management);
- OOHDM (Object Oriented Hypermedia Design Method);
- мова потоку моделювання взаємодії IFML(Interaction Flow Modeling

Language), прийнята групою управління об'єктами OMG (Object Management Group) у березні 2013 року;

- T-Araneus: Management of temporal data-intensive Web sites;
- Strudel;
- Tiramisu;
- WebML (Web Modeling Language);
- Gera;
- розширення веб-додатків UML;
- веб-розробка на основі UML (UWE – UML-based web engineering);
- ACE (Abstract Common Engineering);
- WebArchitect;
- OO-H (Object-Oriented Hierarchy).

Однією з основних дискусійних місць для цієї дисципліни є семінар із моделювання веб-майстрів (MDWE), що проводиться щорічно спільно з Міжнародною конференцією з веб-технологій (ICWE)[3].

2.3 Опис програмної мови PHP

PHP: Препроцесор гіпертексту (або просто PHP) – це скриптова мова на стороні сервера, призначена для розробки веб-сайтів, а також використовується як загальномова мова програмування. Його спочатку було створено Расмусом Лерддорфом в 1994 році, еталонна реалізація PHP тепер випускається The PHP Group. PHP спочатку стояв на Персональній домашній сторінці, але тепер це означає рекурсивний ініціалізм PHP: препроцесор гіпертексту.

PHP-код може бути вбудований в HTML-код, або його можна використовувати в поєднанні з різними системами веб-шаблонів, системами керування веб-контентом та веб-структурами. PHP-код, як правило, обробляється інтерпретатором PHP, реалізованим у вигляді модуля на веб-сервері або як виконуваний файл інтерфейсу Common Gateway Interface (CGI). Веб-сервер поєднує в собі результати інтерпретованого та виконаного PHP-коду, який може бути будь-яким типом даних, включаючи зображення, з створеної веб-сторінкою. PHP-код також може бути виконаний за допомогою інтерфейсу командного рядка (CLI) і може бути використаний для реалізації автономних графічних додатків.

Стандартний інтерпретатор PHP, який підтримує Zend Engine, є вільним програмним забезпеченням, випущеним під ліцензією PHP. PHP широко переноситься і може бути розгорнуто на більшості веб-серверів майже на кожній

операційній системі та платформі безкоштовно.

Мова PHP розвивався без письмової формальної специфікації чи стандарту до 2014 року, причому початкова реалізація виступала в якості де-факто стандарту, який інші реалізації мали намір дотримуватися. Починаючи з 2014 р. Робота йшла над створенням офіційної специфікації PHP.

Розробка PHP почалася в 1994 році, коли Расмус Лерддорф написав декілька програм інтерфейсу Common Gateway Interface (CGI) в C, які він використовував для підтримки своєї особистої домашньої сторінки. Він поширював їх на роботу з веб-формами та спілкуватися з базами даних і назвав цю реалізацію "Персональна домашня сторінка / Форми перекладача" або PHP / FI.

Основою PHP є гілка PHP, яка спочатку була озвучена як PHP наступного покоління (phpng). Його створили Дмитро Стогов, Xinchun Hu та Nikita Popov, і вони мали на меті оптимізувати продуктивність PHP шляхом рефакторингу Zend Engine, зберігаючи при цьому майже повну сумісність мов. Станом на 14 липня 2014 р. Орієнтовні показники на базі WordPress, які служили основним набором стандартних тестів для проекту phpng, демонстрували майже 100% збільшення ефективності. Очікується також, що зміни у phpng допоможуть спростити підвищення продуктивності в майбутньому, оскільки більш компактні структури даних та інші зміни вважаються кращими для успішної переміщення до компілятора, що працює у режимі реального часу (JIT). Через значні зміни, перероблений Zend Engine називається Zend Engine 3, після чого використовується Zend Engine 2, що використовується в PHP 5.

Через значних внутрішніх змін у phpng він повинен отримати новий номер версії для більшості PHP, а не другорядний випуск PHP 5, відповідно до процесу випуску PHP. Найважливіші версії PHP дозволяють відірвати код від зворотної сумісності, отже, PHP 7 дає можливість для інших покращень за межами phpng, які потребують перерв на відстані сумісності. Зокрема, це включало такі зміни:

Багато фатальних механізмів помилок PHP, що спричинили смертельну або відновлюваний рівень, були замінені сучасними об'єктно-орієнтованими винятками.

Синтаксис змінного перейменування перероблений, щоб бути внутрішньо більш послідовним та повним, дозволяючи використовувати оператори `->`, `[]`, `()`, `{}` і `::` з довільними змістовними виразами з лівими сторонами.

Підтримка застарілих методів конструктора PHP 4 була застарілою.

Поведінка попереднього твердження була змінена, щоб бути більш передбачуваною.

Конструктори для декількох класів, вбудованих в PHP, які повернулися на нуль після відмови, були змінені, щоб виключити замість узгодженості.

Кілька незахищених або застарілих серверних прикладних програмних інтерфейсів (SAPI) та розширень були вилучені з ядра PHP, особливо з використанням застарілого розширення mysql.

Поведінка оператора `list ()` було змінено, щоб видалити підтримку рядків.

Підтримка застарілих роздільників PHP-коду PHP (`<% i%>`, `<script language = php> i </ script>`) була вилучена.

Проводиться нагляд, що дозволяє оператору перемикач мати кілька посилань за промовчанням.

Підтримка підтримки шістнадцяткового числа в деяких неявних переходах від рядків до типів номерів була вилучена.

Оператори з лівим та правом зміщенням змінювалися, щоб вести себе послідовніше по всій платформі.

Конверсії між цілими числами та числами з плаваючою точкою були жорсткішими та впроваджені послідовніше на різних платформах.

PHP 7 також включав нові мовні функції. Зокрема, він вводить декларації типу повернення для функцій, які доповнюють існуючі декларації типу параметрів, а також підтримку скалярних типів (цілих, поплашкових, строкових та логічних) у параметрах та декларації типу повернення[4].

2.3.1 Опис ліцензування PHP

PHP є вільним програмним забезпеченням, випущеним під ліцензією PHP, яка передбачає, що, продукти, отримані з цього програмного забезпечення, не можуть називатися "PHP", а також "PHP" не можуть з'являтися на їх ім'я без попереднього письмового дозволу від `group@php.net`. Ви можете вказати, що ваше програмне забезпечення працює разом із PHP, сказавши "Foo for PHP" замість того, щоб називати його "PHP Foo" або "phpfoo".

Це обмеження на використання "PHP" робить PHP-ліцензію несумісною з загальною публічною ліцензією (GPL), тоді як ліцензія Zend несумісна через рекламу, подібну до оригінальної ліцензії BSD.

2.3.2 Опис використання PHP

PHP є загальноприйнятою мовою скриптів, яка особливо підходить для веб-розробки на стороні сервера, і в цьому випадку PHP зазвичай працює на

веб-сервері. Будь-який PHP-код у запитуваному файлі виконується за час виконання PHP, зазвичай для створення динамічного вмісту веб-сторінок або динамічних зображень, що використовуються на веб-сайтах або в інших місцях. Він також може використовуватися для скриптів командного рядка та програм для графічного інтерфейсу користувача (GUI) на стороні клієнта. PHP може бути розгорнуто на більшості веб-серверів, на багатьох операційних системах і платформах, і може використовуватися з багатьма системами управління реляційними базами даних (RDBMS). Більшість постачальників веб-хостингу підтримують PHP для використання їхніми клієнтами. Він доступний безкоштовно, і PHP Group надає повний вихідний код для користувачів для побудови, налаштування та розширення для власного використання.

PHP діє переважно як фільтр, беручи внесок з файлу чи потоку, що містить текстові та / або PHP-інструкції, і виводить інший потік даних. Найчастіше виведення буде HTML, хоча це може бути JSON, XML або подвійні дані, такі як зображення чи аудіо формати. Починаючи з PHP 4, синтаксичний аналізатор PHP збирає вхід для створення байт-коду для обробки Zend Engine, що забезпечує покращену продуктивність по відношенню до попередника-інтерпретатора.

Спочатку розроблена для створення динамічних веб-сторінок, тепер PHP зосереджується головним чином на скриптах на сервері, і це схоже на інші мови сценаріїв на стороні сервера, які забезпечують клієнту динамічний вміст веб-сервера, наприклад, ASP.NET Microsoft, "JavaServer Pages", "Sun Microsystems", і mod_perl. PHP також залучила розробку багатьох програмних рамок, які надають будівельні блоки та структуру дизайну для сприяння швидкому розробці додатків (RAD). Деякі з них включають PRADO, CakePHP, Symfony, CodeIgniter, Laravel, Yii Framework, Phalcon і Zend Framework, пропонує функції, подібні до інших веб-структур.

Станом на січень 2013 р. PHP використовувався на більш ніж 240 мільйонів веб-сайтах (39% з вибірки) та був встановлений на 2,1 мільйона веб-серверів.

Інтерфейс командного рядка, php-cli і два драйвери сценаріїв ActiveX для сценаріїв Windows для PHP були створені.

2.3.3 Опис безпеки PHP

У 2017 році 3% всіх вразливостей, перерахованих в Національну базу

вразливостей, були пов'язані з PHP. Історично приблизно 30% всіх вразливостей, перерахованих з 1996 р. В цій базі даних, пов'язані з PHP. Технічні недоліки безпеки самої мови або її основних бібліотек не є частими (22 у 2009 році, близько 1% від загального, хоча PHP застосовується приблизно до 20% перелічених програм). Визнаючи, що програмісти роблять помилки, деякі мови включають в себе відстеження перевірки, щоб автоматично виявляти відсутність перевірки вхідних даних, що викликає багато проблем. Така особливість розробляється для PHP, але її включення до випуску раніше було відхилено кілька разів.

Існують розширені захисні патчі, такі як Suhosin та Hardening-Patch, спеціально розроблені для середовищ веб-хостингу.

Історично старі версії PHP мали деякі параметри конфігурації та значення за замовчуванням для таких параметрів виконання, які зробили деякі PHP-програми схильними до проблем безпеки. Серед них найбільш відомі магістральні інструкції `magic_quotes_gpc` та `register_globals`; останній зробив будь-які параметри URL-адреси статті змінними PHP, відкривши шлях для серйозної вразливості системи, дозволяючи зловмиснику встановити значення будь-якої неініціалізованої глобальної змінної та перешкоджати виконанню PHP-скрипту. Підтримка налаштувань "магічні котирування" та "zareєструвати глобали" застаріла від PHP 5.3.0 і вилючена з PHP 5.4.0.

Іншим прикладом можливої вразливості параметрів виконання часу є помилка у вимкненні виконання PHP (наприклад, через директиву конфігурації двигуна) для каталогу, де зберігаються завантажені файли; це може призвести до виконання шкідливого PHP-коду, вбудованого в завантажені файли, наприклад, коли користувачі можуть завантажувати зображення. Найкраща практика полягає в тому, щоб або знайти каталог зображень за межами кореневого каталогу, доступний для веб-сервера, і обслуговувати його через посередницький скрипт, або відключити виконання PHP для каталогу, в якому зберігаються завантажені файли.

Крім того, включення динамічного завантаження розширень PHP (за допомогою директиви конфігурації `enable_dl`) в середовищі спільного веб-хостингу може призвести до проблем безпеки.

Приховані типи перетворень, які призводять до того, що різні цінності розглядаються як рівні, іноді проти намірів програміста, можуть призвести до проблем безпеки. Наприклад, результат порівняння `0e1234 == 0` є вірним, оскільки перше порівняне значення розглядається як наукове позначення, що має

значення (0×101234), тобто нуль. Помилки, подібні до цього, призвели до вразливостей для аутентифікації на Simple Machines Forum, Туро3 та phpBB, коли порівнювали хеш паролів MD5. Рекомендований спосіб полягає в тому, щоб використовувати `hash_equals()` (для захисту часових нападів), `strcmp` або ідентифікаційний оператор (`===`), оскільки `0e1234 === 0` призводить до помилок.

У результаті аналізу понад 170 000 дефектів веб-сайтів, опублікованих Zone-H у 2013 році, найбільш часто використовувана методика (53%) полягала в експлуатації уразливості до файлу включення, здебільшого пов'язаного з небезпечним використанням функцій PHP, включати, вимагати та `allow_url_fopen[5]`.

2.4 Опис системи управління контентом

Система керування вмістом (CMS) керує створенням та модифікацією цифрового вмісту. Він зазвичай підтримує декілька користувачів у спільному середовищі.

Функції CMS дуже різняться. Більшість CMS включають в себе веб-публікації, управління форматами, редагування історії та управління версіями, індексацію, пошук та пошук. За своєю природою системи управління контентом підтримують відокремлення вмісту та презентації.

Система керування веб-контентом (WCM або WCMS) – це CMS, призначена для підтримки управління вмістом веб-сторінок. Найпопулярніші CMS також є WCMS. Веб-вміст містить текстові та вбудовані графіки, фотографії, відео, аудіо, карти та програмний код (наприклад, для програм), який відображає вміст або взаємодіє з користувачем.

Така система управління контентом (CMS) зазвичай має дві основні компоненти:

Програма керування вмістом (CMA) – це інтерфейсний інтерфейс користувача, який дозволяє користувачеві навіть із обмеженим досвідом додавати, змінювати та видаляти вміст із веб-сайту без втручання веб-майстра.

Програма доставки контенту (CDA) збирає цю інформацію та оновлює веб-сайт.

Цифрові системи управління активами – це ще один тип CMS. Вони керують вмістом з чітко визначеним автором чи правом власності, такими як документи, фільми, зображення, номери телефонів та наукові дані. Компанії також використовують CMS для зберігання, контролю, перегляду та публікації документації.

На основі статистики частки ринку найпопулярнішою системою управління контентом є WordPress, яка використовується більш ніж 28% усіх веб-сайтів в Інтернеті та 59% всіх веб-сайтів, що використовують відому систему керування контентом, а потім Joomla та Drupal.

Загальні особливості

Системи управління контентом зазвичай надають такі функції:

- 1) SEO-дружні URL-адреси;
- 2) інтегрована і онлайн-допомога;
- 3) модульність та розширюваність;
- 4) користувацька та групова функціональність;
- 5) підтримка шаблонів для зміни дизайну;
- 6) встановлення та оновлення майстрів;
- 7) інтегровані журнали аудиту;
- 8) дотримання різних рамок та стандартів доступності, таких як WAI-ARIA.

Переваги

- скорочена потреба в коді з нуля;
- легко створювати єдиний зовнішній вигляд;
- контроль версії;
- редагувати керування правами.

Недоліки

- обмежена або відсутність можливості створювати функціональні можливості, не передбачені в CMS (наприклад, макети, веб-програми тощо);
- зростає потреба в спеціальних знаннях та тренінгах для авторів контенту.

2.4.1 CMS WordPress

WordPress (WordPress.org) – це безкоштовна та відкрита система керування контентом (CMS) на базі PHP та MySQL. Особливості включають архітектуру плагінів і систему шаблонів. Це найбільше пов'язано з веденням блогу, але підтримує інші типи веб-вмісту, включаючи більше традиційних списків розсилки та форумів, медіа-галерей та інтернет-магазинів. WordPress – це найпопулярніша система керування сайтом, що використовується. WordPress також використовувався для інших доменів програм, таких як просторові системи відображення (PDS).

WordPress був випущений 27 травня 2003 року його засновниками, Matt Mullenweg та Майк Літт, як вилка b2 / safelog. Програмне забезпечення випускається відповідно до ліцензії GPLv2 (або пізнішої).

Щоб працювати, WordPress має бути встановлений на веб-сервері, як частина служби хостингу в Інтернеті, наприклад, WordPress.com, або на комп'ютері з пакетом програм WordPress.org, щоб як власне користуватися мережевим хостом. Локальний комп'ютер може використовуватися для однокористувацьких цілей тестування та навчання.

WordPress має систему веб-шаблонів, використовуючи шаблонний процесор. Його архітектура є фронтальним контролером, що спрямовує всі запити для нестатичних URI на один файл PHP, який аналізує URI та визначає цільову сторінку. Це дозволяє підтримувати більш доступні для людей читання.

2.4.1.1 Теми у CMS WordPress

Користувачі WordPress можуть встановлювати та переключатися між різними темами. Теми дозволяють користувачам змінювати зовнішній вигляд і функціональність веб-сайту WordPress, не змінюючи основний код або вміст сайту. Кожен веб-сайт WordPress вимагає щонайменше однієї теми, і будь-яка тема повинна бути розроблена з використанням стандартів WordPress з структурованим PHP, дійсним HTML (HyperText Markup Language) та каскадними таблицями стилів (CSS). Теми можуть бути безпосередньо встановлені, використовуючи інструмент адміністрування WordPress "Appearance" на інформаційній панелі, або тематичні папки можуть бути скопійовані безпосередньо в тематичну директорію, наприклад через FTP. PHP, HTML та CSS, знайдені в темах, можуть бути безпосередньо модифіковані, щоб змінити поведінку теми, або тема може бути "дитиною", яка успадковує параметри з іншої теми та вибірково перевизначає функції. Теми WordPress, як правило, поділяються на дві категорії: безкоштовні та преміальні. Багато безкоштовних тем перелічено в каталозі теми WordPress, а теми преміум доступні для покупки з ринків і окремих розробників WordPress. Користувачі WordPress також можуть створювати та створювати власні спеціальні теми. Безкоштовна тема Підкреслення, створені розробниками WordPress, стала популярною основою для нових тем.

2.4.1.2 Плагіни у CMS WordPress

Архітектура плагінів WordPress дозволяє користувачам розширювати

можливості та функціональність веб-сайту чи блогу. Станом на березень 2017 р. У WordPress доступні понад 55 286 плагінів, кожен з яких пропонує власні функції та функції, які дозволяють користувачам адаптувати свої сайти до їхніх конкретних потреб. Ці налаштування варіюються від оптимізації пошукової системи, до порталів клієнтів, які використовуються для відображення особистої інформації для входу користувачів, до систем управління контентом, до функцій відображення вмісту, таких як додавання віджетів та навігаційних смуг. Не всі доступні плагіни завжди в курсі оновлень, і, як наслідок, вони можуть не працювати належним чином або взагалі не працювати. Більшість плагінів доступні через WordPress за допомогою завантаження та встановлення файлів вручну через FTP або через інформаційну панель WordPress. Тим не менше, багато третіх сторін пропонують плагіни за допомогою власних веб-сайтів, багато з яких є платними пакетами.

Веб-розробники, які хочуть розробляти плагіни, повинні вивчати систему ключів WordPress, яка складається з більш ніж 300 хуків, поділених на дві категорії: `action-hook` та `filter-hook`.

2.4.1.3 Особливості WordPress

WordPress також має інтегроване керування посиланнями; зручна для пошуку система, чиста постійна структура; можливість призначати декілька категорій для публікацій; і підтримка позначення повідомлень. Автоматичні фільтри також включені, забезпечуючи стандартизоване форматування та стиль тексту в публікаціях (наприклад, перетворення звичайних цитат у інтелектуальні цитати). WordPress також підтримує стандарти Trackback та Pingback для відображення посилань на інші сайти, які самі пов'язані з публікацією чи статтею. Дописи WordPress можна редагувати у форматі HTML, використовуючи візуальний редактор або використовувати один із декількох плагінів, які дозволяють використовувати різні налаштування редагування.

До версії 3 WordPress підтримує один блог за встановлення, хоча декілька паралельних копій можуть запускатися з різних каталогів, якщо вони налаштовані на використання окремих таблиць бази даних. WordPress Multisites (раніше згаданий як WordPress Multi-користувач, WordPress MU або WPMU) був форпором WordPress створеного, щоб дозволити декілька блогів існувати в рамках однієї інсталяції, але може керуватися централізованим супровідником. WordPress MU дає змогу тим, хто має веб-сайти, розміщувати власні спі-

льності для блогів, а також керувати та моделювати всі блоги з однієї інформаційної панелі. WordPress MS додає вісім нових таблиць даних для кожного блогу. З моменту виходу WordPress 3, WordPress MU об'єднався з WordPress. b2 / cafelog, більш відомий як b2 або cafelog, був попередником WordPress. За оцінками, b2 / cafelog був встановлений приблизно з 2000 блогів на травень 2003 року. Він був написаний на PHP для використання з MySQL Мішелем Вальдрігі, який зараз є розробником WordPress. Хоча WordPress є офіційним наступником, ще один проект b2evolution також активно розвивається.

WordPress вперше з'явився в 2003 році як спільне зусилля між Меттом Малленвегом та Майком Літтсом для створення вилки b2. Крістін Селк Тремулет, друг Малленвега, запропонував назву WordPress.

У 2004 році умови ліцензування конкуруючого пакету Movable Type були змінені Six Apart, в результаті чого багато хто з найбільш впливових користувачів мігрують на WordPress. До жовтня 2009 р. Звіт з MarketShare CMS з відкритим вихідним кодом дійшов висновку, що WordPress користується найбільшою популярністю у будь-якій системі керування контентом з відкритим кодом.

Станом на лютий 2017 року WordPress використовується 58,7% усіх веб-сайтів, на яких відомо, що система управління контентом є відомим. Це на 27,5% від 10 мільйонів найбільших веб-сайтів.

Переможець Infoworld "Найкраща нагорода з програмного забезпечення з відкритим вихідним кодом: Співпраця", нагороджена в 2008 році.

Переможець відкритого коду CMS Awards "Загальна найкраща CMS з відкритим вихідним кодом", яка була присуджена в 2009 році.

Переможець категорії "Слайд-зал слави CMS у категорії" Відкрите джерело 2010 ", присвоєний в 2010 році.

Переможець Infoworld "Bossie award" за краще програмне забезпечення з відкритим вихідним кодом ", нагороджений в 2011 році.

Переможець премії CMS Critic Award "Краща CMS для персональних веб-сайтів", нагороджений у 2015 році.

WordPress має п'ятизірковий рейтинг приватності від Electronic Frontier Foundation.

2.4.1.4 Вразливості у CMS WordPress

У програмному забезпеченні було виявлено багато проблем безпеки, зокрема в 2007, 2008 та 2015 роках. За даними Secunia, WordPress у квітні 2009

року мав сім не виправлених повідомлень із безпеки (з 32 загальних), з максимальним рейтингом «менш критично». Secunia підтримує оновлений список вразливостей WordPress.

У січні 2007 року багато популярних блогів з оптимізації пошукових систем (SEO), а також багатьох низькопрофільних комерційних блогів із рекламою AdSense були націлені на атаку з використанням WordPress. Окремі уразливості на одному з веб-серверів веб-сайту проекту дозволили злочинцю представити коди, що використовується, у формі заднього боку для деяких завантажень WordPress 2.1.1. Випуск 2.1.2 розглянув цю проблему; наразі випущений порада порадив усім користувачам оновити систему негайно.

У травні 2007 року дослідження показало, що 98% блогів WordPress, що працюють, були доступними для використання, оскільки вони використовували застарілі та не підтримувані версії програмного забезпечення. Частково, щоб пом'якшити цю проблему, WordPress зробив оновлення програмного забезпечення набагато простішим автоматизованим процесом "одним клацанням" у версії 2.7 (випущений в грудні 2008 року). Однак параметри безпеки файлової системи, необхідні для ввімкнення процесу оновлення, можуть стати додатковим ризиком.

У інтерв'ю у червні 2007 року, Stefan Esser, засновник команди реагування на безпеку PHP, критично оцінив історію безпеки WordPress, посилаючись на проблеми з архітектурою програми, що зробило за необхідне складним створення коду, захищеного від вразливостей SQL injection, оскільки як і деякі інші проблеми.

У червні 2013 року було встановлено, що деякі з 50 найбільш завантажених плагінів WordPress були уразливими до загальних веб-атак, таких як SQL injection та XSS. Окремий огляд 10-ти кращих плагінів електронної торгівлі показав, що сім з них були вразливими.

З метою підвищення безпеки та оптимізації загального досвіду оновлення, автоматичні фонові оновлення були представлені в WordPress 3.7.

Індивідуальні інсталяції WordPress можуть бути захищені плагінами безпеки, які перешкоджають переліку користувачів, приховують ресурси та блокують зонди. Користувачі також можуть захистити свої установки WordPress шляхом виконання таких дій, як підтримання всієї інсталяції WordPress, теми та плагінів, оновлення яких використовується лише з надійними темами та плагінами, редагування файлу .htaccess сайту для запобігання багатьом типів атак на SQL injection та блокування неавторизованих доступ до конфіденційних файлів. Особливо важливо оновлювати плагіни WordPress, оскільки майбутні

хакери можуть легко перелічити всі плагіни, які використовує сайт, а потім запускати сканування, шукаючи будь-які вразливі місця щодо цих плагінів. Якщо уразливості виявлені, їх можна використовувати для того, щоб дозволити хакерам завантажувати власні файли (наприклад, скрипт PHP Shell), які збирають конфіденційну інформацію.

Розробники також можуть використовувати інструменти для аналізу потенційних уразливостей, зокрема WPScan, WordPress Auditor та WordPress Sploit Framework, розробленої за допомогою 0pc0deFR. Ці типи інструментів досліджують відомі вразливості, такі як CSRF, LFI, RFI, XSS, SQL injection та користувацький перелік. Проте не всі вразливості можуть бути виявлені інструментами, тому доцільно перевірити код плагінів, тем та інших надбудов від інших розробників.

У березні 2015 року, як повідомили багато експертів із питань безпеки та оптимізаторів пошукових систем, включаючи пошукову систему, що плагін SEO для WordPress під назвою Yoast, який використовується більш ніж 14 мільйонами користувачів у всьому світі, має вразливість, що може призвести до вибуху, де хакери можуть робити Blind SQL injection. Щоб вирішити цю проблему, вони негайно представили нову версію 1.7.4 того ж плагіна, щоб уникнути будь-яких перешкод у мережі через відсутність безпеки, що існував у плагіні.

У січні 2017 р. Аудитори безпеки в Sucuri визначили вразливість в API REST API WordPress, що дозволить будь-якому неаутентичному користувачеві змінювати будь-які публікації або сторінки на сайті, на якому працює WordPress 4.7 або вище. Аудитори спокійно повідомляли розробникам WordPress, і через шість днів WordPress випустив патч з високим пріоритетом до версії 4.7.2, який вирішив проблему.

Застереження про відбитки пальців на полотні, яке зазвичай надається веб-переглядачем Tor для веб-сайтів на базі WordPress.

Мінімальна вимога PHP до версії WordPress – це PHP 5.2, який був випущений 6 січня 2006 р., який не підтримується групою PHP, і не отримав жодних патчів безпеки з 6 січня 2011 року.

За відсутності конкретних змін у коді форматування за замовчуванням, на веб-сайтах на базі WordPress використовується елемент canvas, щоб визначити, чи може браузер правильно відтворити смайли. Оскільки Tor Browser не виявляє дискримінації між цим законним використанням API Canvas та прагненням до відбитка пальців полотна, він попереджає, що веб-сайт намагається "витягнути зображення HTML5 зображення". Постійні зусилля намагаються

знайти шляхи заспокоїти захисників конфіденційності, зберігаючи при цьому можливість перевіряти належні можливості рендеринга Емої.

2.4.1.5 Плагін WooCommerce

WooCommerce – це плагін для електронної комерції з відкритим кодом для WordPress. Він призначений для інтернет-продавців малого та великого розміру за допомогою WordPress. Запущений 27 вересня 2011 року, плагін швидко став популярним завдяки своїй простоті встановлення та налаштування і безкоштовного базового продукту.

WooCommerce вперше був розроблений розробником WordPress темою WooThemes, який найняв Майкла Джоллі та Джеймса Костера, розробників у Jigowatt, для роботи на вил Jigoshop, який став WooCommerce. У серпні 2014 р. WooCommerce працював на 381 187 сайтах (або 17,77% електронних комерційних сайтів в Інтернеті).

У листопаді 2014 року в Сан-Франциско, штат Каліфорнія, відбулася перша конференція WooConf, присвячена електронній комерції за допомогою WooCommerce. Він залучив 300 відвідувачів.

У травні 2015 року WoThemes та WooCommerce були придбані компанією Automattic, оператором WordPress.com та основним співробітником програмного забезпечення WordPress.

WooCommerce використовується низкою сайтів з високим рівнем трафіку, таких як Small Press Expo. На третій тиждень вересня 2015 року "Тенденції" зазначили, що WooCommerce використовує 30% сайтів електронної комерції та мільйони активних встановлень. Електронна комерція швидко зростає у всьому світі, і WooCommerce має більше 39 мільйонів завантажень як плагін, і наразі активно працює на більш ніж 3 мільйонам веб-сайтів і є найпопулярнішою платформою електронної комерції в 2018 році. WooCommerce має приблизно 4% з мільйонів HTML сторінок. У 2018 році статистика показує, що відсоток Інтернет-магазинів, які використовують WooCommerce через плагін Wordpress.org, складає понад 30% всіх магазинів. WooCommerce має повний контроль над часткою ринку з видатним 42% усіх інтернет-магазинів, що працюють від Woocommerce.

Оскільки придбання Automattic WooCommerce продовжує збільшувати свою частку ринку, вона стала однією з провідних електронних комерційних платформ в Інтернеті.

З багатьма темами Woocommerce, що продаються на сторонніх веб-сайтах, важко точно визначити, скільки тем можна пов'язати з цим плагіном Wordpress, але ось деякі статистичні дані Woocommerce для постачальників великих тем.

WooCommerce залучила значну популярність, оскільки основний продукт, крім багатьох розширень і плагінів, є вільним та відкритим вихідним кодом. У 2018 році WooCommerce має близько 330 розширень і понад 1000 плагінів. Крім того, існують тисячі платних надбудов за фіксованими цінами. Багато преміум-теми тепер пропонують можливості з WooCommerce, а також плагіни, які роблять тематичні рамки сумісними[6].

2.5 Шаблон проектування MVC

Model-View-Controller (MVC, «Модель-Представлення-Контролер», «Модель-Вид-Контролер») – схема поділу даних програми, призначеного для користувача інтерфейсу і керуючої логіки на три окремих компоненти: модель, уявлення і контролер – таким чином, що модифікація кожного компонента може здійснюватися незалежно .

Модель (Model) надає дані і реагує на команди контролера, змінюючи свій стан.

Подання (View) відповідає за відображення даних моделі користувачеві, реагуючи на зміни моделі .

Контролер (Controller) інтерпретує дії користувача, сповіщаючи модель про необхідність змін .

2.5.1 Історія розвитку MVC

Концепція MVC була описана Трюгве Реенскаугом в 1978 році, який працював в науково-дослідному центрі «Xerox PARC» над мовою програмування «Smalltalk». Пізніше, Стів Бурбек реалізував шаблон в Smalltalk-80 .

Остаточна версія концепції MVC була опублікована лише в 1988 році в журналі Technology Object.

Згодом шаблон проектування став еволюціонувати. Наприклад, була представлена ієрархічна версія HMVC; MVA, MVVM.

Подальший виток популярності привнесло розвиток фреймворків, орієнтованих на швидку розгортку, на мовах Python і Ruby, Django і Rails. На момент 2017 року, фреймворки з MVC зайняли помітні позиції по відношенню до

решти фреймворками без цього шаблону .

2.5.2 Відмінності опису концепції шаблону

З розвитком об'єктно-орієнтованого програмування і поняття про шаблони проектування – був створений ряд модифікацій концепції MVC, які при реалізації у різних авторів можуть відрізнятися від оригінальної. Так, наприклад, Еріан Верми в 2004 році описав приклад узагальненого MVC.

У передмові до дисертації «Naked objects» Річарда Поусона (Richard Pawson), – Трюгве Реенскауг згадує свою неопубліковану найбільш ранню версію MVC, згідно з якою :

- модель ставилася до «розуму» користувача;
- під поданням мався на увазі редактор, що дозволяє користувачеві переглядати і оновлювати інформацію;
- контролер був інструментом для зв'язування уявлень воєдино і застосовувався користувачем для вирішення його завдань.

2.5.3 Призначення MVC

Основна мета застосування цієї концепції полягає в відділенні бізнес-логіки (моделі) від її візуалізації (уявлення, виду). За рахунок такого поділу підвищується можливість повторного використання коду. Найбільш корисне застосування даної концепції в тих випадках, коли користувач повинен бачити ті ж самі дані одночасно в різних контекстах і / або з різних точок зору. Зокрема, виконуються наступні завдання.

До однієї моделі можна приєднати кілька видів, при цьому не зачіпаючи реалізацію моделі. Наприклад, деякі дані можуть бути одночасно представлені у вигляді електронної таблиці, гістограми і кругової діаграми.

Не торкаючись реалізацію видів, можна змінити реакції на дії користувача (натискання мишею на кнопки, введення даних) – для цього досить використовувати інший контролер.

Ряд розробників спеціалізується тільки в одній з областей: або розробляють графічний інтерфейс, або розробляють бізнес-логіку. Тому можливо добитися того, що програмісти, які займаються розробкою бізнес-логіки (моделі), взагалі не будуть обізнані про те, яке уявлення буде використовуватися.

2.5.4 Концепція MVC

Концепція MVC дозволяє розділити модель, уявлення і контролер на три окремих компоненти.

Модель. Модель надає дані і методи роботи з ними: запити до бази даних, перевірка на коректність. Модель не залежить від уявлення (не знає як дані візуалізувати) і контролера (не має точок взаємодії з користувачем) просто надаючи доступ до даних і управління ними.

Модель будується таким чином, щоб відповідати на запити, змінюючи свій стан, при цьому може бути вбудовано повідомлення «спостерігачів».

Модель, за рахунок незалежності від візуального представлення, може мати кілька різних уявлень для однієї «моделі».

Представлення. Представлення відповідає за отримання необхідних даних з моделі і відправляє їх користувачеві. Представлення не обробляє жодних введених даних користувача.

Контролер. Контролер забезпечує «зв'язок» між користувачем і системою. Контролер і направляє дані від користувача до системи і навпаки. Використовує модель і уявлення для реалізації необхідного дії.

Функціональні можливості і розбіжності. Оскільки MVC не має суворої реалізації, то реалізований він може бути по-різному. Немає загальноприйнятого визначення, де повинна розташовуватися бізнес-логіка. Вона може знаходитися як в контролері, так і в моделі. В останньому випадку, модель буде містити всі бізнес-об'єкти з усіма даними і функціями.

Деякі фреймворки жорстко задають де повинна розташовуватися бізнес-логіка, інші не мають таких правил.

Також не вказано, де повинна знаходитися перевірка введених користувачем даних. Проста валідація може зустрічатися навіть у поданні, але частіше вони зустрічаються в контролері чи моделлю.

Ітернаціоналізація і форматування даних також не має чітких вказівок по розташуванню.

2.5.5 Умовно-обов'язкові модифікації

Для реалізації схеми «Model-View-Controller» використовується досить велика кількість шаблонів проектування (в залежності від складності архітектурного рішення), основні з яких – «спостерігач», «стратегія», «компоновщик».

Найбільш типова реалізація – в якій уявлення відокремлено від моделі шляхом встановлення між ними протоколу взаємодії, що використовує «апарат подій» (позначення «подіями» певних ситуацій, що виникають в ході виконання програми, – і розсилка повідомлень про них всім тим, хто підписався на отримання) : при кожному особливому зміні внутрішніх даних в моделі (позначеному як «подія»), вона сповіщає про нього ті залежать від неї уявлення, які передплатили такого оповіщення – і уявлення оновлює я. Так використовується шаблон «спостерігач».

При обробці реакції користувача – уявлення вибирає, в залежності від реакції, потрібний контролер, який забезпечить ту чи іншу зв'язок з моделлю. Для цього використовується шаблон «стратегія», або замість цього може бути модифікація з використанням шаблону «команда».

Для можливості однотипного поводження з подоб'єкти складно-складеного ієрархічного виду – може використовуватися шаблон «компоновщик». Крім того, можуть використовуватися і інші шаблони проектування – наприклад, «фабричний метод», який дозволить задати за замовчуванням тип контролера для відповідного виду[7].

2.6 UML діаграма WordPress

UML (Unified Modeling Language) – уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація[8].

Діаграма, представлена на рис. 2.1, зображує структуру бази даних WordPress.

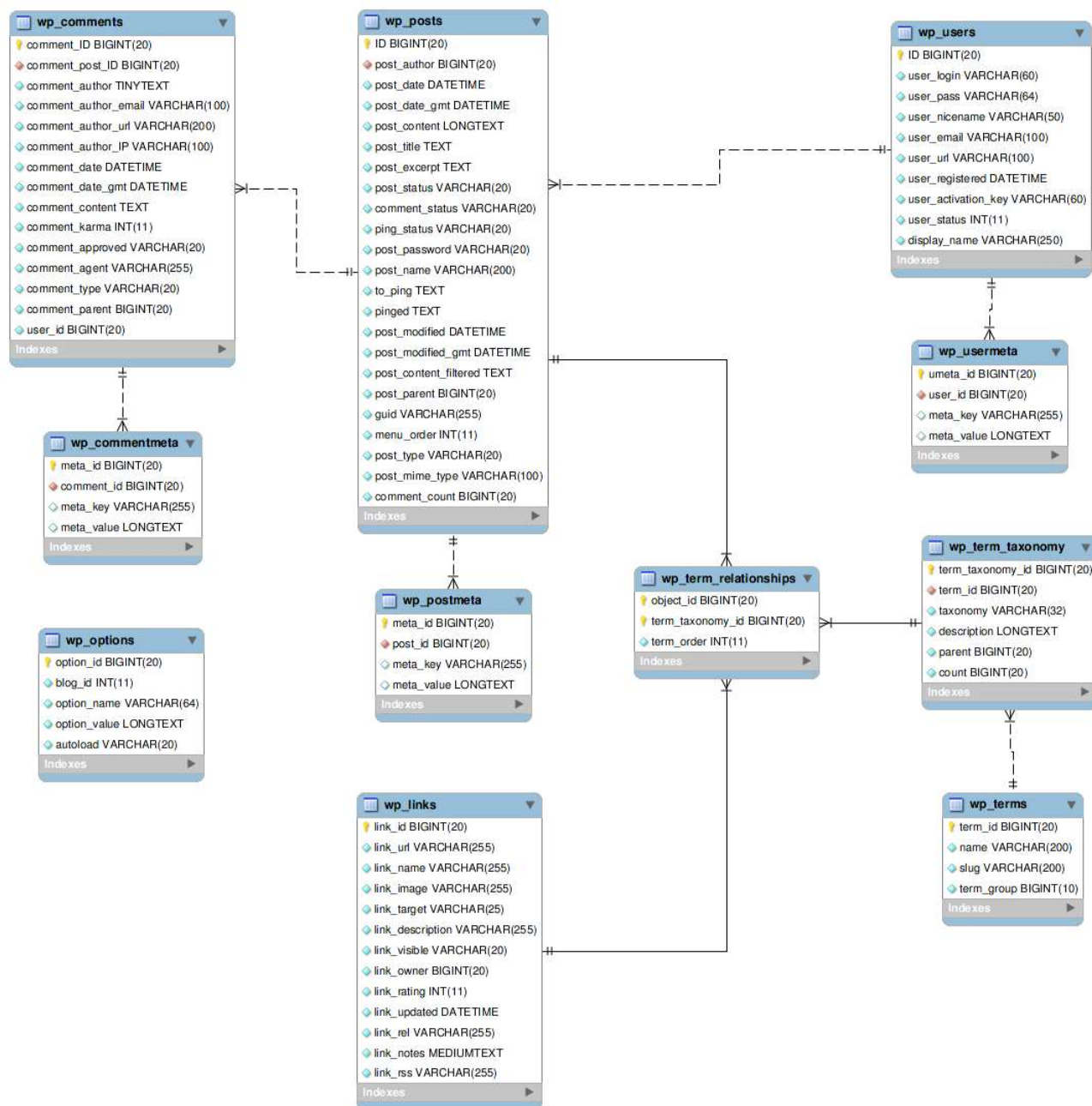


Рисунок 2.1 – Діаграма бази даних WordPress

3 ОПИС ПРОЕКТНОГО І ТЕХНІЧНОГО РІШЕННЯ

Розробка веб-додатків – це процес та практика розробки веб-програм. Існує консенсус щодо того, що процеси, що беруть участь, є розширенням стандартних процесів розробки програмного забезпечення. З огляду на це, поряд з його унікальними характеристиками, популярні рамки включають в себе спіральний підхід і бізнес-орієнтований підхід до розробки додатків, серед інших моделей, які відповідають вимогам для ітеративного процесу.

Як і в традиційному настільному застосуванні, веб-додатки мають різний рівень ризику. Особистий домашній сайт набагато менш ризикований, ніж, наприклад, веб-сайт з фондової торгівлі. Для деяких проектів безпеки, помилки програмного забезпечення тощо є основними проблемами. Якщо час виходу на ринок або технічна складність викликає занепокоєння, документація, планування тестування, контроль над змінами, аналіз вимог, архітектурного опису та формальної розробки та практики будівництва можуть пом'якшити ризик.

3.1 Мікрофреймворк WP Emerge

WP Emerge – це мікрофреймворк, який модернізує WordPress як CMS, надаючи інструменти для реалізації MVC структури та не тільки.

Інтеграція є додатковою тому можливо використовувати її лише на потрібних сторінках або на всіх сторінках.

За своєю структурою WP Emerge подібний до таких популярних фреймворків як Laravel, Slim або Symfony, тому дуже підходить для сучасних розробників.

Мікрофреймворк – це термін, який використовується для позначення мінімалістичних фреймворків які використовуються для веб-додатків. Це контрастує з каркасами повного складу.

Мікрофреймворк не має більшості функціональних можливостей, які можна очікувати в повноцінного фреймворку, наприклад:

- облікові записи, автентифікація, авторизація, ролі тощо;
- абстракція бази даних через об'єктно-реляційне відображення;
- верифікація вхідних даних та вхідні канали;
- двигун веб-шаблону;
- як правило, мікрофреймворк полегшує прийом HTTP-запиту, марш-

рутизацію HTTP-запиту на відповідний контролер, відправлення контролера та повернення HTTP-відповіді. Мікрорамники часто розробляються спеціально для побудови API для іншої служби або програми.

Крім того є можливість використовувати спеціально створену тему WP Emerge Starter Theme яка представляє собою стартову тему для початку розробки.

3.2 Хостінг та сервер

Перед тим як приступити до розробки веб-додатку треба мати сервер та хостінг на якому буде розміщений сайт для безпосередньої розробки.

В моєму випадку було вибрано найпопулярніший хостінг в Україні <https://www.ukraine.com.ua/> (див. рис. 3.1).

The screenshot displays the website for 'hosting UKRAINE', which offers domain registration and hosting services. The top navigation bar includes links for 'Хостинг', 'Бизнес-хостинг', 'Регистрация доменов', 'VPS', 'Выделенные сервера', 'FAQ', 'Форум', and 'Карьера'. A search bar and contact information are also present.

The main section is titled 'Регистрация доменов' (Domain Registration) and lists various domain extensions with their respective prices. Below this, there are several service categories: 'Трансфер домена', 'Как оформить заказ', 'Наши преимущества', 'Способы оплаты', and 'Новости'.

The central part of the page features a comparison table of hosting plans, categorized into four types: 'Сайт SSD', 'Лучший SSD', 'Мастер SSD', and 'Эксперт SSD'. Each plan includes details on disk space, number of sites, PHP memory limit, and server location. Prices are listed in Ukrainian Hryvnia (UAH) per month.

Параметр	Сайт SSD	Лучший SSD	Мастер SSD	Эксперт SSD
Место на SSD диске	1 GB	5 GB	10 GB	10 GB
Бесплатный SSL сертификат	✓	✓	✓	✓
Сайтов	1	5	10	неограниченно
PHP memory_limit	64 MB	128 MB	256 MB	256 MB
Субдоменов	неограниченно	неограниченно	неограниченно	неограниченно
Расположение серверов	Украина, Беларусь, Литва, Латвия, Эстония, Польша, Чехия, Словакия, Венгрия, Румыния, Болгария, Сербия, Черногория, Македония, Словения, Хорватия, Босния и Герцеговина, Албания, Молдова, Грузия, Армения, Азербайджан, Казахстан, Кыргызстан, Узбекистан, Таджикистан, Туркменистан, Узбекистан, Таджикистан, Туркменистан	Украина, Беларусь, Литва, Латвия, Эстония, Польша, Чехия, Словакия, Венгрия, Румыния, Болгария, Сербия, Черногория, Македония, Словения, Хорватия, Босния и Герцеговина, Албания, Молдова, Грузия, Армения, Азербайджан, Казахстан, Кыргызстан, Узбекистан, Таджикистан, Туркменистан, Узбекистан, Таджикистан, Туркменистан	Украина, Беларусь, Литва, Латвия, Эстония, Польша, Чехия, Словакия, Венгрия, Румыния, Болгария, Сербия, Черногория, Македония, Словения, Хорватия, Босния и Герцеговина, Албания, Молдова, Грузия, Армения, Азербайджан, Казахстан, Кыргызстан, Узбекистан, Таджикистан, Туркменистан, Узбекистан, Таджикистан, Туркменистан	Украина, Беларусь, Литва, Латвия, Эстония, Польша, Чехия, Словакия, Венгрия, Румыния, Болгария, Сербия, Черногория, Македония, Словения, Хорватия, Босния и Герцеговина, Албания, Молдова, Грузия, Армения, Азербайджан, Казахстан, Кыргызстан, Узбекистан, Таджикистан, Туркменистан, Узбекистан, Таджикистан, Туркменистан
Цена	от 43.33 грн/мес	от 88.89 грн/мес	от 112.44 грн/мес	от 168.22 грн/мес
Действия	Заказать	Заказать	Заказать	Заказать

Below the table, there is a section for 'PHP' settings, including 'PHP 7.2, 7.1, 7.0, 5.6, 5.5, 5.4, 5.3, 5.2, 5.1' and 'Полноценный режим PHP без safe_mode'. There is also a '30-дневный возврат денег' (30-day money back guarantee) offer.

Рисунок 3.1 – Хостінг Україна

Після чого у «Панелі управління» необхідно придбати та зареєструвати домен а після створити сайт (див. рис. 3.2).

Щоб сайт став доступний в мережі Інтернет, потрібно прив'язати домен до хостингу. Для цього:

- 1) вкажіть для домену хостингові DNS;
- 2) додайте домен в панель управління хостингом.

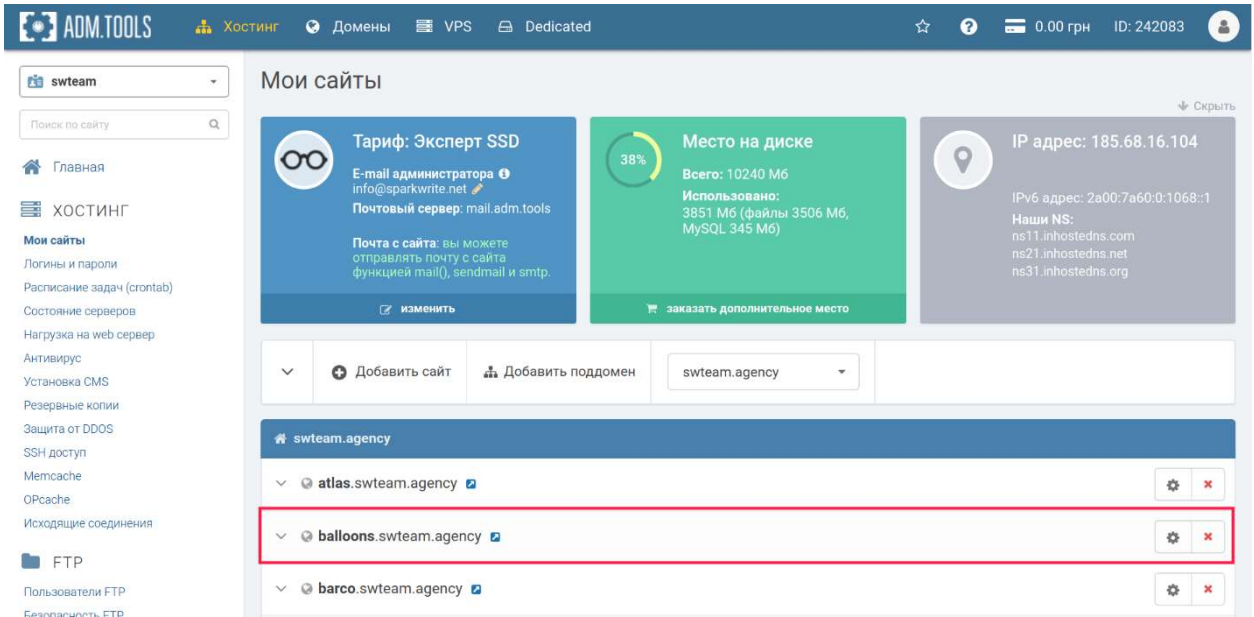


Рисунок 3.2 – Створенний сайт

Після необхідно створити базу даних для майбутнього сайту (див. рис. 3.3).

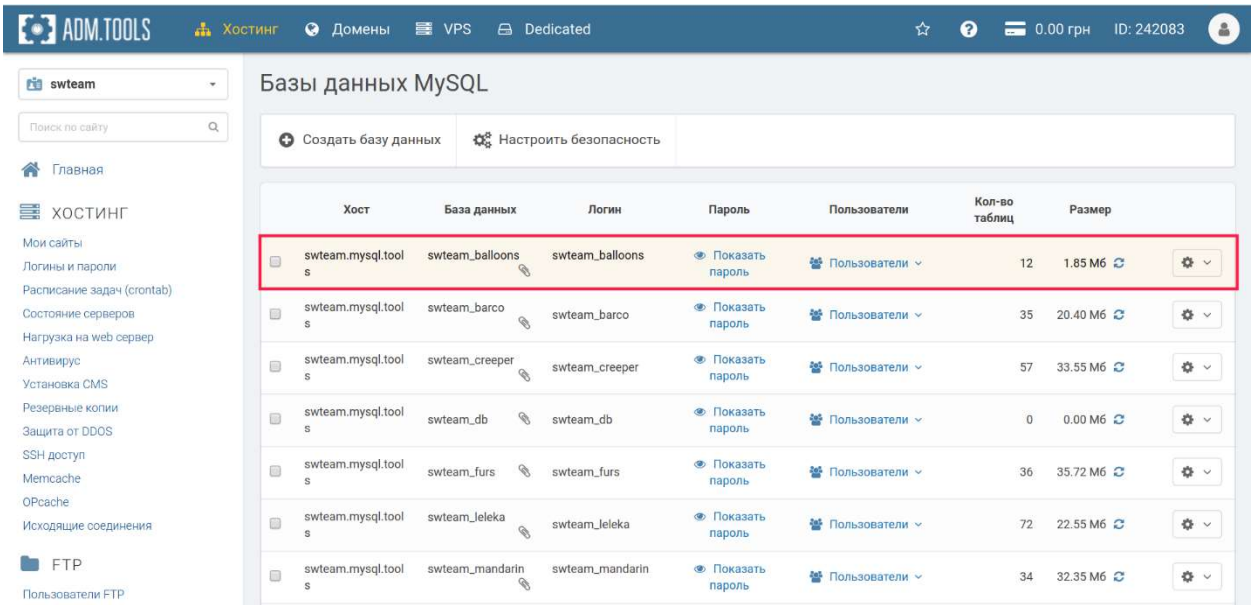


Рисунок 3.3 – Створена база даних

Після того як було успішно створено сайт та базу даних можна приступати до встановлення CMS WordPress.

3.3 Встановлення WordPress та WP Emerge

Хостінг «Україна» надає можливість встановити CMS як вручну так і за допомогою додаткових сервісів. Завдяки додатковим сервісам можливо окрім WordPress також встановити і інші популярні системи керування вмістом такі як concrete5, joomla, Drupal, OpenCart, Coppermine, modx-revolution, moodle, osCommerce, smf, typo3, VamShop, Zen Cart.

Серед них є також ті котрі дозволяють також створювати та керувати сайтами електронної комерції.

В основному, вважаючи провідну платформу електронної комерції з відкритим кодом для WordPress, WooCommerce може похвалитися великою і швидко зростаючою спільнотою користувачів. Його бездоганна інтеграція з WordPress робить його легким встановлення та керування, що є однією з основних причин того, чому роздрібні торговці, такі як Cupcake Studio та EcoKitty, зробили це своєю електронною комерційною платформою з відкритим вихідним кодом.

Мобільна привабливість WooCommerce робить це привабливим вибором для брендів, які прагнуть задовольнити зростаючу кількість клієнтів, які використовують свій планшет або телефон для покупки. Насправді, платформа навіть дозволяє брендам створювати мобільні додатки для своєї вітрини з установкою надбудови.

WooCommerce залучила значну популярність, оскільки базовий продукт, крім багатьох розширень і плагінів, є безкоштовним та відкритим кодом. У 2018 році WooCommerce має близько 330 розширень і понад 1000 плагінів. Крім того, існують тисячі платних надбудов за фіксованими цінами.

Оскільки WooCommerce можна використовувати лише з WordPress, це не може бути найбільш привабливим рішенням для деяких роздрібних торговців та брендів. Однак для тих, хто вже використовує WordPress, масштабованість, яку пропонує WooCommerce, робить її особливо привабливою. Проте, як можна здогадатися, платні розширення необхідні, щоб максимально ефективно використовувати рішення електронної комерції (див. рис. 3.4).

Завдяки безмежній гнучкості та доступ до сотень безкоштовних та преміум-розширень WordPress, WooCommerce тепер набирає 30% всіх Інтернет-магазинів – більше, ніж будь-яка інша платформа. З WooCommerce, ви можете продавати як матеріальні так і цифрові товари усіх розмірів та форм, пропонувати варіації товарів, різноманітні конфігурації.

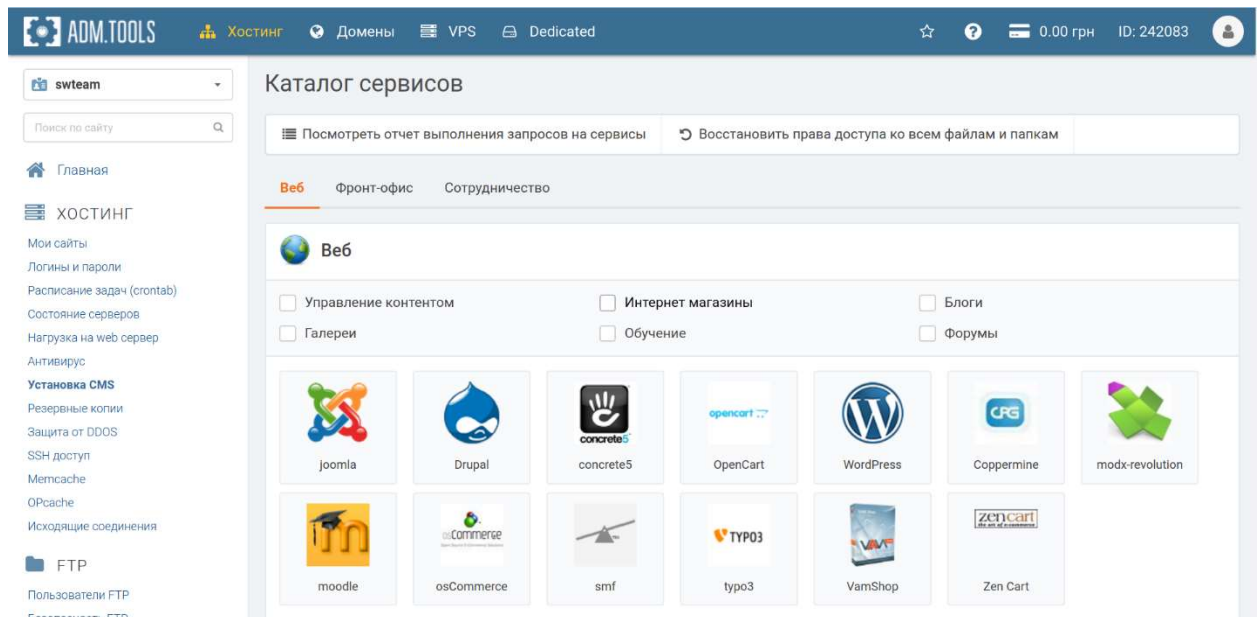


Рисунок 3.4 – Каталог сервисів CMS

Після вибору встановлення CMS WordPress треба вибрати адресу створеного сайту та створеної бази даних, та заповнити інші необхідні поля такі як пароль, логин, електронну пошту адміністратора, ім'я сайту та мову інтерфейсу (див. рис. 3.5).

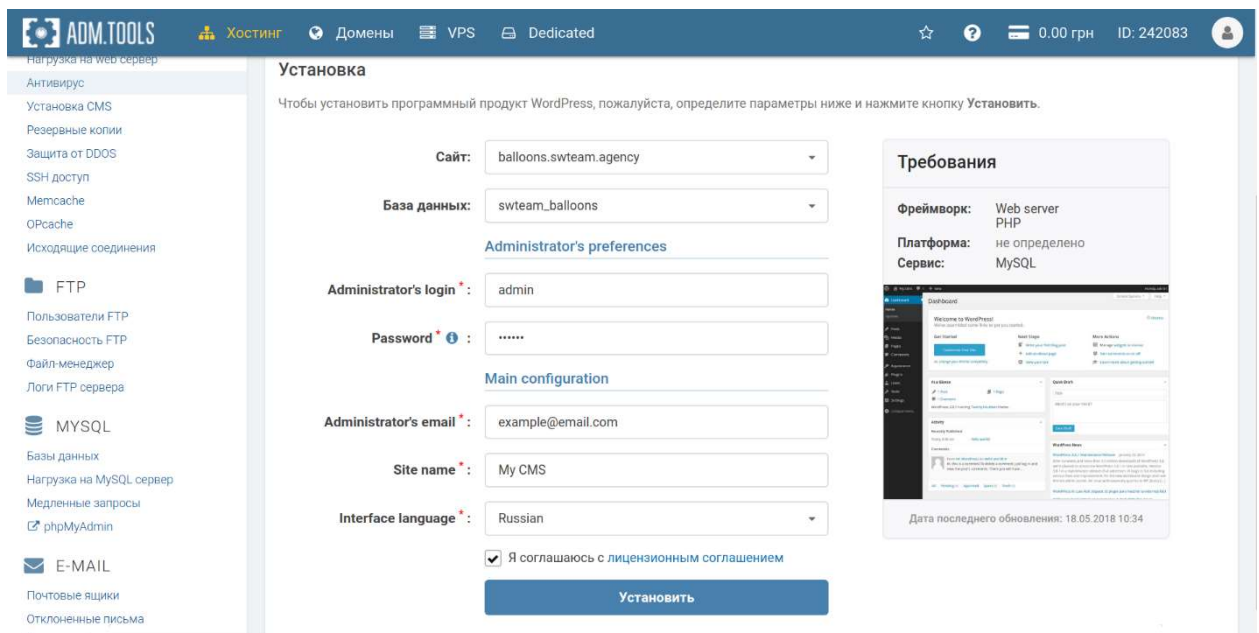


Рисунок 3.5 – Встановлення WordPress

Коли WordPress буде успішно встановлено наступним етапом буде встановлення WP Emerge. Для встановлення WP Emerge необхідний Composer.

Composer – менеджер пакетів прикладного рівня для мови програмування PHP що забезпечує стандартний формат для управління залежностями у програмному забезпеченні та необхідними бібліотеками.

Для початку потрібно завантажити стартову тему WP Emerge у папку з темами WordPress після чого у цій стратовій темі створити на основі config.json.dist файл конфігурації config.json.

Після чого треба виконати команду:

```
composer install
```

та дочекатися поки усі необхідні для роботи мікро фреймворку компоненти встановляться. Далі активувати тему у адмін панелі (див. рис. 3.6).

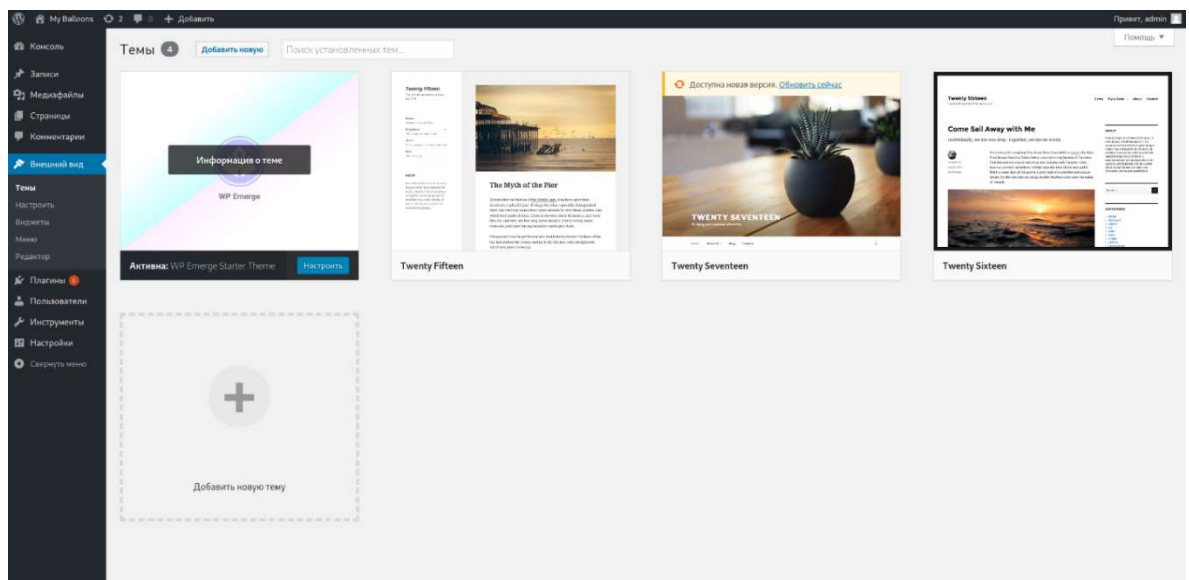


Рисунок 3.6 – Активована стартова тема

3.4 Структура проекту

Після того як усе необхідне встановлено та налагоджено є можливість розглянути структуру проекту (див. рис. 3.7).

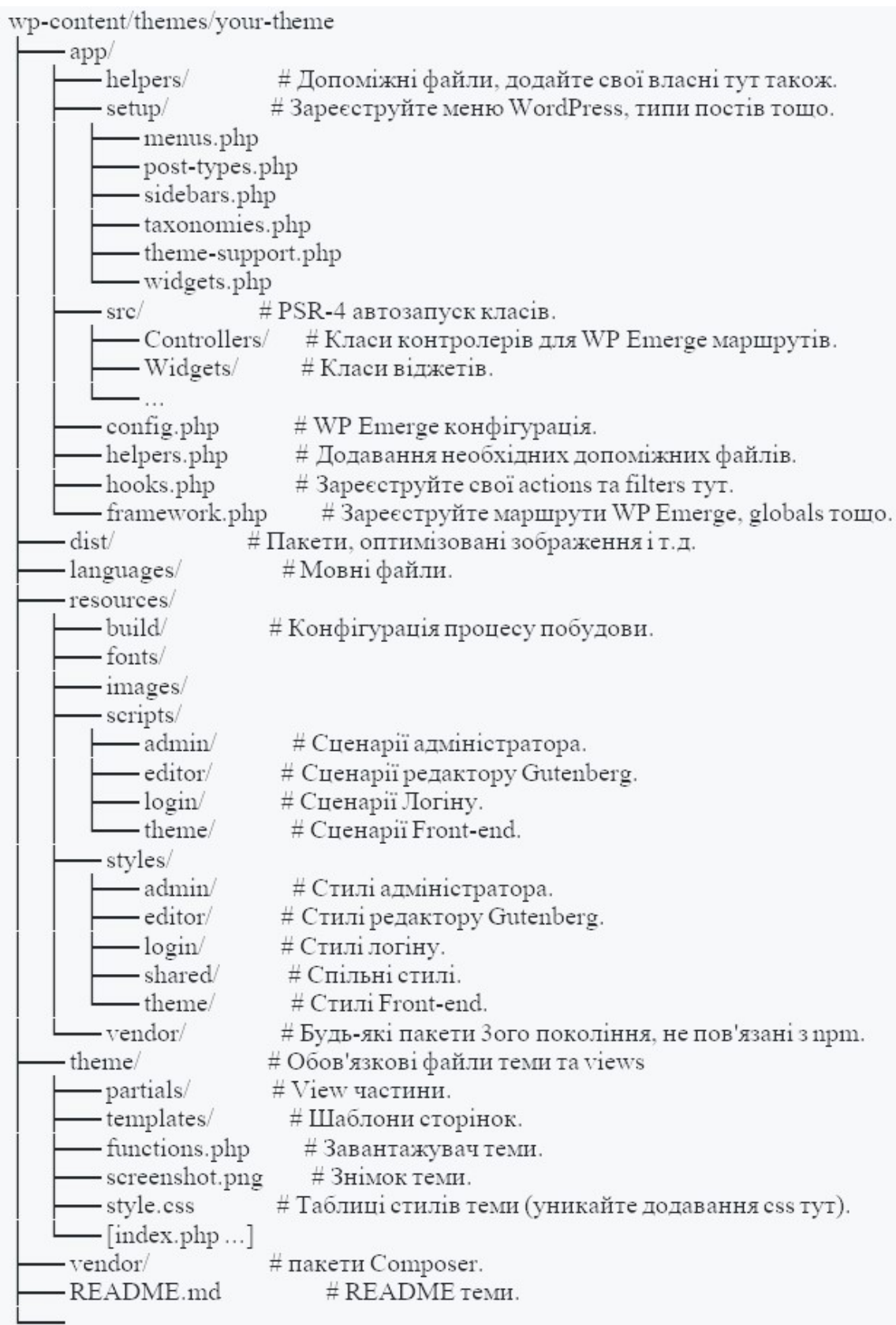


Рисунок 3.7 – Структуру проекту

Відомі каталоги

app/helpers/

Каталог для додавання допоміжних файлів PHP тут. Допоміжні файли повинні містити лише визначення функцій. Нижче наведено інформацію про те, де робити дії, фільтри, класи тощо.

app / setup /

Каталог для зміни файлів відповідно до потреб. Ці файли повинні містити реєстрації та декларації об'єктів WordPress, таких як типи публікацій, таксономії тощо.

app / src /

Каталог для додавання файлів класу PHP тут. Усі класи в області App \ namespace автоматично завантажуються відповідно до стандарту PSR-4.

resources/images/

Каталог для додавання зображення для стилю. Оптимізовані копії будуть розміщені в dist / images / під час виконання процесу збірки.

resources/styles/theme/

Каталог для додавання файлів .css та .scss, щоб додати їх до інтерфейсного пакету. Необхідно не забувати про @import їх в index.scss.

resources/styles/[admin,editor,login]/

Ці директорії призначені для адміністраторів, редакторів та груп входу відповідно. Вони працюють однаково з основними resources/styles/theme/directory.

resources/scripts/theme/

Каталог для додавання файлів JavaScript-файли тут, щоб додати їх до зовнішнього блоку. Точка вводу – index.js.

```
resources/scripts/[admin,editor,login]/
```

Ці директорії призначені для адміністраторів, редакторів та груп входу відповідно. Вони працюють ідентично основним ресурсам:

```
resources/scripts/theme/ directory  
theme/.
```

Каталог для додавання view теми, додаються у theme/partials/ чи theme/templates/ відповідно. Треба уникати додавання будь-якої логіки PHP, якщо це не стосується макетування (логіка PHP повинна йти в допоміжні файли або контролери WP Emerge).

3.5 Застосування основного функціоналу

Маршрути з можливістю інтеграції правила перезапису

```
Router::get( '/', 'HomeController@index' );
```

```
Router::get( '/custom', 'CustomController@custom' )  
->query( function ( $query_vars ) {  
    return [  
        // WP_Query query vars ...  
    ];  
} );
```

```
Router::get( ['post_id', get_option('page_on_front')],  
'HomeController@index' );
```

```
Router::get( function() {  
    return is_front_page();  
}, 'HomeController@index' );
```

- увімкніть використання контролерів, щоб розділити вашу бізнес-логіку подалі від презентації;
- використовуйте існуючі маршрути або додавайте нові, переписавши їх;
- використовуйте вбудовані динамічні умови маршруту або визначайте свої власні налаштування;
- використовуйте анонімні функції для швидких одноразових умов.

Контролери WP Emerge

```
class HomeController {
    public function index( $request ) {
        $name = $request->get( 'name' );
        return app_view( 'templates/home.php' )
            ->with( [
                'welcome' => 'Welcome, ' . $name . '!',
            ] );
    }
}
```

- відокремити незв'язану бізнес-логіку в контролерах та пов'язану з ними бізнес-логіку в методах контролерів;
- отримувати об'єкт, що представляє поточний запит, і відповісти на відповідь PSR-7;
- використовуйте різні методи для різних маршрутів;
- відповідайте за допомогою перегляду, json, перенаправлення тощо;
- легко перевірити.

Посередники WP Emerge

```
Router::get( '/', 'HomeController@index' )
    ->add( function( $request, $next ) {
        // perform action before
        $response = $next( $request );
        // perform action after
        return $response;
    } );
```

- хук перед та / або після обробки маршрутів (наприклад, контролерів);
- додати глобально або до певних маршрутів або груп маршрутів;
- можливості, такі як Flash і OldInput.

PSR-7 Відгук

```
class HomeController {
    public function index( $request ) {
        return app_response()
            ->withHeader( 'X-Custom-Header', 'foo' );
    }
}
```

- використовуйте об'єкти PSR-7 для ваших відповідей;
- легко потік та модифікація перед виведенням;
- використовує реалізацію Guzzle.

View Composers

```
View::addComposer( 'templates/about-us', function( $view ) {
    $view->with( ['hello' => 'world'] );
} );
```

- передайте загальний контекст для частинок незалежно від того, який контролер чи батьківський вигляд використовує їх;
- працюйте з будь-яким движком перегляду (Php, Blade, Twig).

Сервісний контейнер

```
// getContainer() used for brevity's sake – use a Service Provider instead.
$container = WPEmerge::getContainer();
$container['my_service'] = function( $container ) {
    return new MyService( $container['my_dependency'] );
};
```

- визначте свої залежності в службовому контейнері;
- перевизначити будь-які та всі WP Emerge залежно від необхідності;

- вмикає ін'єкцію залежності;
- використовує Pimple.

Постачальники послуг

```
class MyServiceProvider implements ServiceProviderInterface {
    public function register( $container ) {
        $container['my_service'] = function( $container ) {
            return new MyService( $container['my_dependency'] );
        };
    }

    public function boot( $container ) {
        // bootstrap code if needed
    }
}
```

- зареєструйте залежності в контейнері служби і завантажте їх, якщо необхідно;
- дозволяє логічно розподілити ваші залежності на окремих постачальників;
- власні залежності WP Emerge встановлюються через постачальників послуг.

Макети перегляду PHP

```
index.php
<?php
/**
 * App Layout: layouts/app.php
 */

if ( is_single() ) {
    Theme::partial( 'loop', 'single' );
} else {
    Theme::partial( 'loop' );
}
```

layouts/app.php

```
<?php
get_header();

if ( ! is_singular() ) {
    app_the_title( '<h2 class="pagetitle">', '</h2>' );
}

// Output the actual view content:
app_layout_content();

get_sidebar();

get_footer();
```

- виключіть повторювані виклики `get_header ()` / `get_footer ()` у кожному перегляді;
- зменшіть шапку та покращуйте зручність перегляду, не жертвуючи гнучкістю.

Підтримка користувацьки двигунів view

```
// getContainer() used for brevity's sake – use a Service Provider instead.
$container = WPEmerge::getContainer();
$container[ WPEMERGE_VIEW_ENGINE_KEY ] = function( $container )
{
    return new MyViewEngine();};
```

- замініть двигун перегляду, який використовується в сервісному контейнері;
- Blade and Twig доступні як додаткові пакети;
- ви навіть можете написати власний движок перегляду та використовувати його без проблем.

4 ПРОСУВАННЯ САЙТУ (SEO)

4.1 Просування сайту на WordPress

«70-80% користувачів ігнорують платні оголошення, приділяючи особливу увагу органічним результатам.» – imFORZA.

SEO або пошукова оптимізація – це методи і (або) стратегії, які використовуються для просування веб-сайту в пошукових системах. Головною метою SEO є збільшення числа відвідувачів, які приходять на сайт з пошукового трафіку.

Просування сайту складається з:

- технічних питань роботи сайту (виправлення помилок, збільшення швидкості роботи ресурсу, оптимізації вмісту і т.д.);
- розвитку сайту як бренду (збільшення відсотку впізнаваності ресурсу, зростання посилань на сторінки сайту, наявності відгуків і т.д.).

Поєднання цих складових і дозволяє піднімати позиції сайту в пошукових системах.

Отже, для успішної розкрутки сайту обов'язковою умовою є наявність якісного веб-сайту та унікального контенту. Після цього в гру вступають SEO фахівці, які мають спеціальні знання і володіють методами та способами просування сайту. Вони і займаються його вдосконаленням.

Сьогодні багато бізнесів обирають, в якості платформи, сайти на WordPress. Яких результатів можна досягти, просуваючи такі сайти? Одним з яскравих прикладів є зростання органічного трафіку сайту Kinsta, який використовує платформу WordPress (див. рис. 4.1).

Використання WordPress позбавляє від необхідності ручного застосування багатьох рутинних операцій і складних для розуміння технологій. Від користувача потрібні лише базові знання в галузі пошукової оптимізації, щоб зробити налаштування CMS.

У складі WordPress SEO пропонується два види таксономії: вертикальний рубрикатор (категорії) і горизонтальний варіант (мітки, теги). Основне завдання обох видів класифікатора – полегшити пошук контенту. Необхідно продумати логічну і цілісну структуру категорій і постаратися використовувати мітки найбільш ефективно, стежачи, щоб їх загальна кількість зберігалася в розумних межах.

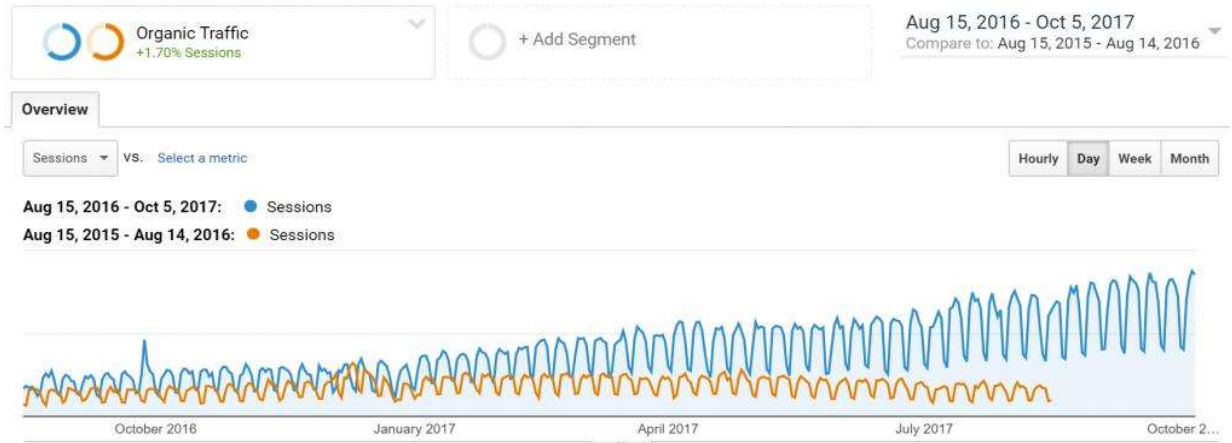


Рисунок 4.1 – Трафіку сайту

Зверніть увагу на те, що трафік збільшився на 571% менше ніж за 13 місяців. Без перебільшення, це чудовий результат просування! Крім того, на ресурсі втричі збільшилась кількість органічних ключових слів. І все це – завдяки системній роботі SEO фахівців (див. рис. 4.2).

Organic keywords



Рисунок 4.2 – Трафіку сайту

4.2 Наявність WWW у просуванні сайту

Під час запуску свого сайту, багато людей задаються питанням: а чи потрібно використовувати в його адресі фрагмент WWW (аббревіатура – “World

Wide Web”)? Деякі відомі веб-ресурси і досі використовують його в своїх адресах, хоча більшість вже повністю відмовилися від нього. Чи впливає наявність WWW на подальше SEO просування?

Правильна відповідь: ні, наявність WWW ніяк не впливає на розкрутку сайту.[9]

Тоді чому все ще існують ресурси, які використовують цю аббревіатуру? Раніше використання WWW було стандартним і обов’язковим для всіх веб-ресурсів. Для сайтів, які почали своє існування в ті часи, сьогодні досить складно “переїхати” на нову адресу без втрат. Тому багато старих брендів просто продовжують ним користуватись.

Ще однією причиною, чому великі компанії з хорошими показниками трафіку використовують WWW в адресі свого ресурсу, полягає в DNS (аббр. – Domain Name System). Справа в тому, що домени без WWW не можуть вести запис CNAME для перенаправлення трафіку на інший ресурс. Втім, сьогодні цей нюанс вже можна обійти.

Таким чином, використання WWW – це лише справа власних уподобань. В будь-якому випадку, не забудьте налаштувати переадресацію так, щоб користувачі могли переходити за двома адресами.

Наприклад, на сайт Lemarbet можна потрапити як на першу адресою www.lermarbet.com, так і по другому – lermarbet.com

У Google Search Console можна вказати бажаний домен для Google. Втім, якщо ви використовуєте плагін Yoast SEO (про нього мова піде далі), така установка не обов’язкова – він автоматично встановить теги і покаже Google яку версію сайту слід індексувати.

4.3 Встановлення плагіну для SEO на WordPress

Одним із важливих переваг використання платформи WordPress є можливість досить швидкої та якісної розкрутки сайту. З цією метою ми рекомендуємо встановити спеціальний плагін, який допоможе вам у цьому.

Найпопулярнішими є безкоштовні плагіни Yoast SEO і All-In-SEO Pack. У своєму виборі я зупинився на Yoast SEO, оскільки він:

- 1) Дозволяє аналізувати сторінки та ключові слова;
- 2) Автоматично генерує XML Sitemap;
- 3) Включає на сайті “хлібні крихти”;
- 4) Додає схему розмітки.

Крім того у нього є безліч інших переваг, які відкриваються в процесі роботи.

Всі подальші поради по просуванню сайту на WordPress, ми будемо ілюструвати за допомогою плагіну Yoast SEO, то ж рекомендуємо його встановити (див. рис. 4.3).

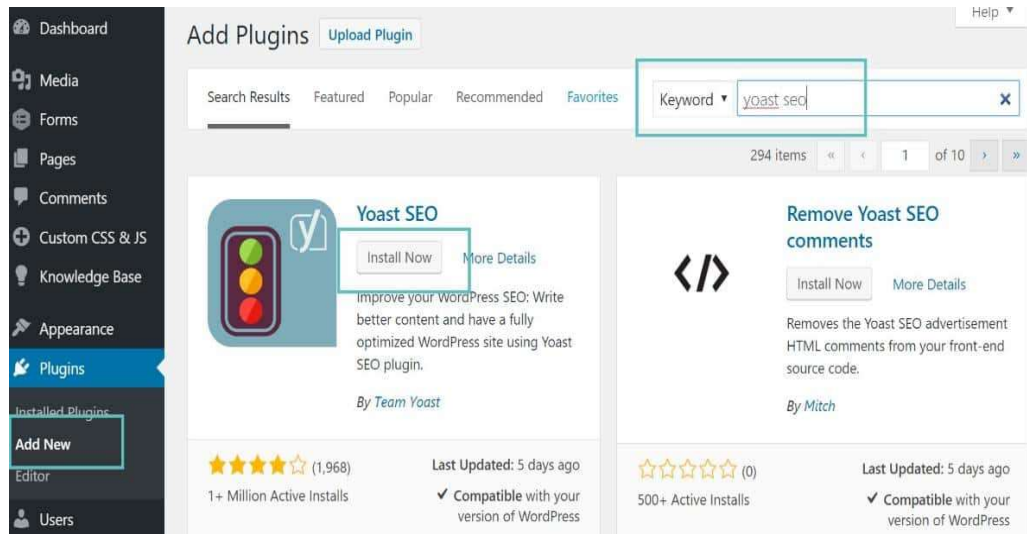


Рисунок 4.3– Плагін Yoast SEO

Крок 1. У розділі плагінів, панелі інструментів WordPress, необхідно натиснути «Додати». Після необхідно знайти Yoast Seo і натиснути «Встановити»:

Yoast SEO є оригінальним плагіном для WordPress SEO з 2008 року. Це улюблений інструмент для мільйонів користувачів, починаючи від пекарні за куточком до найпопулярніших сайтів на планеті. Завдяки Yoast SEO, ви отримуєте надійний набір інструментів, який допомагає вам досягти цілей першого місця в результатах пошуку. Yoast: SEO для всіх.

Крок 2. Потім натисніть «Активувати» (див. рис. 4.4):

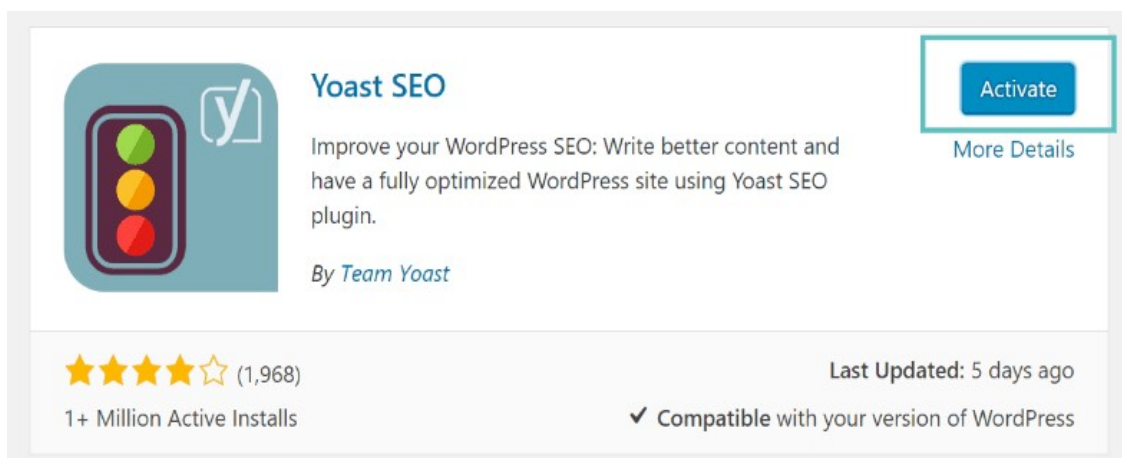


Рисунок 4.4 – Активування Yoast SEO

Крок 3. У вас з'явилося меню «SEO», в якому є доступ до налаштувань плагіну:

Плагін встановлено (див. рис. 4.5).

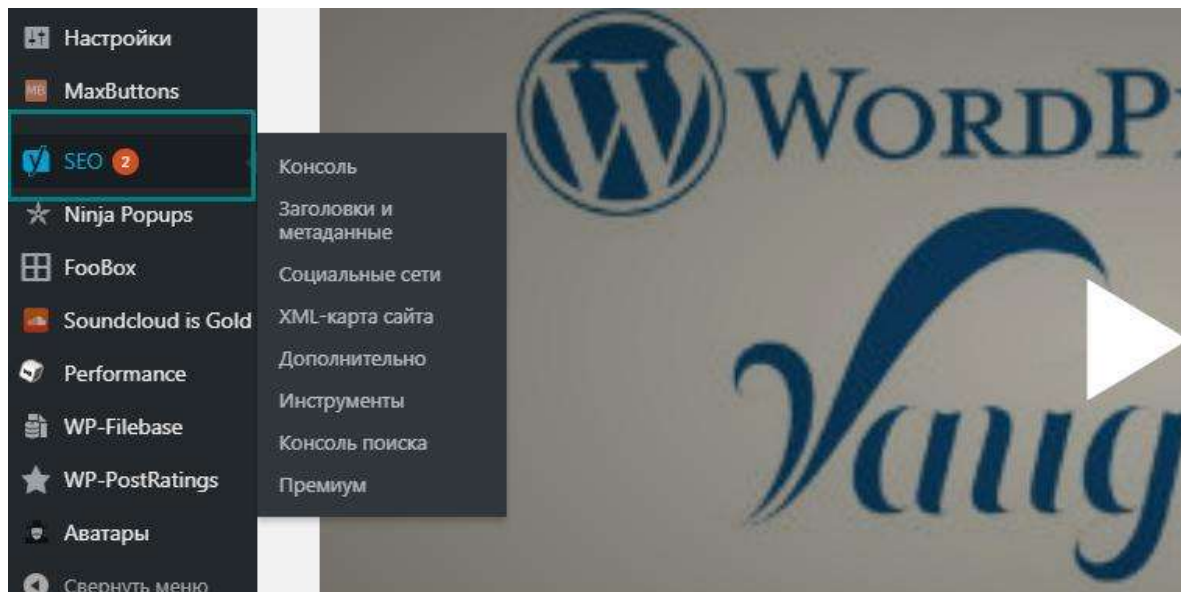


Рисунок 4.5 – Консоль Yoast SEO

4.4 Налаштування перманентних посилань

Після встановлення плагіну, необхідно правильно налаштувати перманентні посилання (або скорочено з англ. *permalinks* – “пермалінкі”).

Перманентними називаються постійні посилання на ваші сторінки в WordPress. Вони статичні і не змінюються в залежності від подальших налаштувань платформи. Це дозволить ділитися посиланнями, знаючи, що за ним завжди можна буде потрапити на сторінки сайту.

За замовчуванням, WordPress генерує посилання з досить дивною структурою. Вона включає в себе не тільки назву сторінки, але і дату.

Виглядає це ось так:

<http://wp.dev/2017/11/03/hello-world/>

Для просування сайту це далеко не найкраща структура URL-адреси, а для пошукового робота Google вона має величезне значення. Адреса сторінки повинна обов’язково містити заголовок і (або) ключові слова публікації. Більше того, бажано використовувати максимально короткі словосполучення.

Для пермалінків ми рекомендуємо давати назву публікації латинськими літерами.

Ось приклад, як повинна виглядати адреса сторінки:

`http://wp.dev/hello-world/`

Для того, щоб налаштувати такий тип посилань, необхідно в меню налаштувань обрати вкладку “Постійні посилання” (англ. – “Permalink”). Зі списку треба вибрати “Назва запису” (англ. “Post name”), після чого натиснути “Зберегти зміни”.

Важливо! Після внесення такої зміни всі попередні посилання втратять свою актуальність – за ними неможливо буде потрапити на ваш ресурс. Якщо ви тільки налаштовуєте свій сайт, то це нічим вам не загрожує, але якщо ж ваш сайт існує давно, обов’язково налаштуйте переадресацію з усіх попередніх сторінок.

Зробити це можна відредагувавши файл .htaccess або використовуючи сторонній плагін Change Permalink Helper[10].

4.5 Використовування ключових слів в текстах

Займаючись просуванням сайту варто пам’ятати про те, що для індексації сторінок Google використовує комп’ютерний алгоритм. Відповідно, ключові слова на сторінці все так само важливі, як і раніше. Різниця полягає лише в якості: якщо раніше достатньо було публікувати матеріали з максимальною кількістю ключових слів (як правило їх ставили внизу сторінок з текстом блідо-сірого кольору), то тепер пошуковий робот став більш уважним до змісту. Більш того, має значення частота, доречність і місце розташування ключових слів на кожній сторінці вашого сайту. Тому тексти потрібно писати так, щоб їх було, дійсно, цікаво і легко читати. При цьому, в кожному з них обов’язково мають бути присутніми ті ключові слова, за якими ви плануєте просувати свій сайт. Плагін Yoast SEO може допомогти вам у підготовці текстів.

Наприклад, ми можемо користуватись рекомендаціями, які дає плагін щодо обраного ключового слова на певній сторінці. Виглядає це наступним чином (див. рис. 4.6):

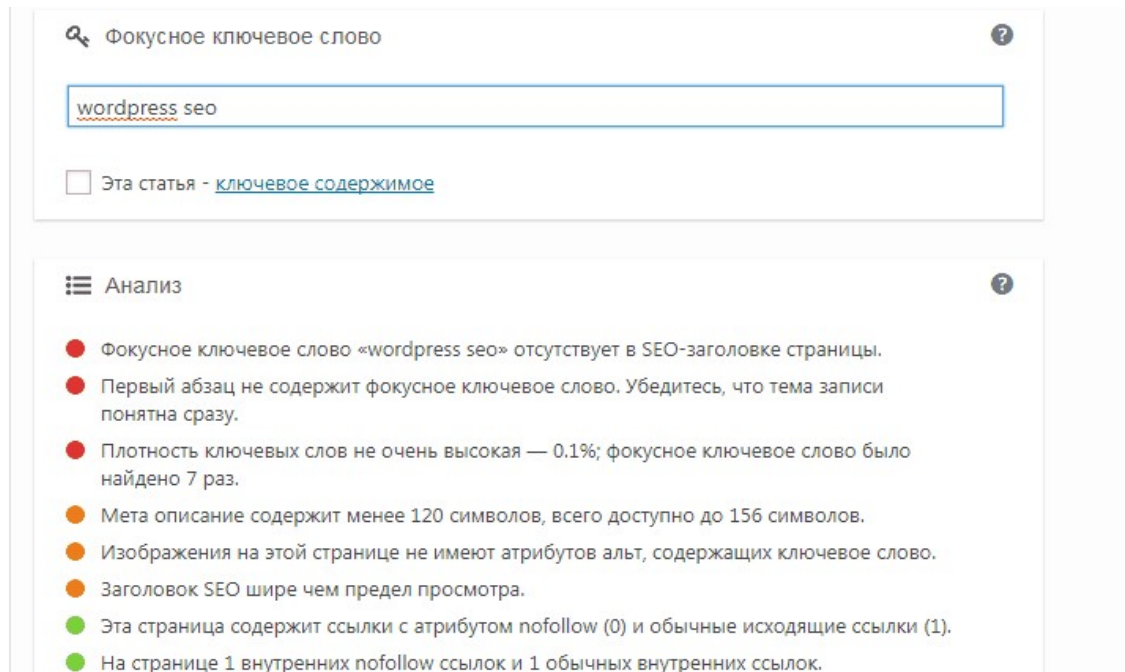


Рисунок 4.6 – Ключові слова Yoast SEO

Одним з важливих показників є щільність ключових слів на певній сторінці. В цілому, намагайтеся досягти показника в 0,5% – цього буде достатньо для того, щоб пошуковий робот зрозумів про що йде мова.

4.6 Теги Title

Говорячи про ключові слова, варто окремо зупинитися на тегах Title. Тег Title – елемент HTML, який визначає назву веб-сторінки. Тег Title відображається на сторінках результатів пошукової системи (SERPs) як заголовок для даного результату та є важливими для зручності використання, оптимізації пошукових систем та спільного використання. Тег Title веб-сторінки має бути точним та стислим описом змісту сторінки. Бажано, щоб в Title сторінки також перебували ключові слова (одне або словосполучення). Більш того, краще розмістити ключове слово на початку тега Title (хоча це і не завжди можливо). Також можливо встановити Title самостійно в плагіні Yoast SEO (див. рис. 4.7):

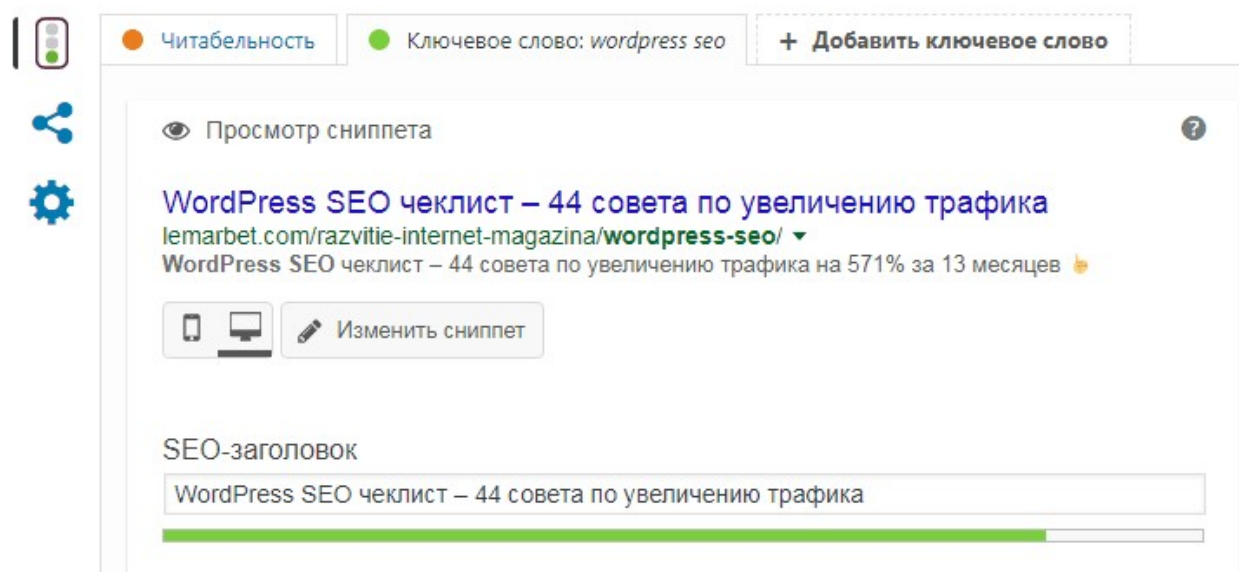


Рисунок 4.7 – Тег title у Yoast SEO

Важливо! Не забувайте про те, що видима межа в Google 65 знаків. Тому якщо ваш текст буде довшим – частково він не буде відображатись.

Варто пам'ятати про такий нюанс: з тегами Title пов'язаний ще один важливий показник – кількість переходів на сторінку (CTR). Збільшити CTR сторінки на 20% можна за допомогою спліт і A / B – тестування. Так ви не тільки дізнаєтесь який з варіантів Title відповідає запитам відвідувачів, але і покажете пошуковому роботу більш релевантний контент.

Для проведення таких тестів вам допоможуть плагіни WordPress Title Experiments і Nelio AB Testing.

Для Google все ще важливо, щоб ви приділяли увагу оптимізованим тегам H1 і Title, оскільки навіть невеликі зміни в них дають значні результати та сприяють зростанню органічного трафіку.

4.7 Збільшення CTR за допомогою Meta Description

Meta Description не впливає на ранжування в пошуковому трафіку, але впливає на показник CTR. Іншими словами: опис сторінки не важливий для пошукового робота, але дуже важливий для відвідувачів. Якісний опис може спонукати користувача перейти на певну сторінку. Побічно це може вплинути на рейтинг вашого ресурсу, адже якщо люди переходить на сторінку, значить вона відповідає запиту.

Налаштувати вручну Meta Description ви також можливо за допомогою плагіна Yoast SEO (див. рис. 4.8):

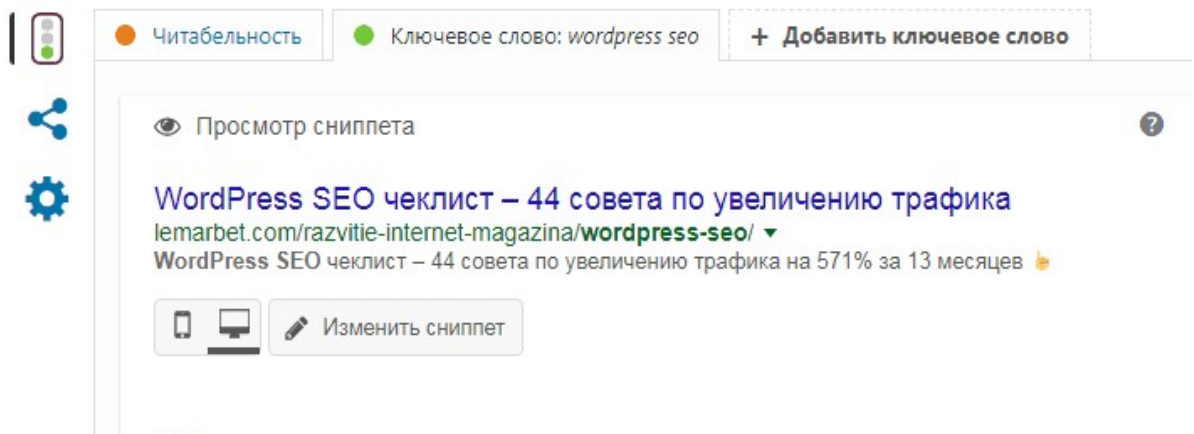


Рисунок 4.8 – Meta Description у Yoast SEO

Важливо! Довжина Meta Description має обмеження в 156 символів. Решта тексту не буде відображатись у результатах пошукової видачі.

Обов'язково вкажіть ключове слово в Meta Description. Якщо пошуковий запит користувача відповідає вашому ключовому слову, Google позначає таке слово жирним шрифтом (див. рис. 4.9):

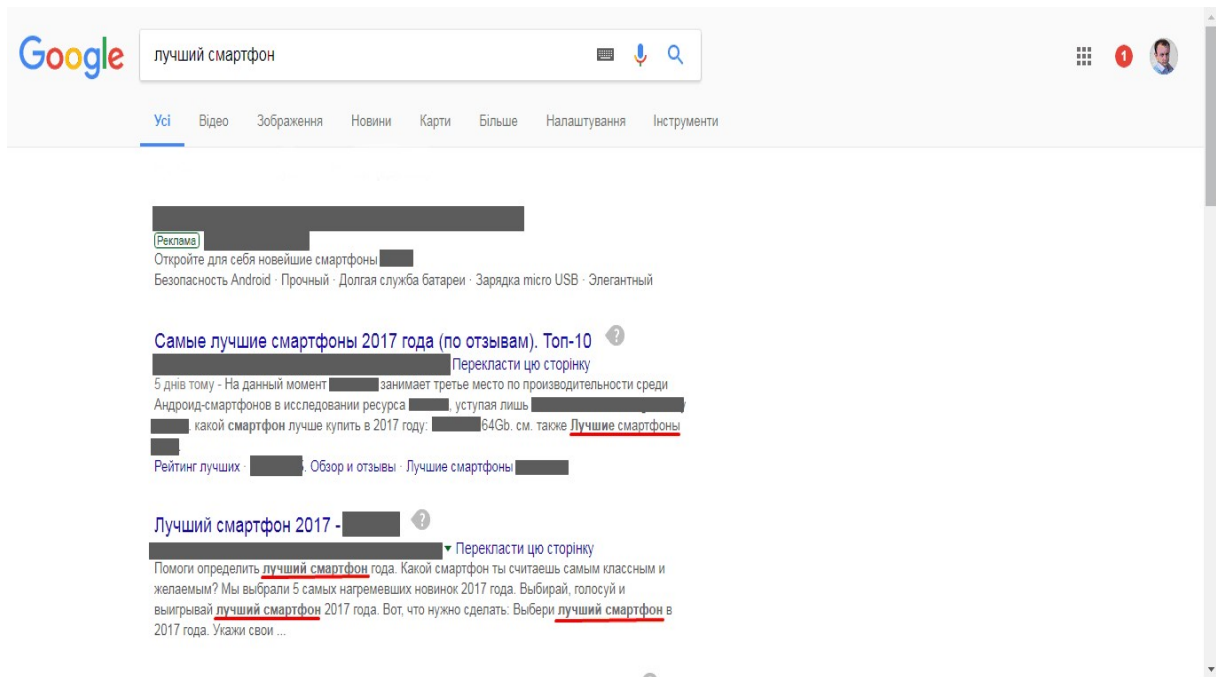


Рисунок 4.9 –Ключове слово в Meta Description

4.8 Meta Keywords

Вже ні для кого не секрет, що Meta Keywords залишилось в минулому і ніяк не впливає на просування сайту.

Хоча є спостереження деяких SEO фахівців, які підтверджують, що декілька пошукових систем все ж використовують дані з Meta Keywords для ранжування сайтів. Це Yahoo, Baidu і Yandex[11].

Включити або вимкнути Meta Keywords можна також за допомогою плагіна Yoast SEO:

Крок 1. В меню плагіну “SEO” необхідно відкрити розділ “Заголовки і Мета-теги”. Потім перейдіть на вкладку “Інше”. В налаштуваннях є питання “Використовувати метатег keywords?” (див. рис. 4.10):

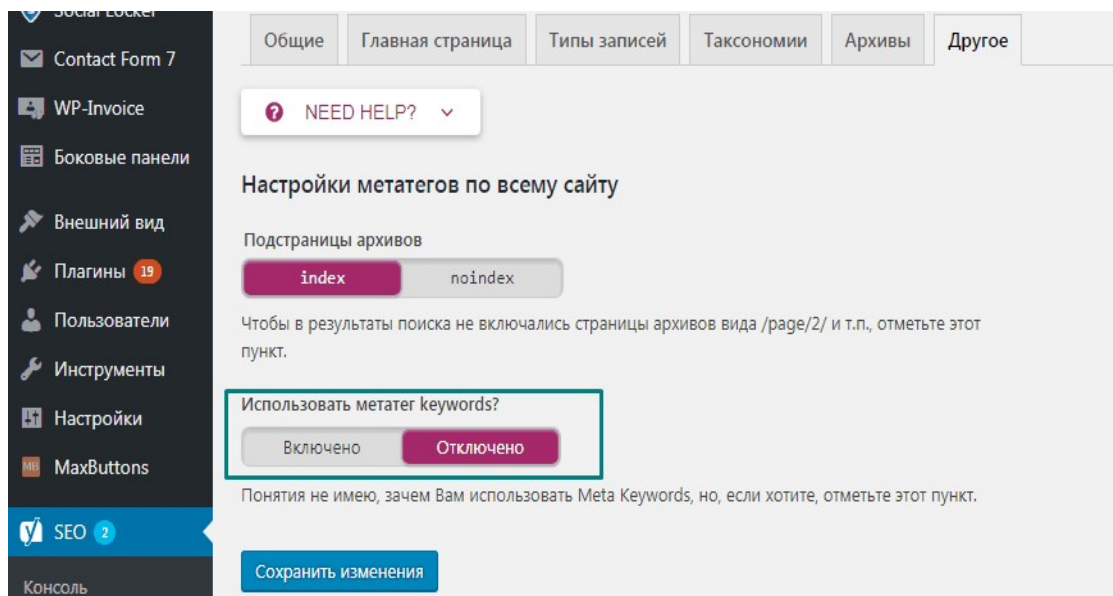


Рисунок 4.10 – Ключовое слово в Meta Description

Крок 2. Якщо ввімкнути Meta Keywords, то на сторінках з публікаціями з’явиться новий розділ “МЕТА ключові слова”, в якому можна вводити список з ключовими словами, які просуваються (див. рис. 4.11):

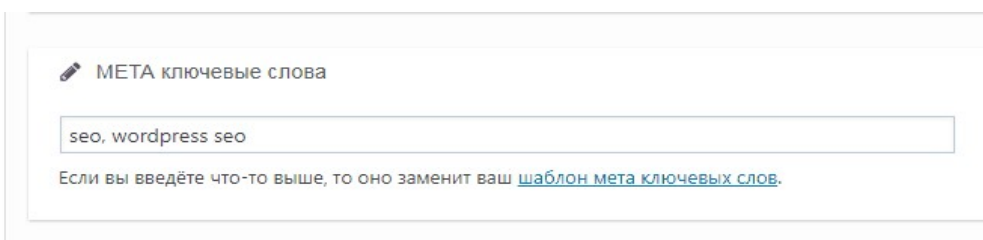


Рисунок 4.11 – Ключовое слово в Meta Description

Якщо ж цільова аудиторія використовує Google або Bing, тоді краще не витрачати час на створення таких списків.

4.9 Теги H1, H2, H3

Для того, щоб визначати релевантність контенту сайту, пошуковий робот Google перевіряє теги всіх заголовків HTML-сторінки. H1, H2, H3 і т. д.

Оптимальним є наявність одного тегу H1 на одній сторінці. Тегів H2 і H3 може бути декілька, але вони повинні зберігати ієрархію вкладеності: H1 – це назва тексту, H2 – заголовки розділів, H3 – заголовки підрозділів і т.д.

Тег H1 є одним з найбільш важливих, то ж обов'язково повинен містити основне ключове слово. У додаткових заголовках H2, H3 також може бути вказано ключове слово, але це не обов'язково.

Заголовки в текстах використовуються не тільки для легкості читання, але також і в якості додаткового способу просування сайту.

Встановлювати теги заголовків можна за допомогою редактора текстів (див. рис. 4.12):

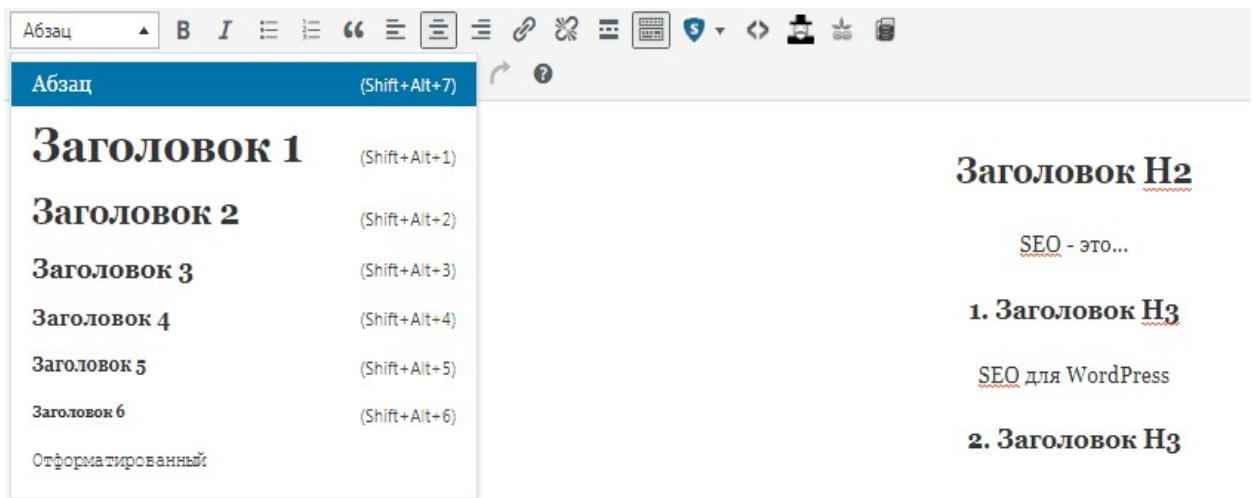


Рисунок 4.12 – Теги заголовків

Для швидкого встановлення певного тегу в редакторі, ви можете використовувати наступні комбінації клавіш:

ALT + SHIFT + 2 (заголовок H2)

ALT + SHIFT + 3 (заголовок H3)

4.10 Кількість слів

Фраза “Контент – це король”. Вона так само актуальна, як і раніше. Що шукають багато користувачів інтернету? Відповіді на свої питання, вирішення проблем, розваги і т.д. Значна частина, доступного людям, контенту – текстова. Тому всі пошукові системи, з року в рік, намагаються вдосконалювати алгоритми швидкого пошуку необхідного контенту.

Якщо розміщуються на сайті якісні, оптимізовані тексти, значить ресурс не тільки цікавий читачам, але й має певну вагу для пошукових систем.

Що потрібно пам’ятати, створюючи новий контент? Те, що цікаві і великі тексти мають перевагу, над маленькими нотатками або новинами. Хоча в усьому, звичайно ж, має бути міра.

Ще в 2012 році на блозі serpIQ було опубліковано дослідження про те, чи впливає обсяг тексту на результати пошукової видачі. Ось одна з ілюстрацій, яка доводить, що перші позиції утримують сайти, на яких розміщені довгі тексти (див. рис. 4.13):

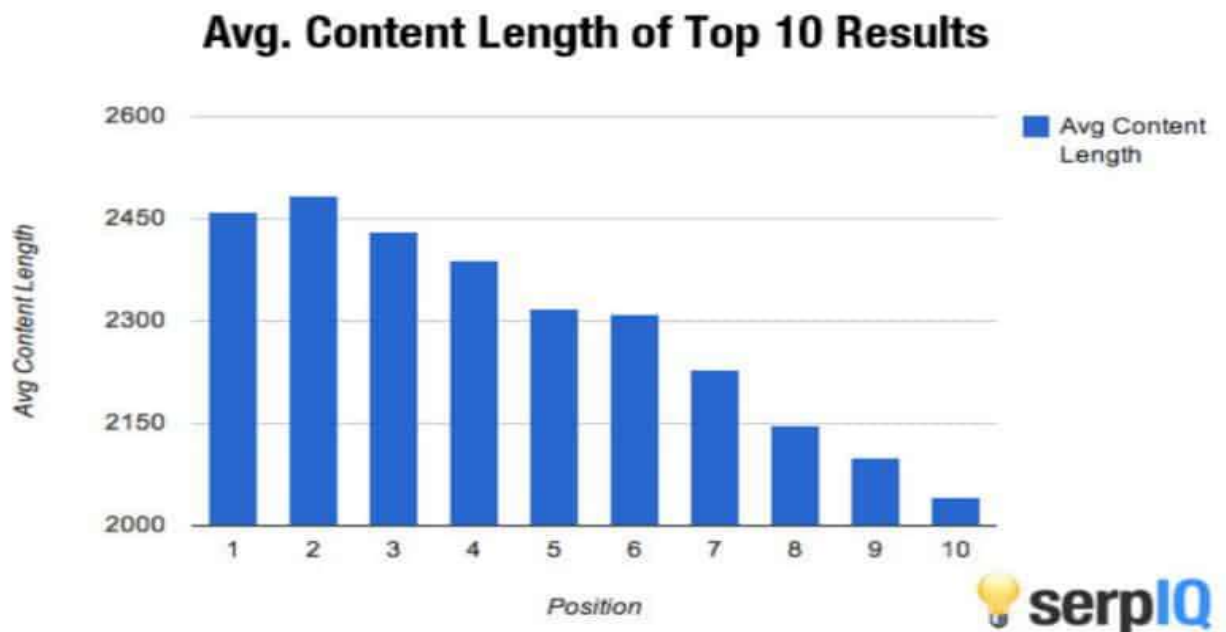


Рисунок 4.13 – Результати пошукової видачі

Так, цьому графіку більше 5 років, але він не втратив актуальності.

Інша компанія – CanIRank, опублікувала на початку 2017 року власне дослідження про вплив розмірів тексту на просування сайту.

Вони звернули увагу на залежність між довжиною контенту та кількістю зворотних посилань на нього. Але головним висновком стало підтвердження, що довжина тексту все ж має значення для просування сайту (див. рис. 4.14):

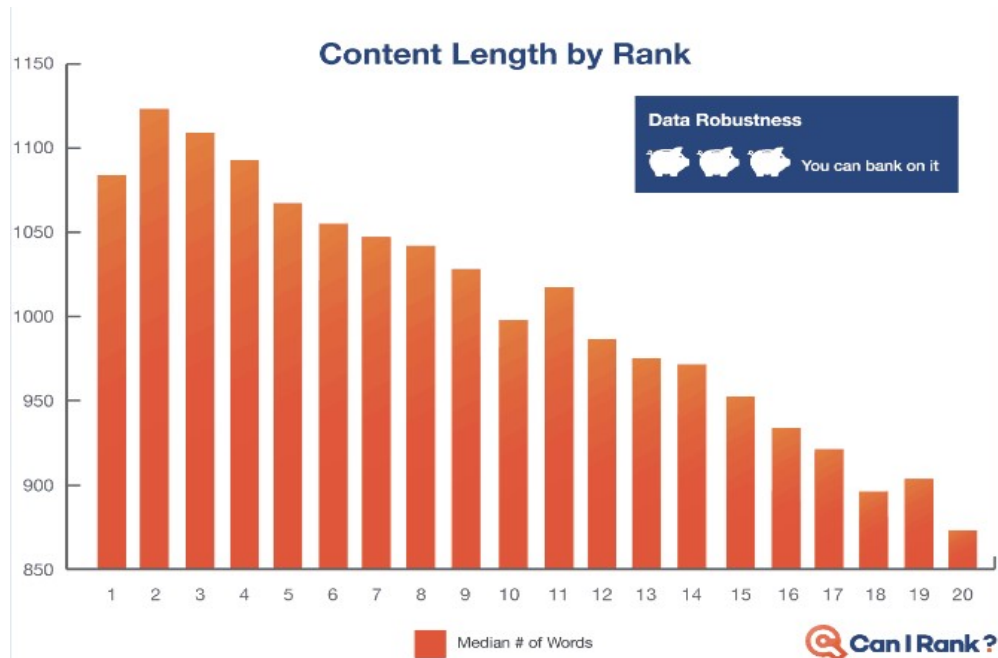


Рисунок 4.14 – Довжина тексту

Ще одна перевага лонгвідів в тому, що довгі тексти, як правило, мають безліч синонімів ключових слів, які просуваються. В результаті, такий текст ранжується не тільки за кількістю ключів, але і по всім синонімам, що є додатковою перевагою на тлі конкурентів.

4.11 Використання коротких URL-адрес

Однією з переваг сайтів на платформі WordPress є можливість швидкого просування в пошуковій системі Google. Зокрема, платформа WordPress дозволяє встановлювати короткі URL-адреси.

Практика доводить: довжина URL-адреси впливає на CTR. Користувачі швидше переходять на сторінки з короткими і зрозумілими посиланнями, ніж на сторінки з заплутаними назвами в URL-адресах.

За замовчуванням, WordPress генерує URL-адреси на основі назви статті.

Наприклад:

<https://lemarbet/blog/wordpress-seo-checklist-50-tips-2017/>

Є можливість скоротити його до короткої фрази або навіть до одного слова (яким може бути і те, за яким просуваємося).

Крок 1. На сторінці з публікацією в редакторі, є рядок, в якому зазначене постійне посилання на статтю. Натисніть на кнопку “Змінити”, яка знаходиться біля неї (див. рис. 4.15):

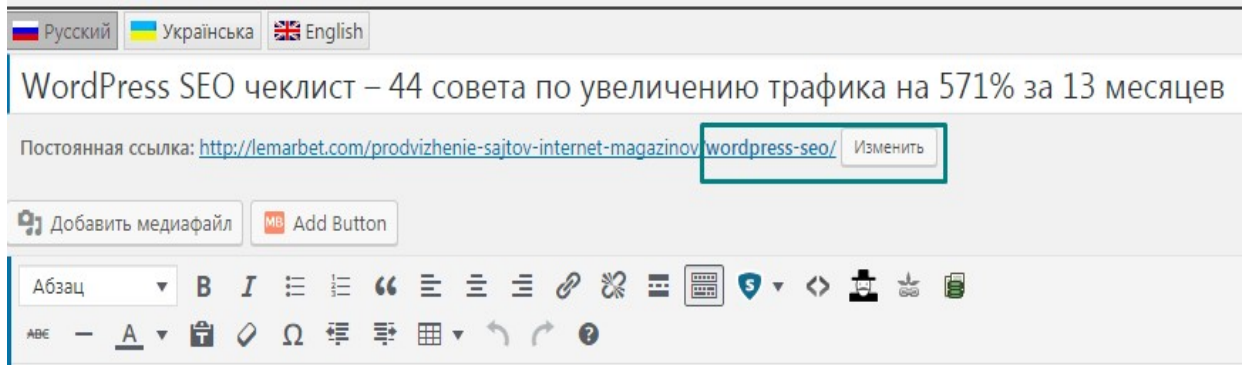


Рисунок 4.15 – Зміна url

Крок 2. Ввести свою URL-адресу. Обов’язково необхідно розділити кожне слово дефісом. Зберегти результат.

У підсумку, отримуємо більш читабельну URL-адресу:

<https://lemarbet/blog/wordpress-seo/>

Важливо! Для успішної розкрутки сайту, кожна URL-адреса в WordPress повинна бути унікальною.

4.12 Використання розумних назв для зображень

Типовою помилкою багатьох власників сайтів на платформі WordPress є неправильна назва зображень. Як не дивно, але коректна назва також має значення для просування сайту.

Для того, щоб на вашому ресурсі не виникало подібних помилок, варто пам’ятати кілька правил:

Файл із зображенням повинен мати назву, яка за змістом пов’язана зі змістом тексту. Наприклад, замість DC0000.JPG назвіть картинку wordpress-seo.JPG.

Завжди додавати дефіси між словами в назвах зображень. Наприклад, такі назви як `wordpress_seo.JPG` або `wordpressseo.JPG`, Google не зможе прочитати.

Google сприймає слова з дефісами як окремі. Якщо є бажання надати назву з двох слів, правильно це зробити ось так: `wordpress-security.JPG`.

Якщо до картинки можна застосувати ключове слово – необхідно вказати його і там. Як мінімум, воно може бути на головному зображенні.

4.13 Використання переваг HTTPS для розкрутки

Пошукова система Google офіційно заявила про те, що HTTPS є одним з факторів ранжування сайтів. Відповідно, це необхідно враховувати при плануванні просування свого ресурсу.

Цікаво, що в результаті аналізу 1 мільйона URL-адрес, виявилось, що понад 33% перших позицій сторінок в Google використовують HTTPS (див. рис. 4.16).

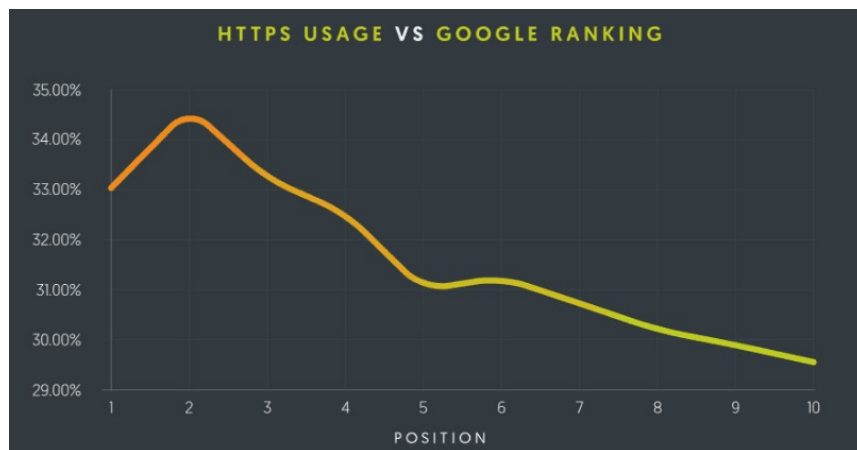


Рисунок 4.16 – Використання HTTPS

4.14 Опис Google Knowledge Graph

Google Knowledge Graph – це особливий тип розмітки, який відображається праворуч від списку з результатами пошукової видачі і більш детально розповідає про певних людей, місця і т.д (див. рис. 4.17).

Така інформація краще сприймається користувачами інтернету, оскільки повністю дає відповіді на цілий ряд можливих питань.

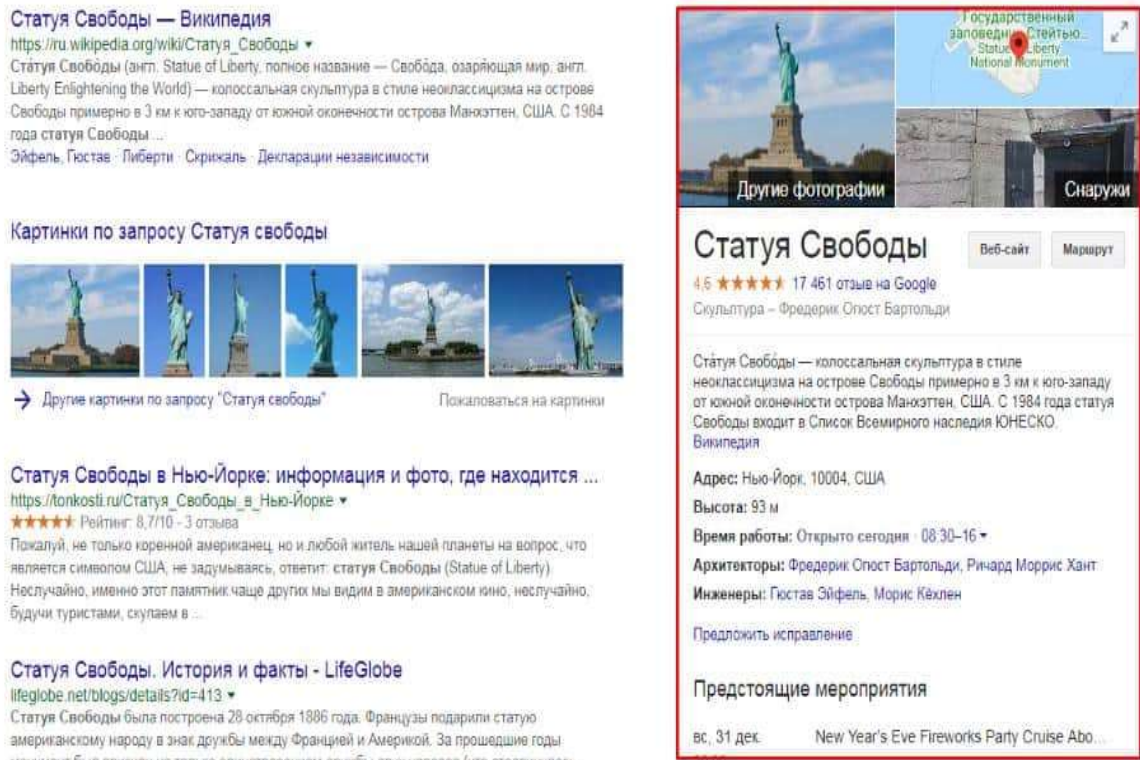


Рисунок 4.17 – Google Knowledge Graph

Для того, щоб розмістити інформацію про свою компанію, на сайті потрібно розмістити Schema Markup, використовувати теги в заголовках, а також марковані списки. Правда, ніхто не може дати 100% гарантії, що це спрацює відразу і ваш сайт буде представлений таким чином.

4.15 Опис Google Analytics

Установка коду Google Analytics є першим кроком для всіх типів сайтів. Потрібно зв'язати облікові записи Google Analytics і Google Search Console.

В налаштуваннях властивостей для домену необхідно натиснути «Search Console Settings», і він допоможе налаштувати таку зв'язку.

Після цього, необхідно додати додатковий розділ в свій обліковий запис Google Analytics, за допомогою якого буде можливість бачити пошукові запити та рейтинг найпопулярніших сторінок на сайті.

Ще один розділ, на який слід звертати увагу, це канал «Organic Search» – трафік з пошукових систем (див. рис. 4.18).

Є багато SEO факторів, які слід враховувати під час запуску WordPress сайту. Очевидно, що це не найпростіше завдання (особливо для новачків), адже принципи пошукової оптимізації змінюються досить часто[12].

Органічний пошук – це спосіб введення одного або декількох умов пошуку як окремої рядка тексту в пошукову систему. Результати пошуку в органічному пошуку, відображаються як списки сторінок, на основі релевантності пошукових термінів; і виключити рекламу. Враховуючи, що неорганічні результати пошуку не відфільтрують плату за клік рекламу.

Насправді, з точки зору SEM-аналітики питання співвідношення і взаємодії органічних і платних результатів пошуку (і як наслідок – органічного і платного трафіку) є основоположними.

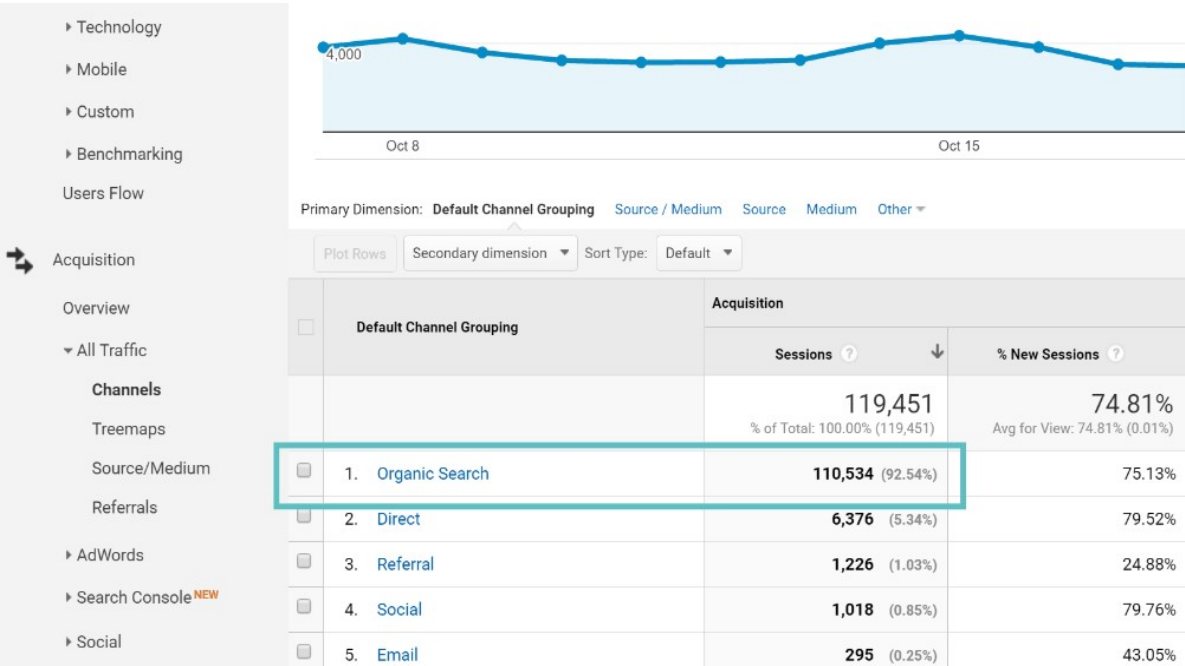


Рисунок 4.18 – Налаштування Organic Search

5 УПРАВЛІННЯ ПРОЕКТОМ

Розробка проекту інтернет-магазину керувалася за етапами сценарію наведеному нижче.

Нижче окреслено "ідеальний сценарій" для оцінок за часом, однак існує лише кілька проектів, які відповідають ідеальним клієнтам, ідеальним контекстом, ідеальним часом і досконалістю. Набагато важливіше мати чіткі очікування, прозорість та максимально освічені складності, пов'язані з індивідуальним дизайном, розробкою та продуктивністю.

Планування проектування та розробки веб-сайтів слід розділити на наступні етапи.

- вступний рівень веб-сайту: 6 – 10 тижнів;
- базовий веб-сайт: 8 – 12 тижнів;
- стандартний веб-сайт: 10 – 14 тижнів;
- розширений веб-сайт: 14 – 18 тижнів;
- веб-сайт підприємства: 18 – 24 тижні.

План проекту веб-сайту Хронологія Оцінки базуються на:

- 3 раунди змін до дизайну;
- відгуки та оновлення надаються протягом 3 робочих днів від клієнта;
- вміст 100% завершено фазою 4;
- додаткові функції не додаються до проекту середнього проекту;
- затвердження, отримані протягом 3 робочих днів запиту;
- проект перебуває на початковій часовій шкалі (при будь-якій причині слід відновити початкову шкалу часу, розподіл ресурсів буде перерваний, і ваш проект буде жонглюватися в буферному просторі, аніж завдання початкового ресурсу та плани ведення).

Фаза 1: планування

- 1) стратегія та засідання проекту;
- 2) засідання зацікавлених сторін всередині;
- 3) технічні характеристики Короткий опис;
- 4) схема технології.

Фаза 2: Створення проекту та вирівнювання ресурсів

- налаштування проекту в системах обліку;
- налаштування проекту в системі управління проектами (дизайн, розробка, SEO, соціальні медіа, мобільні додатки та інше – залежно від рівня послуг, на які підписано);

- встановлення завдань, відповідальних сторін, етапи та результати;
- налаштування сервера та встановлення належного програмного забезпечення.

Фаза 3: Дизайн

- стратегічний сеанс та стильний дизайн (1 – 2 години);
- засідання зацікавлених сторін;
- дизайн домашньої сторінки V1;
- розробка дизайну домашньої сторінки та раунди переходів * V2, V3;
- дизайн інтер'єру сторінок (Про нас та зв'язок з нами) V1;
- внутрішній дизайн дизайну сторінок та раунди змін * V2, V3;
- розбір блогів та блог Post Design V1;
- розгляд блогів та публікації дизайну блогів та раунди змін * V2, V3.

Термін виконання даної фази може займати мінімум 10 (десять) робочих днів, а необмеженого часу – в залежності від часу відгуку клієнтів, чіткості, рішучості та обсягу змін, що входять до їх інвестиційного плану. Найперше, що може вплинути на часову лінійку на етапі проектування, – це не надання своєчасних / чітких редагувань.

- Версії 1 – 3 можна виконати через 10 робочих днів;
- 4 – 6 версій можна зробити через 11 – 20 робочих днів;
- 7 – 9 версій можна зробити через 21- 30 робочих днів;
- 10 – 12 версій можна зробити через 31 – 40 робочих днів.

Загальний огляд основного етапу розробки. (Якщо є проміжний або розширений веб-сайт, саме там буде користувальницьке кодування для розширених функцій).

- 1) скрізь всі ваші дизайн PSD (PSD = Photoshop Designs);
- 2) встановити базу даних, WP тощо;
- 3) розробка глобальних таблиць стилів (CSS) на основі узгодженого дизайну стилю (для одного браузера – Chrome і настільних комп'ютерів);
- 4) розробка та кодування домашньої сторінки:
заголовок, настроювані Nav / меню, повзунки Ajax / JQuery, віджети, тіло, спеціальні елементи, Footer;
- 5) розробка та кодування інтерфейсу сторінки:
сторінки, тіло, бічні панелі, віджети тощо;
- 6) розділ блогів, публікації категорії, архіви повідомлень, останні публікації, індивідуальні повідомлення, бокові панелі, СТА, віджети.

Час для цієї фази може займати мінімум 15 (п'ятнадцять) робочих днів і

необмежений термін, залежно від часу відгуку клієнтів, чіткості, рішучості та кількості на замовлення сторінок або спеціальних функцій, що входять до їх інвестиційного плану. Чим більше користувацький дизайн / спеціально розроблені сторінки та користувацька функціональність у вас, тим довші фази розробки. Різновиди речей, які будуть охоплювати час розробки та поставити вас у проміжну або розширений термін проекту, будуть, електронна комерція, системи членства, форуми, системи резервування, системи лістингу, каталоги, функції відображення та ін.

Фаза 5: Розробка контенту

Цей етап може бути одним із самих складних для прогнозування часу. Коли ви плануєте новий дизайн та розробку веб-сайту, не забувайте відкласти бюджет спеціально для CONTENT. Це найменша частина веб-проекту та один з найважливіших елементів. Це може спричинити найбільші затримки і, якщо це не зроблено правильно, найчастіше втрачають можливості за допомогою нового веб-сайту.

Весь процес передачі, створення та керування вмістом буде залежати від рівня ваших інвестицій в SEO та цілей. Чим агресивніше ви ставитесь до місця розташування, тим більше продуктів та послуг, які ви хочете знайти, а також більше місцеположень, які ви хочете знайти, буде DRAMATICALLY збільшити час, необхідний для цієї фази.

Фаза керування вмістом складається з наступних кроків:

- 1) створення навігації;
- 2) оптимізація навігації;
- 3) створення змісту (ми будемо запитувати відсутній вміст – ви можете написати та надати або передавати на замовлення вміст у нашій команді);
- 4) передача вмісту;
- 5) нові завантаження вмісту;
- 6) стильний вміст (це також включає в себе всю роботу SEO, яка необхідна для створення сторінок для SEO);
- 7) дослідження зображень, збір, придбання та публікація зображень;
- 8) зв'язок інтер'єру та домашньої сторінки;
- 9) інтеграція та зв'язок соціального облікового запису;
- 10) інтеграція та тестування форм.

Весь вміст повинен бути отриманий протягом 5 днів після завершення фази розробки. Команда передасть весь вміст з вашого існуючого веб-сайту

на ваш новий веб-сайт. Також буде додано вміст, який буде надіслано протягом 5 днів після завершення вашої фази розробки. Якщо є будь-які додаткові зміни чи зміни до веб-сайту, можливо буде легко внести ці зміни через систему керування вмістом нового веб-сайту.

Фаза 6: Адаптивний / Мобільний та Крос Браузерний

- 1) реконструйте веб-сайт для всіх веб-переглядачів (IE, Firefox, Safari тощо);
- 2) редизайн веб-сайту для реагування (мобільні розробки, планшети тощо).

Фаза 7: Гарантія якості – До оживлення сайту

- 1) огляд сайту;
- 2) тестовий заголовок;
- 3) перевірка навігації;
- 4) тестові форми;
- 5) тестові посилання;
- 6) тестова функціональність;
- 7) тестовий нижній колонтитул;
- 8) перегляньте налаштування сайту;
- 9) favicon;
- 10) перевірте соціальні посилання;

Фаза 8: на сторінці SEO

- 1) оптимізація URL-адреси;
- 2) на сторінці SEO (час буде змінюватись залежно від вашого плану та того, що включено);
 - а) теги заголовка (H1, H2, H3);
 - б) заголовки сторінок;
 - в) мета описами;
 - г) мета-ключові слова;
 - д) категорії / мітки;
 - е) внутрішні з'єднання;
- 3) оптимізація зображень.

Фаза 9: Підготовка до оживлення сайту

- 1) підготовчий сервер для Go Live;
- 2) налаштування записів MX для хостингу електронної пошти;
- 3) перемістити файли в середовище Live Server;
- 4) змінити посилання DNS;
- 5) розповсюдження домену – може зайняти 3 дні;

- б) тренінг для клієнтів – як використовувати CMS (веб-сайт системи управління контентом).

Фаза 10: Аудит після оживлення сайту

- після того, як Сайт буде Live, відбудуться такі події Live Live;
- остаточний перехресний браузерний аудит;
- мобільний відгук аудиту;
- тестування форм;
- тренінг для клієнтів;
- сприяння запуску сайту (соціальні медіа, блоги, електронні бюлетені та інше – залежить від вашого маркетингового плану);
- надайте хостинг, підтримку, необхідні оновлення, технічне обслуговування;
- повний звіт про помилки сайту та виправлення;
- відгуки веб-майстрів;
- представлення сайту в пошукових системах;
- підвищення безпеки запитаних форм;
- 1 година Додаткової підготовки до WP.

Фаза 11: Стимулювання соціальних медіа

- 1) маркетингові зображення, призначені для соціальних пропозицій запуску нового веб-сайту;
- 2) акції соціального маркетингу.

Фаза 12: Звітування, вимірювання та аналітика – поточні

- 1) встановлення Google Analytics;
- 2) (SEO-клієнти) SEO MOZ та звітування про Page Rank;
- 3) повний звіт про помилки сайту та виправлення;
- 4) відгуки веб-майстрів;
- 5) представлення сайту в пошукових системах;
- 6) підвищення безпеки запитаних форм;
- 7) повний огляд сайту та оцінка наступної фази.

Фаза 13: Вміст та блоги – поточні

- 1) стратегія контенту;
- 2) створення вмісту.

Фаза 14: Off Page SEO – постійно;

- 1) створення стратегії зворотного зв'язку;
- 2) рада будівництва влади;
- 3) створення рейтингу для вибраних ключових фраз;
- 4) вимірювання;

5) звітність.

Фаза 15: Маркетинг соціальних мереж – постійний

- 1) маркетинговий календар;
- 2) стратегія;
- 3) створення вмісту;
- 4) графіка Creation;
- 5) просування;
- 6) маркетинг;
- 7) звітність.

Додаткові функції, які потребують додаткового часу, Інвестиції та ресурси, які потрібно додати, включені, але не обмежуються:

- 1) зміни дизайну на етапі розробки;
- 2) електронна комерція;
- 3) система членства;
- 4) система каталогів;
- 5) система лістингу;
- 6) користувацьке кодування;
- 7) додаткові спеціально розроблені та розроблені сторінки;
- 8) користувацький дизайн та розробка цільових сторінок;
- 9) системи реєстрації;
- 10) системи планування;
- 11) додаткова безпека;
- 12) величезні веб-сайти;
- 13) медіа / Веб-сайти новин з функціями розширеної категорії;
- 14) список бажань Функціональність;
- 15) передачі складних змісту та редагування вмісту;
- 16) передача вмісту з високим вмістом та очищення;
- 17) тривалий час для перегляду відгуків, переглядів та схвалень;
- 18) агресивні плани SEO[13].

ВИСНОВКИ

Експерти вважають, що електронна торгівля – ключовий напрямок економічного розвитку в 21 столітті. Це, безсумнівно, вірно і стосовно України, де протягом декількох наступних років продовжиться стрімке зростання електронної торгівлі. В даний час ми спостерігаємо стрімке проникнення інтернету не тільки в міста-мільйонники, а й в малорозвинуті регіони України. Очевидним є те що через декілька років в електронну торгівлю буде залучено більшу кількість користувачів, і кожен п'ятий буде щотижня здійснювати покупки онлайн. Ті підприємства, що раніше інших створять свої представництва в інтернеті, отримають перевагу над тими, хто працює суто в традиційній формі.

Для впевнених дій в сьогоденні, необхідно знати минуле: передумови зародження електронної комерції, а також ключові моменти її розвитку. Даного поняття трохи більше 50 років, але швидкість розвитку даного напрямку дійсно вражає. Безумовно, інтернет в цілому і інтернет-торгівля, зокрема, не могли залишитися без уваги з боку держави. Всі передові країни розробляють законодавство в сфері інтернет-торгівлі, Україна не виняток. У роботі ми дали опис сучасного процесу регулювання даної сфери і позначили основні проблеми, які мають місце бути.

Також в першому розділі були розглянуті переваги і недоліки інтернет-торгівлі і дана характеристика загальним принципам побудови інтернет-магазину.

Були систематизовані знання про найбільш поширених і доступних на даний момент видах оплати і службах доставки, які є сполучними ланками між покупцем і онлайн-магазином.

Найбільш розвинений ринок електронної комерції в США і Європі. Саме тому ми приділили більше уваги зарубіжного сектору інтернет-торгівлі і виявили ключові фактори успіху побудови інтернет-магазину на прикладі гіганта Amazon.com. Безумовно, що українська дійсність буде розвиватися по кордонному образу і подобою, з невеликими поправками на менталітет і досвід наших співвітчизників, і є сенс вже сьогодні аналізувати і впроваджувати іноземну практику. У першому розділі також були виявлені основні проблеми української інтернет-торгівлі та запропоновано шляхи їх вирішення. На жаль, частина освітлених моментів не залежить від власників інтернет-бізнесу і не може бути вирішена тільки їх зусиллями – потрібне обов'язкове втручання держави.

Отримані в ході досліджень і роботи з теоретичним матеріалом знання

були закріплені в розробці та наладженні інтернет-магазину.

Всі поставлені завдання були вирішені, і мета дипломної роботи досягнута.

ПЕРЕЛІК ПОСИЛАНЬ

1. Мэтт Зандстра. PHP: об'єкти, шаблони и методики програмування: Пер. с англ. – СПб.: Диалектика-Вильямс, 2018. – 576 с.
2. Ашманов И., Иванов А. Оптимизация и продвижение сайтов в поисковых системах. – СПб.: Питер, 2009. – 400 с.
3. Джош Локхарт. Сучасний PHP: нові можливості та хороші практики: Пер. с англ. – СПб.: Символ-Плюс, 2016. – 270с.
4. Luke Welling, Laura Thomson. PHP and MySQL Web Development, 5th edition. 2016. – 688с.
5. Крис Питт. Про PHP MVC: Пер. с англ. – СПб.: Питер, 2013. – 500 с.
6. WordPress [Електронний ресурс] – Режим доступу:
<https://ru.wikipedia.org/wiki/WordPress>
7. Джесс Чедвик ASP.NET MVC 4: разработка реальных веб-приложений с помощью ASP.NET MVC – М.: «Вильямс», 2013. – 432 с.
8. Фаулер М., Скотт К. UML. Основы: Пер. с англ. – СПб.: Символ-Плюс, 2002. – 192 с.
9. Стефан Спенсер, Джиммі Хардінг, Дженніфер Шехан. Соціальна електронна комерція: збільшення продажів та розширення торгової марки: Пер. с англ. – СПб.: Питер, 2015. – 310 с.
10. Раскрутка. Секреты эффективного продвижения сайтов / А. Бабаев, Н. Евдокимов, М. Боде і ін. – СПб.: Питер, 2013. – 272 с.
11. Брайан Клифтон. Google Analytics для профессионалов. – М.: «Диалектика», 2013. – 608 с.
12. SEMBOOK. Энциклопедия поискового продвижения сайтов Ingate / Н. Неелова, Г. Загребельный, Е. Фролкина. – Москва.: Питер, 2014. – 520 с.
13. Лапыгин Ю. Н. Управление проектами: от планирования до оценки эффективности. – М.: Омега-Л, 2008. – 252 с.

Д О Д А Т К И

ДОДАТОК А

КОД YOAСТ SEO ДЛЯ SOCIAL MARKUP

```

<meta property="og:locale" content="en_US" />
<meta property="og:type" content="article" />
<meta property="og:title" content="How to Install WordPress Locally the Easy
Way" />
<meta property="og:description" content="Sometimes it can be more convenient
to work with WordPress on your own machine. Check out how to install
WordPress locally the easy way with DesktopServer." />
<meta property="og:url" content="https://lemarbet/blog/install-wordpress-locally/"
/>
<meta property="og:site_name" content="Lemarbet Managed WordPress Hosting"
/>
<meta property="article:publisher"
content="https://www.facebook.com/kinstahosting" />
<meta property="article:section" content="Development" />
<meta property="article:published_time" content="2017-11-03T03:15:35-08:00"
/>
<meta property="article:modified_time" content="2017-11-06T18:55:53-08:00" />
<meta property="og:updated_time" content="2017-11-06T18:55:53-08:00" />
<meta property="fb:admins" content="1407415661" />
<meta property="og:image" content="https://lemarbet/wp-
content/uploads/2016/11/install-wordpress-locally.jpg" />
<meta name="twitter:card" content="summary_large_image" />
<meta name="twitter:description" content="Sometimes it can be more convenient
to work with WordPress on your own machine. Check out how to install
WordPress locally the easy way with DesktopServer." />
<meta name="twitter:title" content="How to Install WordPress Locally the Easy
Way" />
<meta name="twitter:site" content="@lemarbet" />
<meta name="twitter:image" content="https://lemarbet/wp-
content/uploads/2016/11/install-wordpress-locally.jpg" />
<meta name="twitter:creator" content="" />

```

ДОДАТОК Б

UML ДІАГРАМА БАЗИ ДАНИХ WORDPRESS

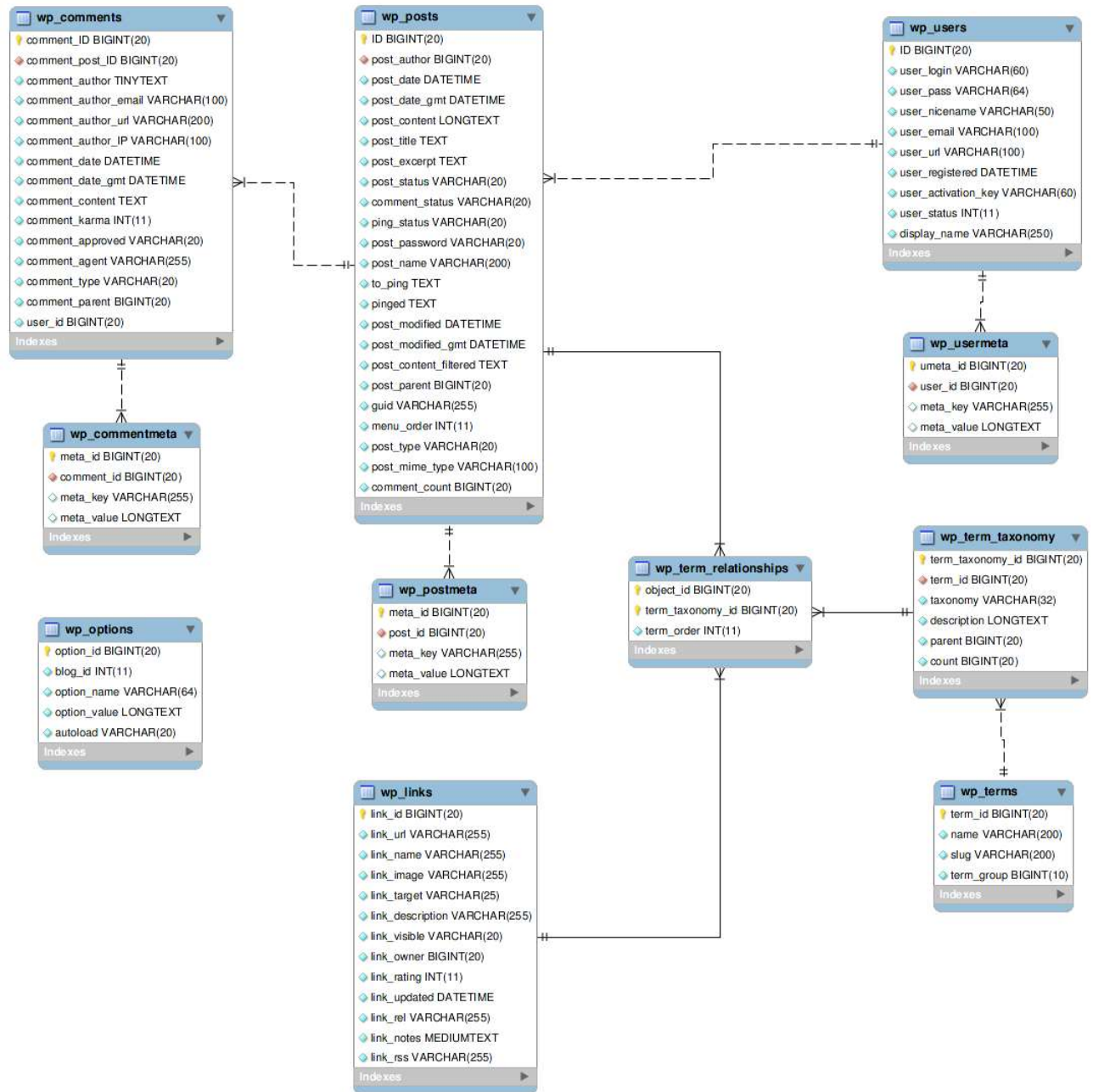


Рисунок Б.1 – діаграма бази даних wordpress