МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,

управління та адміністрування

Кафедра інформаційних технологій

Бакалаврська кваліфікаційна робота

на тему: Розробка інформаційної системи Каховського заводу залізобетонних виробів

Виконав студент 4 курсу групи

К-42 Напрямок 6.05.01.01 комп'ютерні науки,

Саєнко Андрій Миколайович

_____

Керівник к.т.н., доцент

Трегубова Ірина Анатоліївна

Рецензент к.ф-м.н.

Вітавецка ЛарисаАнатоліївна

Одеса 2019

Вступ 5
1 Аналіз предметної області 8
1.1 Обґрунтування потреби в веб-сайту 8
1.2 Поняття інформаційної системи та веб –сайта 8
1.3 Класифікація веб-сайтів 9
1.4 Засоби розробки 10
1.4.1 Мова розмітки гіпертекстових документів HTML 10
1.4.2 Динамічна мова розмітки гіпертекстових документів DHTML 12
1.4.3 Розширювана мова розмітки XML 13
1.4.4 Мова програмування PHP 16
2 Аналіз вибору системи для виконання завдання та основні етапи розробки веб-сайту 17
2.1 Вибір системи для розробки веб-сайту 17
2.1.1 Обґрунтування вибору системи управління вмістом 18
2.2 Аналіз доступного ПЗ 18
2.2.1 Аналіз варіантів ОС 18
2.2.2 Аналіз варіантів ПЗ для веб-серверів 19
2.2.3 Аналіз додаткового ПО 20
2.3 Аналіз готових рішень 21
2.3.1 Аналіз фронт-енд фреймворків 21
2.3.2 Аналіз CMS 21
2.4 Основні етапи виконнання завдання 29
2.4.1 Етапи розробки веб-сайта 29
2.5 Проектування веб-сайта 33
2.5.1 Розробка інтерфейсу 34
2.5.2 Розробка адміністративної частини сайту 34
3 Проектна частина 36
3.1 Загальна структура сайта 36
3.2 Дизайн веб-сайту 37
3.3 Опис створення сторінок веб-сайта 42
3.4 Розробка сайту на основі CMS 43

3.4.1 Опис встановлення на сервер CMS Wordpress 43
3.4.2 Створення сторінок сайту 44
3.4.3 Установка шаблону 48
3.4.4 Встановлення плагінів 50
3.4 Створення інтернет-магазину 53
3.4.1 Створення рубрики для інтернет магазину 53
3.4.2 Створення товарів інтернет-магазину 54
3.4.3 Створення корзини 55
3.4.4 Створення сторінки “Мы” 56
Висновки 58
Перелік джерел посилання 59
Додаток А Лістинг програми 59
А.1 Код шаблону сайту 61
А.2 Код файлу header.php 68
А.3 Код файлу footer.php 70
А.4 Код файлу basket-tamplate.php 70
А.5 Код файлу front-page.php 71

ВСТУП

Актуальність теми**.** Web-сайт – це своєрідний інтерфейс між підприємством та його оточенням - партнерами, постачальниками, клієнтами. Тому створення сайта є одним з головних завдань підприємницької діяльності, в тому числі і в мережі Internet.

Нині все більша кількість компаній починає використовувати Internet для обміну інформацією та оптимізації роботи з клієнтами. Найпоширенішою формою присутності у мережі є створення сайта (Web-представництва).

Структурована інформація сайта дає змогу партнерам і клієнтам отримати повне уявлення про підприємство і його діяльність.

Споживач одержує з рекламних банерів сайта і комерційних та ділових видань багато корисних Web-адрес, з яких можна дізнатися ціни, замовити товари.

Такий спосіб передачі повідомлень потенційним клієнтам ефективніший за звичайні засоби комунікації: пошту, телефон, факс. За допомогою таких обов'язкових елементів, як інтерфейс або навігація, клієнт може самостійно вибрати необхідну для вивчення інформацію. Сайт є доповненням до рекламної кампанії, забезпечуючи можливість зворотного зв'язку й інтерактивної роботи з даними підприємства.

Метою роботи є створення інформаційної системи для підприємства “Каховський завод залізобетонних виробів”. Другорядною метою є набуття та вдосконалення навичок створення та керування інформаційними системами. Отже можна виділити такі задачі:

– розроблення базових принципів подання Каховського завода залізобетонних виробів у WWW;
– визначення основних принципів побудови веб-сайта та окремих його компонент;
– розроблення базової структури інформаційного наповнення веб-сайта та принципів його категоризації;

– вироблення підходів щодо організації колективу розробників і адміністраторів та підвищення ефективності роботи з інформаційним наповненням веб-сайта;

– розроблення політики інформаційної безпеки веб-сайта.

Предметом дослідження є теоретичні, методологічні та прикладні аспекти у створенні комерційних веб-сайтів з вікористанням нових технологій та стандартів W3C (Консорціум Всесвітньої павутини).

Методами дослідженняє: аналітичний, статистичний, метод порівняння, систематизації даних та програмний.

Враховуючи поставлені вище завдання, я вирішив, що найбільш оптимальним засобом для розробки такого роду електронної комерції – є система управління вмістом Wordpress.

Wordpress – система керування змістом з відкритим кодом, яка через свою простоту в установленні та використанні широко застосовується для створення веб-сайтів. Сфера використання – від блогів до складних веб-сайтів. Вбудована система тем і плагінів у поєднанні з вдалою архітектурою дозволяє конструювати на основі WordPress практично будь-які веб-проекти.

Аналіз останніх досліджень і публікацій. Сьогодні проблема створення ефективного веб-сайта постає перед усіма без винятку вищими навчальними закладами. Проте дослідники і практики у цій галузі не є консолідованими, чим і пояснюється брак якісних наукових публікацій з цієї тематики. Попри те, що університет та академічне середовище самі по собі володіють низкою важливих особливостей порівняно з іншими типами організацій та спільнот, подання їх у WWW часто не розглядається у наукових колах як окрема проблема. Розробники сайтів університетів об’єднуються переважно для обміну конкретною практичною інформацією (прикладом є спільнота UniversityWebDevelopers – http://cuwebd.ning.com)[1][1)] .

[1)] [1] University web developers URL : http://cuwebd.ning.com (дата звернення 03.04.2019).

Об'єкт дослідження– процес подання підприємства у мережі Internet

Предмет дослідження – теоретичні методологічні ,програмні , прикладні та практичні аспекти створення та керування інофрмаційних систем.

Практична значущістьотриманих результатів полягає у тому, що отримані висновки сприятимуть подальшому розвитку особистих якостей та навичок для подальшої роботи у сфері створення веб-сайтів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Обґрунтування потреби в веб-сайту

На сьогоднішній день практично кожна організація має власний веб-сайт. В умовах використання сучасних інформаційних технологій – це необхідний чинник існування, що дозволяє розширити поле рекламної діяльності і привернути тим самим додаткових клієнтів.

- створення і розробка сайтів включає:
- затвердження первинного технічного завдання на розробку сайта.
- визначення структурної схеми сайту – розташування розділів, контента і навігації.
- веб-дизайн – створення графічних елементів макету сайту, стилів і елементів навігації.

1.2 Поняття інформаційної системи та веб –сайта

Інформаці́йна систе́ма – сукупність методів організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів.

Інформація, доступна користувачам Internet, розташовується на комп'ютерах (веб-серверах), на яких встановлено спеціальне програмне

забезпечення. Значна частина цієї інформації організована у вигляді веб-сайтів. Кожен з них має своє ім'я (адреса) в Internet.

Веб-сайт – це інформація, представлена в певному вигляді, яка розташовується на веб-сервері і має своє ім'я (адреса). Для перегляду веб-сайтів на комп'ютері користувача використовуються спеціальні програми, які називаються браузерами. Залежно від того, яке ім'я (адреса) сайту ми задамо в рядку "Адреса", браузер завантажуватиме в своє вікно відповідну інформацію.

Веб-сайт складається із зв'язаних між собою веб-сторінок. Веб-сторінка є текстовим файлом з розширенням *.htm, який містить текстову інформацію і спеціальні команди – HTML-коди, що визначають в якому вигляді ця інформація відображатиметься у вікні браузера. Кожна сторінка веб-сайту також має свій Internet адрес, який складається з адреси сайту і імені файлу, відповідного даній сторінці. Таким чином, веб-сайт – це інформаційний ресурс, що складається із зв'язаних між собою гіпертекстових документів (веб-сторінок), розміщений на веб-сервері і такий, що має індивідуальну адресу.

1.3 Класифікація веб-сайтів

В даний час у всесвітній павутині розміщено декілька мільйонів веб-сайтів і їх число постійно росте. Це особисті сторінки, що містять інформацію про автора, його інтереси. Їх створюють для того, щоб знайти друзів по інтересах, розширити свій кругозір, свій світ.

Інформаційні сайти. До них відносяться сайти учбових закладів, співтовариств по інтересах, фірм і інше.

Сайти-портали. До них відносяться крупні веб-сайти, які пропонують велику кількість сервісів, починаючи від каталога зображень до форумів і поштових скриньок.

Інтернет-магазини. Такі сайти в наш час набирають дуже велику популярність з боку, як користувачів, так і з боку підприємців. Це веб-сайти, в яких кожна людина може купити будь-який товар не відходячи від свого місця біля комп’ютера.

Соціальні сайти. Це веб-сайти на яких кожна людина має змогу спілкуватися зі старими друзями та знаходити нових.

1.4 Засоби розробки

1.4.1 Мова розмітки гіпертекстових документів HTML

Обмін інформацією в Інтернет здійснюється за допомогою протоколів прикладного рівня, що реалізують той або інший прикладний сервіс (пересилку файлів, гіпертекстової інформації, пошта і так далі). Одним з найбільш молодих і популярних сервісів Інтернет, розвиток якого і привело до сплеску популярності самої Інтернет, стала WorldWideWeb (WWW), заснована на протоколі HTTP (HyperTextTransferProtocol– протокол передачі гіпертекстовій інформації). Гіпертекстові документи, представлені в WWW, мають одну принципову відмінність від традиційних гіпертекстових документів – зв'язки, в них що використовуються, не обмежені одним документом, і більш того, не обмежені одним комп'ютером. Для підготовки гіпертекстових документів використовується мова HTML (HyperTextMarkupLanguage – мова розмітки гіпертекстових документів), що надає широкі можливості по форматуванню і структурній розмітці документів, організації зв'язків між різними документами, засоби включення графічної і мультимедійної інформації. HTML-документи є видимими за допомогою спеціальної програми – браузера. Найбільшого поширення в даний час набули браузери MozillaFirefox і Internet Explorer компанії Microsoft (MSIE). Реалізації MozillaFirefox доступні практично для всіх сучасних програмних і апаратних платформ, реалізації MSIE доступні для всіх Windows платформ, Macintosh і деяких комерційних Unix-систем.

HTML-документ складається з тексту, що є змістом документа, і тег і в , що визначають його структуру і зовнішній вигляд при відображенні браузером. Простий html-документ виглядає таким чином:

```
<html >
<head >
<title>Назва</title>
</head>
<body >
<p>Тілодокумента
</body>
</html>
```

Як видно з прикладу, тег є ключовим словом, поміщеним в кутові дужки. Розрізняють одинарні теги, як, наприклад, <p > , і парні, як <body></body>, в останньому випадку дія тега розповсюджується тільки на текст між його відкриваючою і закриваючою дужкою. Теги також можуть мати параметри – наприклад, при описі сторінки можна задати колір фону, колір шрифту і т.д.: <Bodybgcolor="white" text="black">.

Текст всього документа полягає в теги <html > , сам документ розбивається на дві частини – заголовок і тіло. Заголовок описується тегами <head>, в яких можуть бути включені назва документа (за допомогою тегов<title>) і інші параметри, що використовуються браузером при відображенні документа. Тіло документа поміщене в теги <body> і містить власне інформацію, яку бачить користувач. За відсутності тегів форматування весь текст виводиться у вікно браузера суцільним потоком, переклади рядків, пропуски і табуляції розглядаються як пробільні символи, декілька пробільних символів, що йдуть підряд, замінюються на один. Для форматування використовуються наступні основні теги:

<p> – початок нового абзацу, може мати параметр, що визначає вирівнювання:

<p align=right>;

 – переклад рядка в межах поточного абзацу;

<u></u> – виділення тексту підкресленням

Посилання на інший документ встановлюється за допомогою тега... , де URL – повна або відносна адреса документа. При цьому текст, ув'язнений в тег <a>, зазвичай виділяється підкресленням і кольором, і після натискання мишею по цьому посиланню браузер відкриває документ, адреса якого вказана в параметрі href. Графічні зображення вставляються в документ за допомогою тега<imgsrc="URL">.

1.4.2 Динамічна мова розмітки гіпертекстових документів DHTML

Динамічний HTML (Dynamic HTML, DHTML) не є якоюсь особливою мовою розмітки сторінок. Це всього лише термін, вживаний для позначень HTML-сторінок з динамічно змінним вмістом.

Реалізація DHTML “стоїть” на трьох “китах”: безпосередньо HTML, каскадні таблиці стилів і мови сценаріїв. Ці три компоненти DHTML зв'язано між собою об'єктною моделлю документа (DOM, DocumentObjectModel), що є по суті інтерфейсом прикладного програмування (API). DOM зв'язує воєдино три перераховані компоненти, додаючи простому документу HTML нову якість – можливість динамічної зміни свого вмісту без перевантаження сторінки.

Об'єктна модель документа робить всі елементи сторінки програмованими об'єктами. З її допомогою через мови сценаріїв можна дістати доступ і управляти всім, що є в документі. Кожен елемент HTML доступний як індивідуальний об'єкт, а це означає, що можна змінювати значення будь-якого параметра будь-якого тега HTML-сторінки, і, як наслідок, документ дійсно стає динамічним. Будь-яка дія користувача (натискання кнопкою миші, переміщення миші у вікні браузера або натиснення клавіші клавіатури) об'єктною моделлю документа трактується як подія, яка може бути перехоплене і оброблене процедурою сценарію.

1.4.3 Розширювана мова розмітки XML

На сьогоднішній день вже всім фахівцям в області веб-технологий стало очевидне, що існуючих стандартів передачі даних по Internet недостатньо. Формат HTML, ставши свого часу проривом в області відображення вмісту вузлів Internet, вже не задовольняє всім необхідним на даний момент вимогам. Він дозволяє описати те, яким чином повинні бути відображені дані на екрані кінцевого користувача, але не надає ніяких засобів для ефективного опису передаваних даних і управління ними.

Крім того, каменем спотикання для багатьох компаній, що займаються розробкою програмного забезпечення, є необхідність сумісного використання різних компонентів, забезпечення їх взаємодії, можливості обміну даними між ними.

До недавнього часу не існувало стандарту, що надає засоби для інтелектуального пошуку інформації, обміну даними, адаптивної обробки отримуваних даних.

Вирішенням всіх описаних вище проблем став затверджений в 1998 році міжнародною організацією, мова XML. XML (eXtensibleMarkupLanguage) – це розширювана мова розмітки, призначена для опису в текстовій формі структурованих даних. Цей текстовий (text-based) формат, багато в чому схожий з HTML, розроблений спеціально для зберігання і передачі даних.XML дозволяє описувати і передавати такі структуровані дані, як:

- окремі документи.
- метадані, що описують вміст якого-небудь вузла Internet.
- об'єкти, що містять дані і методи роботи з ними (наприклад, елементи управління ACTIVEX або об'єкти Java).
- окремі записи (наприклад, результати виконання запитів до баз даних)

– всілякі веб-посилання на інформаційні і людські ресурси Internet (адреси електронної пошти, гіпертекстові посилання і ін.)

Дані, описані на мові XML, називаються XML-документами. Мова XML легко читана і достатньо проста для розуміння. Якщо Ви були знайомі з HTML, то навчитися складати XML-документи не складе для Вас ніяких труднощів[2][1)].

Початковий текст XML-документа складається з набору XML-элементів, кожен з яких містить початковий і кінцевий теги. Кожна пара тегів представляє частину даних. Тобто, як і HTML, мова XML для опису даних використовує теги. Але, на відміну від HTML, XML дозволяє використовувати необмежений набір пар тегів, кожна з яких представляє не то, як увязнені в неї дані повинні виглядати, а то, що вони означають.

Будь-який елемент XML-документа може мати атрибути, характеристики, що уточнюють його. Атрибут – це пара ім'я = "значення", яка задається при визначенні елементу в початковому тегу.

Принцип розширюваності мови XML полягає в можливості використання необмеженої кількості пар тегів, визначуваних творцем XML-документа.

Принцип незалежності визначення внутрішньої структури документа від способів представлення цієї інформації полягає у відділенні даних від процесу їх обробки і відображення. Таким чином, отримані дані можна використовувати відповідно до потреб клієнта, тобто вибирати потрібне оформлення, застосовувати необхідні методи обробки.

Управляти відображенням елементів у вікні програми-клієнта (наприклад, у вікні браузера) можна за допомогою спеціальних інструкцій – стильових таблиць XSL (eXstensibleStylesheetLanguage). Ці таблиці XSL дозволяють визначати оформлення елементу залежно від його місцерозташування усередині документа, тобто до двох елементів з

[1)][2] Основы XML для начинающих пользователей.URL:https://www.ibm.com/developerworks/ru/library/x-newxml/index.html (дата звернення 03.04.2019)

однаковою назвою можуть застосовуватися різні правила форматування. Крім того, мовою, лежачою в основі XSL, є XML, а це означає, що таблиці XSL більш універсальні, а для контролю коректності складання таких стильових таблиць можна використовувати DTD-описания або схеми даних, розглянуті нижче.

Формат XML, в порівнянні з HTML, має невеликий набір простих правил розбору, який дозволяє розбирати XML-документи, не удаючись до яких-небудь зовнішніх описів використовуваних XML-елементів. У загальному випадку XML-документи повинні задовольняти наступним вимогам:

– кожен відкриваючий тег, що визначає деяку частину даних в документі, обов'язково повинен супроводжуватися таким, що закриває, тобто, на відміну від HTML, не можна опускати закриваючі теги.
– вкладеність тегів в XML строго контролюється, тому необхідно стежити за порядком проходження відкриваючих і закриваючих тегів.
– у XML враховується регістр символів.
– вся інформація, розташована між початковим і кінцевим тегами, розглядається в XML як дані, і тому враховуються всі символи форматування (тобто пропуски, переклади рядків, табуляції не ігноруються, як в HTML).

У XML існує набір зарезервованих символів, які повинні бути задані в XML-документи тільки спеціальним чином. Багато фахівців розглядають XML як нову технологію інтеграції програмних компонентів. Основними перевагами використання XML є:

– інтеграція даних з різних джерел. XML можна використовувати для об'єднання різнорідних структурованих даних на середньому рівні трирівневих веб-систем, баз даних.

– локальна обробка даних. Отримані дані у форматі XML можна розбирати, обробляти і відображати безпосередньо на клієнтові без додаткових звернень до сервера.
– перегляд і маніпулювання даними в різних розрізах. Отримані дані можуть оброблятися і бути видимими клієнтом різними способами залежно від потреб кінцевого користувача.
– можливість часткового оновлення даних. За допомогою XML можна оновлювати тільки ту частину структурованих даних, яка була змінена, а не всю структуру цілком.

Всі ці переваги роблять XML незамінним інструментом для розробки гнучких засобів пошуку інформації в базах даних, могутніх трирівневих веб-додатків, а також додатків, що підтримують транзакції. Іншими словами, за допомогою XML можна формувати запити до баз даних різних структур, що дозволяє здійснювати пошук інформації в численних несумісних один з одним базах даних.

Використання XML на середньому рівні трирівневих веб-додатків дозволяє здійснювати ефективний обмін даними між клієнтами і серверами систем електронної комерції.

Крім того, мова XML може використовуватися як засіб для опису граматики інших мов і контролю правильності складання документів.

Інструменти обробки даних, отриманих у форматі XML, можуть бути розроблені в середовищі VisualBasic, Java або C++.

1.4.4 Мова програмування PHP

Історія PHP починається осінню 1994 року. Коли РасмусЛердорф (RasmusLerdorf) почав працювати над тим, що згодом стало PHP, єдиною метою, яка була у нього в думках, з'ясувати, хто читає його резюме.

Щоб стежити за відвідувачами, він створив CGI скрипт на Perl-і, який вставлявся як спеціальний тег в HTML код його сторінки, і збирав інформацію про відвідувачів.

Щоб справити враження на потенційних працедавців, він дозволив будь-якому відвідувачеві сторінки проглядати збирану статистику відвідин [3][1)].

Він назвав цей код – ToolsforPersonalHomePage, оскільки сам використовував його на своїй персональній домашній сторінці (personalhomepage). Декілька чоловік поцікавилися тим, як вони могли б отримати цей інструмент, і Лердорф ухвалив рішення надати його іншим особам. «Це диво програмного забезпечення. Ви можете дати це і проте залишити це собі», – дотепно відмітив Лердорф. У той час руху OpenSource не існувало. Тоді воно назвалося freeware.

2 АНАЛІЗ ВИБОРУ СИСТЕМИ ДЛЯ ВИКОНАННЯ ЗАВДАННЯ ТА ОСНОВНІ ЕТАПИ РОЗРОБКИ ВЕБ-САЙТУ

2.1 Вибір системи для розробки веб-сайту

Для виконання поставленої задачі , було обрано систему керуванням вмістом з відкритим кодом – Wordpress.

Основні переваги даної системи це :

- підтримка веб-стандартів (XHTML,CMS);
- модулі для підключення (плагіни) з унікально простою системою їх взаємодії з кодом; можливість автоматичного встановлення та оновлення версії безпосередньо з панелі адміністратора;
- підтримка так званих «тем», з допомогою яких легко змінюється як зовнішній вигляд, так і способи виведення даних;
- можливість редагувати шаблони одразу в панелі адміністратора;

[1)][3] Матеріал із офіційного сайту php.net. URL: https://www.php.net/manual/ru/book.mysql.php (дата звернення 03.04.2019)

– багато бібліотек «тем» і «плагінів»;

2.1.1 Обґрунтування вибору системи управління вмістом

Необхідна розвинена гіпертекстова структура – HTML створений саме для цього; необхідно забезпечити гнучку структуру управління матеріалом – управляти HTML-вмістом зручно за допомогою PHP+MySQL. Саме ця зв'язка дозволяє генерувати динамічні HTML-сторінки; необхідно реалізувати варіативність представлення матеріалу – в цьому випадку використання HTML+CSS більш ніж обгрунтовано.

Саме ця комбінація засобів дозволяє створювати мультимедіа документи; необхідно забезпечити підтримку внутрішнього магазину веб-сайта. [4][1)]

На підставі поставленого технічного завдання була вибрана система управління вмістом CMS Wordpress, оскільки саме ця система відповідає всім вимогам.

2.2 Аналіз доступного ПЗ

2.2.1 Аналіз варіантів ОС

Хостинг на якому буде знаходитися сайт працює на ОС „Debian“ , тож для розробки раціонально вибрати найбільш наближену ОС.

„Debian“ – операційна система, що складається з вільного ПЗ з відкритим вихідним кодом. В даний час Debian GNU / Linux– один з найпопулярніших і важливих систем GNU / Linux , в первинній формі зробив значний вплив на розвиток цього типу ОС в цілому .

[1)][4]w3schoolsHTMLTutorials.Інформація про вивчення HTML. URL: https://www.w3schools.com/html/ (дата звернення (02.04.2019).

Debian може використовуватися в якості операційної системи як для серверів, так і для робочих станцій .

Debian став основою цілого ряду дистрибутивів .

Найвідоміші з них (в алфавітному порядку) –antiX, Knoppix, LinuxMint, Maemo, SteamOS, TAILS, Ubuntu.[5][1)]

Оскільки обидві ОС є похідними від „Debian“ вони є досить схожими і одним з основних відмінностей є інтерфейс з яким вони поставляються . У „Mint“ це „Cinamon“ а у „Ubuntu“ – „Unity“ . „Unity “ – досить ресурсоємний інтерфейс в той час як „Cinamon“ не використовує багато ресурсів і є досить наближеним до інтерфейсу „windows“ який є звичним для більшосты користувачів персональних комп’ютерів.

2.2.2 Аналіз варіантів ПЗ для веб-серверів

Серед найбільш популярних веб-серверів можна відилітиApache та nGinx.

Apache HTTP Server був розроблений Робертом Маккула в 1995 році, а з 1999 року розробляється під управлінням ApacheSoftwareFoundation– фонду розвитку програмного забезпечення Apache. Так як HTTP сервер це перший і самий популярний проект фонду його зазвичай називають просто Apache.

Веб-сервер Apache був найпопулярнішим веб-сервером в інтернеті з 1996 року. Завдяки його популярності у Apache сильна документація ,інтеграція зі стороннім софтом.

Адміністратори часто вибирають Apache через його гнучкості і широку поширеність. Він може бути розширений за допомогою системи динамічнозавантажуваних модулів і виконувати програми на великій

[1)] [5] Матеріал з Вікипедії вільній енциклопедії про Debian. URL:https://ru.wikipedia .org/wiki/Debian(дата звернення 02.04.2019).

кількості різних мов програмування без використання зовнішнього програмного забезпечення.

У 2002 році Ігор Сисоєв почав роботу над Nginx для того щоб вирішити проблему C10K – вимога до ПЗ працювати з 10 тисячами одночасних з'єднань. Перший публічний реліз був випущений в 2004 році, поставлена мета була досягнута завдяки асинхронної event-driven архітектурі.

Nginx почав набирати популярність з моменту релізу завдяки своїй легковажності (light-weight resource utilization масштабуватися на мінімальній залозі. Nginx чудовий при віддачі статичного контенту і спроектований так, щоб передавати динамічні запити іншому ПЗ призначеному для їх обробки.

Однак тому що на хостингу використовуються веб-сервер раціонально використовувати його .

2.2.3 Аналіз додаткового ПО

Під додатковим ПО розуміється використання IDE або текстового редактора. Оскільки на сьогодні є досить розвинені текстові редактори які задовольняють основні потреби при розробці простих сайтів то використання ресурсоємних IDE не раціонально. Серед текстових редакторів можливо виділити Atom , SublimeText та NotePad++.

Atom є чудовим інструментом, тому що дозволяє налаштувати свій редактор , присутня досить детальна документецію, підсвічування синтаксису, велика кількість плагінів та інтеграція з GitHub. Найбільшим недоліком є проблеми з продуктивністю .

SublimeText текстовий редактор який має підсвічування синтаксису, велику кількість плагінів та приємний дизайн. Недоліком є те що він є умовно безкоштовний.

Notepad++ є досить швидким текстовим редактором однак він є застарілим та не має такої бази плагінів та доповнень як його конкуренти .

2.3 Аналіз готових рішень

2.3.1 Аналіз фронт-енд фреймворків

Серед фронт-энд фреймворків виділяються Bootstap, Skeleton, Semantic UI та Slides. Bootstrap один з найвідоміших CSS-фреймворків на сьогоднішній день. Має в своєму складі шаблони для відтворення кнопок, сайдбарі, навігаційних панелей, форм та інших елементів сайту. Включає себе JavaScript-розширення.

Основні інструменти bootstrap– шаблони, @media, форми, навігація, АЛЕРТ, типографіка і звичайно ж, сітки. Bootstrap сумісний з усіма основними сучасними браузерами, але в старих версіях браузерів можуть бути проблеми. Підтримує адаптивність. Використовує мови Less і Sass.

Skeleton заснований на JavaScript і CSS, використовується для швидкого і безпомилкового створення адаптивних сайтів які коректнозображаються як на великих моніторах, так і на екранах маленьких гаджетів. Великий плюс Skeleton – його можна пристосувати під будь-який дизайн. Сітка розмітки містить 12 колонок і має базову ширину 960 пікселів. Сумісний з усіма основними браузерами.

Semantic UI маєпідтримкаFirefox, Chrome, Safari, Internet Explorer, Android 4, Blackberry10. Містить велику кількість компонентів: іконки, зображення, написи, меню та інші стандартні компоненти. Підтримує останні версії HTML і CSS і має хорошу добірку скриптових модулів і API. Використовує Less. Slides дуже потужний інструмент для розробки клієнтської частини сайти, має набір готових шаблонів, готових рішень для сучасного веб-програмування на кшталт форм, діалогових вікон, слайдерів, детальну документацію, набір анімацій , є кросс-браузерним. Він в достатній мірі охоплює всі потреби при розробці проекту і єдиним мінусом є те що він платний. Оскільки в компанії мені оплатили його , то він є чудовим вибором .

2.3.2 Аналіз CMS

Так як вибраною мовою програмування став PHP , він за свою історію обзавівся досить великою кількістю готових рішень відфреймворків на кшталт laravel і Symthony до CMS.

Оскільки більшість задач поставлених перед нами є типовими то вигідніше використовувати CMS . Перелік CMS написаних на PHP є досіть великим , виберемо основні з них це Wordpress, Opencart, 1C-bitrix, Joomla та Drupal.

WordPress – система керування вмістом сайту з відкритим вихідним кодом; написана на PHP; сервер бази даних –MySQL; випущена під ліцензією GNU GPL версії 2. Сфера застосування – від блогів до досить складних новинних ресурсів і інтернет-магазинів. Вбудована система «тем» і «плагінів» разом з вдалою архітектурою дозволяє конструювати проекти широкої функціональної складності [6][1)].

E-commerceplatform «OpenCart» була написана в 1998 році Крістофером Манном для WalnutCreek CDROM. Перший публічний реліз відбувся 11 травня 1999 року. Розроблений на мові Perl, спочатку проект розвивався слабо і остаточно був покинутий в 2000 році, коли Манн заявив, що він більше не може розвивати OpenCart, так як у нього є інші зобов'язання.

Друге життя система знайшла завдяки британському розробнику Деніелу Керр, який використовував напрацювання Манна для створення свого власного движка на PHP. Перший реліз оновленого OpenCart відбувся 10 лютого 2009 року –Керр виклав свою систему на GoogleCode під індексом 1.1.1

У вересні 2014 року OpenCart став найпопулярнішим рішенням для інтернет-комерції в Китаї, а за станом на серпень 2015 року на OpenCart працювало 6,42% всіх інтернет-магазинів світу. За цим показником OpenCart

---

[1)][6] Матеріал з Вікипедії — вільній енциклопедії про системи управління сайтом. http://ru.wikipedia.org/wiki/CMS (дата звернення 03.04.2019).

став третім в світі, відставши тільки від WooCommerce і Magento і випередивши OSCommerce, ZenCart і Shopify.

На офіційному сайті вказується, що після установки програмного забезпечення необхідно тільки додати продукти і, при необхідності, замінити шаблон оформлення сайту; корзина товарів окремої настройки не вимагає і працює відразу . В адміністративній панелі є можливість управляти замовленнями і доступно управління декількома платіжними сервісами .

Як переваги програмного забезпечення офіційний сайт відзначає наступні пункти :

- відкритість вихідного коду.
- документованість ПО.
- необмеженість категорій, продуктів і їх виробників.
- необмеженість інформаційних сторінок.
- підтримка багатомовності та перекладу інтерфейсу.
- можливість встановлювати власні теми.

Також, в OpenCart є ряд недоліків:

- складно додавати нові функції для різних версій. Процес установки і видалення модулів організований не зовсім зручно. При установці завжди потрібно стежити за сумісністю модуля з версією движка.
- проблеми з оновленнями. Після зміни версії, виходять з ладу деякі плагіни, що негативно позначається на роботі всього сайту.
- періодично візуальний редактор самостійно створює зайвий код, який ускладнює роботу IT-фахівцям, ускладнює і погіршує індексацію сайту пошуковими системами.
- незручно влаштована функція завантаження картинок. Немає можливості завантажувати картинки пакетом по кілька штук, сортування зображень відбувається тільки в довільному порядку, пошук по зображеннях досить проблематичний, що ускладнює завдання заміни фото до картки товару, якщо на сайті їх багато.

– відсутня можливість виставити автоматичне прописування ЧПУ. У даній CMS ЧПУ прописуються вручну кожен раз після редагування вже доданих товарів або «заливання» нових позиції.
– виникають проблеми при зміні шаблону. Оскільки OpenCart має досить складну структуру коду, зміна шаблону може викликати збої в роботі окремих модулів і всього сайту в цілому.
– сторінка товару закривається після натискання клавіші збереження і якщо потрібно внести ще якісь правки доводиться знову шукати потрібний товар і відкривати його.
– занадто багато зайвих полів при оформленні замовлення. Причому, дані поля розбиті на спойлери. І змінити це, не коригуючи саму систему на технічному рівні, просто не можна.

Розробник може доповнити стандартний код такими рішеннями, які в повній мірі нівелюють вплив всіх негативних факторів. Таким чином почали з'являтися збірки на основі OpenCart. Основна мета таких збірок - спрощення взаємодії з CMS на стороні адміністратора і клієнта. Для цього розробники покращують функціонал вже готового движка, додають різні модулі, працюють над дизайном.

1С-Бітрікс – CMS (система управління контентом веб-проекту) від російської компанії «1С-Бітрікс».

10 лютого 2003 вийшла «Бітрікс: Управління сайтом 3.0» – перша комерційно успішна версія CMS від «Бітрікс» .

«Більшість використовувало« самописний »софт, а ми запропонували універсальне рішення з управління контентом на сайті (CMS)»– засновник компанії Сергій Рижиков в інтерв'ю ForbesRussia №03 (168) 2018

За допомогою CMS будь-який користувач міг створити інтернет-магазин, корпоративний сайт або новинний портал.

Система управління вмістом Joomla! є відгалуженням широко відомої CMS Mambo. Команда незалежних розробників відокремилася від проекту

Mambo через незгоду в економічній політиці. 16 вересня 2005 року в світ вийшла перша версія Joomla !, що є по суті перейменованої Mambo 4.5.2.3 і включає в себе виправлення знайдених на той момент помилок і вразливостей[7][1)].

До літа 2008 року по числу щоденних завантажень Joomla! посіла друге місце після WordPress зі значним відривом від інших подібних систем.

Версія 1.0 вважається застарілою, її офіційна підтримка припинена 1 липня 2009 року.

Підтримка версії 1.6 припинено 19 серпня 2011 року.

Підтримка версії 1.7 припинено 24 лютого 2012 року.

Підтримка версії 1.5 припинено 27 вересня 2012 року.

Підтримка версії .5 тривала до кінця 2014 року.

Поточна версія системи - 3.9.6, випуск якої відбувся 7 травня 2019 року.

CMS Joomla! включає в себе мінімальний набір інструментів при початковій установці, який доповнюється в міру необхідності. Це знижує захаращення адміністративної панелі непотрібними елементами, а також знижує навантаження на сервер і економить місце на хостингу.

Joomla! дозволяє відображати інтерфейс фронтальної та адміністративної частини будь-якою мовою. Каталог розширень містить безліч мовних пакетів, які встановлюються штатними засобами адміністрування. Доступні пакети російського, українського, білоруського та ще деяких мов пострадянського простору.

Функціональність можна збільшувати за допомогою додаткових розширень (компонентів, модулів і плагінів).

Є модуль безпеки для багаторівневої аутентифікації користувачів та адміністраторів (використовується власний алгоритм аутентифікації і «ведення» сесій).

[1)][7]Веб-сайт joomla.ru.ИсторияJoomla. URL: http://joomla.ru/about/history(дата звернення 03.04.2019)

Система шаблонів дозволяє легко змінювати зовнішній вигляд сайту: розташування модулів, шрифти і інше. Можна надати користувачам вибирати одне з декількох відображень. У мережі існує величезний вибір готових шаблонів, як платних, так і безкоштовних. Також існує програмне забезпечення для самостійного створення оригінальних шаблонів.

Передбачені настроюються схеми розташування модулів, включаючи лівий, правий, центральний і будь-яке інше довільне положення блоку. При бажанні вміст модуля можна включити в вміст матеріалу. Наприклад, вираз {loadpositionmod_fpslideshow}, введене (разом з фігурними дужками) в довільне місце в статті, виведе вміст модуля, якому задана позиція виведення як «mod_fpslideshow».

До переваг системи можна віднести те, що всі компоненти, модулі, плагіни і шаблони можна написати самому, розмістити їх в структурованому каталозі розширень або відредагувати існуюче розширення на свій розсуд.

Відбувається регулярний вихід оновлень. Існує публічний «баг-трекер» (система відслідковування помилок). (Див. Список офіційних трекерів.) Існують також трекери міграції зі старих версій Joomla !, трекер побажань розширення функціональності і так далі, де користувачі Joomla! можуть залишати зауваження з приводу роботи CMS, які згодом вивчаються її розробниками, при необхідності включають в чергове оновлення Joomla! виправлення, вирішальні ті чи інші проблеми.

Починаючи з версії 1.6 вбудована багатомовність.

Починаючи з версії 2.5 розширена підтримка баз даних. Реалізована підтримка Microsoft SQL Server, а з версії 3.0 –PostgreSQL . Надалі планується додати підтримку Oracle, SQLite.

Структурно система являє собою ядро, до якого приєднуються компоненти. Компоненти можуть складатися по функціональному призначенню з модуля, плагіна і / або шаблону:

модулі – елементи, які відображаються на сторінці користувачів у вигляді окремих блоків.

плагіни – елементи, які відображаються на адміністративній панелі інструментів і надають додаткову функціональність.

шаблони – елемент, який відповідає за уніфіковане оформлення всього сайту.

В ядро Joomla! входить:

– блок (функціональність визначається HTML-кодом всередині нього) і його різновиди з розширеними можливостями управління з адміністративної панелі: банери (модулі для розміщення реклами), контакти (блок для розміщення контактної інформації та форми зворотного зв'язку), стрічка новин (RSS підписка);
– матеріал (стаття) з мітками (ключові слова) і категоріями (папками) для навігації користувачів, редактором для зручного оформлення матеріалу.

Чи не входять в ядро вільні сторонні розширення:

– плагін для резервного копіювання (backup) і відновлення, наприклад, AkeebaBackup.

Створення власного розширення здійснюється за допомогою написання плагіна на мові PHP з використанням для взаємодії API Joomla !

Drupal – система управління вмістом (CMS), яка використовується також як каркас для веб-додатків (CMF), написана на мові PHP і використовує як сховище даних реляційну базу даних (підтримуються MySQL, PostgreSQL та інші). Drupal є вільним програмним забезпеченням, захищеним ліцензією GPL, і розвивається зусиллями ентузіастів зі всього світу.

Почав розробку 2000 року бельгієць ДрісБёйтарт (DriesBuytaert), який і понині є керівником проекту. Назва drupal є спотвореним вимовою голландського слова druppel (крапля) і з'явилося в результаті забавною помилки. У 2000 році кілька студентів Антверпенского університету використовували загальний ADSL-модем для доступу в інтернет, і для спілкування користувачів цієї маленької локальної мережі Дріс створив внутрішній сайт. Програма, на якій працював сайт, не мала імені, поки Дріс

не зібрався реєструвати доменне ім'я для нового сайту, і в результаті помилки замість dorp.org (від голландського dorp– село, цілком доречне ім'я для невеликого співтовариства) був зареєстрований drop.org (англ. drop– крапля). Аудиторія нового сайту з часом змінилася на які цікавляться розвитком нових веб-технологій, і новий сайт поступово перейшов з любительського експерименту в проект з обговорень ідей веб-програмування. У 2001 році Дріс опублікував вихідний код drop.org під ім'ям «Drupal». З тих пір розробка перемістилася на зареєстрований пізніше сайт drupal.org, який набув статусу офіційного сайту проекту.

З часу створення в процес розробки були залучені сотні людей. Офіційні розробники ядра зараз – це десятки людей, але багато людей надсилають свої патчі, що включаються в код системи. Також функціональність движка розширюється додатковими модулями, які створюють сторонні розробники. Наприклад, в релізі 4.6 взяли участь 50 осіб, що надіслали 338 поправок до коду, а в релізі 4.7 взяли участь вже 338 розробників, що надіслали приблизно 1500 поправок. За п'ять років існування проекту (до 2006 року) вийшло 13 релізів. На даний момент (січень 2011) було розроблено понад 7000 додаткових модулів, що розширюють базові функції CMS. За статистикою звернення до оновлень движка, на березень 2009 року в інтернеті – в межах 120 тис. Активних сайтів на Drupal 6 (без урахування сайтів, що знаходяться за файрволлами, сайтів з відключеним механізмом оновлень). Це також не враховує сайтів, що працюють на ранніх версіях Drupal.

У 2007 році ДрісБёйтарт створив фірму Acquia, спрямовану на комерційну підтримку проектів і сервісів, заснованих на Drupal, натхненний успіхами Canonical і RedHat, що побудували бізнес на підтримку відкритих проектів. У 2008 Бёйтарт отримав докторський ступінь в галузі інформатики та приладобудування («ComputerScienceandEngineering»).

Згодом навколо міжнародного сайту з'явилися національні спільноти, функціонують понад 20 національних сайтів підтримки на різних мовах.

Архітектура Drupal дозволяє застосовувати його для побудови різних типів сайтів – від блогів і новинних сайтів до інформаційних архівів або соціальних мереж.

Критики Drupal ставлять в докір розробникам слабке використання об'єктних можливостей PHP. Дійсно, API Drupal практично не використовує наявні в PHP можливості ООП. Розробники аргументують це слабкою реалізацією ООП в мові (особливо до версії PHP 5). Об'єктна модель в Drupal присутній, але в дещо нетрадиційному для PHP вигляді. В Drupal 7 проте присутній ряд нововведень, що задіюють об'єктні можливості PHP 5. А у версії Drupal 8 офіційно прийнято використання ООП. Однак, в співтоваристві присутні вельми суперечливі думки з даного приводу.

До недоліків (але одночасно і до достоїнств) Drupal можна віднести відсутність зворотної сумісності API. У кожному новому великому релізі відбуваються великі зміни API, коли поряд з додаванням нових функцій прибираються деякі старі або змінюються параметри виклику існуючих. Це призводить до необхідності розробникам сторонніх модулів адаптувати їх для роботи з новими версіями Drupal. Однак зміни API і процедура адаптації модулів до нових версій описуються в документації для кожного релізу, також завжди пропонується механізм автоматизованого апгрейда ядра системи на нову версію. Плюс даної схеми розробки– немає необхідності тягти з версії у версію програмний шар сумісності зі старими API, що полегшує поточний код системи.

2.4 Основні етапи виконнання завдання

2.4.1 Етапи розробки веб-сайта

Послідовність створення сайту і чітка опрацювання етапів – запорука успіху всього проекту. Можливо, не відає користувачеві може здатися, що розробка сайту – нікчемна справа. Ну що тут складного – придумав дизайн,

зверстав, наповнив його парою-трійкою текстів і готово. Насправді ж кожен етап створення сайту – це копітка робота, за яку відповідальний кожен розробник з команди веб-студії. Створення веб-сайту містить в собі не тільки розробку дизайну і програмінг, як ви могли подумати, а й детальний аналіз проекту, співпраця з замовником і пошук рішень для досягнення поставлених цілей проекту. У цій статті ми розглянемо як відбувається поетапна розробка сайту і детально розпишемо порядок створення веб-сайтів.

Основні етапи створення веб-сайту:

– постановка цілей і завдань сайту;
– створення, опрацювання технічного завдання (ТЗ) на розробку сайту;
– прототипування;
– створення макета дизайну сайту;
– верстка;
– програмування;
– наповнення контентом;
– тестування;
– задача готового проекту клієнтові.

Далеко не кожен клієнт нашої веб-студії бачить відмінності в видах веб-сайту, здатний самостійно визначити його цілі і завдання. Він звертається до нас з проханням створити проект. Ми ж, у свою чергу, проводимо подготавітальний етап – аналізуємо для чого клієнтові сайт, які завдання він повинен вирішувати: виконувати функцію представництва компанії в Інтернеті або ж продавати товар і послуги, стати якісною рекламою компанії або бути стартапом для молодої фірми. Все це обговорюється з замовником. Він повинен сам розуміти навіщо йому сайт, в чому він допоможе його компанії і з чим впоратися не зможе.Після того, як поставлені завдання, визначається цільова аудиторія. Важливо зрозуміти для кого створюється ресурс, що замовник може запропонувати і як відреагує аудиторія на оффер. Визначення ЦА "диктує" свої правила, на основі отриманих даних

формується структура сайту, контент, дизайн, функціонал та ін.Далі проводиться маркетинговий аналіз, до якого включено аналіз конкурентів, сезонності товару або послуги, діяльності та ніші, конкурентоспроможності пропозиції. Визначаються зачіпки і фішки, які допоможуть привернути увагу відвідувачів. По завершенню розробляється прототип сайту і демонструється клієнту.Наступний етап розробки сайту – формування технічного завдання. Технічне завдання – це документ, який складається разом з клієнтом. Він включає детальні характеристики ресурсу:

- побажання по дизайну (кольорове оформлення, застосування фірмового стилю або його розробка, співвідношення графічних елементів до текстових);
- семантичне ядро;
- структуру сайту, кількість сторінок, категорій, блоків;
- функціонал сайту (тільки стандартний або розробка додаткових модулів, докладний їх опис та цілі);
- застосовувані технології;
- технічні вимоги до ресурсу.

Технічне завдання – це основа, на яку покладається кожен фахівець, який бере участь в розробці. Тому в складанні ТЗ участь замовника – необхідно. Воно обумовлюється і редагується до тих пір, поки клієнт дасть згоду і підпише документ. Тільки після цього проект переходить в стадію створення. На етапі формування і обговорення ТЗ також розробляється структура сайту, його навігація, кількість категорій і підкатегорій, їх послідовність розміщення і т.д. Вся інформація також включається в документ, який підписується замовником.Всі етапи створення сайту важливі для якісного функціонування ресурсу, але розробка дизайну – один з головних. Адже дизайн – це те, що бачить відвідувач в першу чергу, оцінює його і приймає рішення залишитися на сторінці або закрити вкладку браузера.Дизайнер грунтується на технічному завданні, малює кнопки,

банери та інші графічні елементи. Іншими словами, той прототип, який був створений на першому етапі розробки сайту, отримує естетичний зовнішній вигляд, проводиться в кольорах, обраними замовником. Якщо компанія має корпоративний стиль, то дизайн розробляється в його відповідності. Або ж може спочатку розробити фірмовий стиль, а після – дизайн на його основі.Важливо відзначити, що дизайнер малює дизайн не кожної сторінки, а шаблони кількох основних, використовуючи тенденції веб-дизайну. Готовий макет передається клієнту і очікує свого схвалення. Якщо необхідні доопрацювання, дизайнер виконує їх і знову показує макет замовнику. Дизайн допрацьовується доти, поки він не буде затверджений.Наступний етап – злиття дизайну з системою управління сайту (верстка)Верстальник за допомогою мови HTML переводить готовий дизайн в робочий проект. Ресурс отримує життя, стає динамічним, все кнопки працюють. Ресурс стає кросбраузерності і правильно відображається у всіх існуючих інтернет-браузерах. На це ж етапі створюються стилі CSS. Далі ресурс підключається до системи управління. З її допомогою створюються сторінки, категорії і підкатегорії проекту, визначається порядок роботи вкладки меню, розставляються гіперпосилання. Ресурс отримує остаточну структуру, яка буде представлена відвідувачам. Іноді сайти можуть використовуватися без CMS (наприклад, Лендінзі), тому сайт просто верстається на HTML і CSS мовами без прікруткі до движку.На цьому ж етапі верстаються категорії і підкатегорії, контент, який буде розміщений на сторінці. Велика увага верстальник приділяє верстці картки товару для інтернет-магазину, яка повинна відповідати технічним і призначеним для користувача вимогам.На завершальному етапі верстки розробляється адаптивна версія сайту. Вона не обов'язкова, але рекомендована клієнтам, які бажають охопити мобільний трафік.На етапах створення простого сайту (наприклад, одностранічнік) в програмуванні немає необхідності. Програміст підключається до роботи, якщо розробляється великий і складний проект. Завданням фахівця коштує розробка складних функцій: калькулятора, системи оплати за допомогою

банківських карт, конвертер валюти, тобто всіх тих функцій, які не входять в стандартний пакет функціоналу.Передостанній етап створення веб-сайту – наповнення сторінок графічним та інформаційним контентом. Розміщується відео, фото, тексти та інша інформація, яку зможе побачити або прочитати відвідувач.На основі семантичного ядра пишуться SEO-статті (якщо передбачається просування ресурсу СЕО-методом), контент-менеджер розміщує графічні елементи в логічній структурі. Сторінки проходять внутрішню оптимізацію.Найважливішу роль в процесі створення сайту грає його тестінг. Виконує роботу тестувальник, який моніторить функціональність ресурсу по ряду критеріїв і виявляє помилки, які повинні бути усунені.

Ми не винесли вибір домену та хостингу, як окремий етап. Він необхідний, але може проводитися в будь-який момент процесу створення ресурсу. По завершенню роботи готовий проект передається замовнику разом з логінами і паролями до системи управління. Після здачі проекту клієнт може замовити послугу просування і підтримки проекту, сплативши їх вартість. Рідко після того, як паролі та доступ до сайту видані замовнику, співпраця з виконавцем закінчується. Найчастіше клієнти замовляють додаткові послуги, наприклад:

- оновлення каталогу, матеріалів, викладка товарів і їх опис;
- навчання роботі з сайтом;
- підтримка сайту;
- просування ресурсу;
- написання статей, наповнення категорій СЕО-текстами і ін.

Важливо відзначити, що вся поетапне створення сайту контролюється замовником. По закінченню кожного етапу готовий результат демонструється клієнту і узгоджується з ним.

.

2.5 Проектування веб-сайта

Проектування і розробка сайтів включає:

– затвердження первинного технічного завдання розробки сайту.
– визначення структурної схеми сайту – розташування розділів, контента і навігації.
– веб-дизайн – створення графічних елементів макету сайту, стилів і елементів навігації.
– розробка програмного коду, модулів, бази даних і інших елементів сайту необхідних в проекті.
– тестування і розміщення сайту в мережі Інтернет.

2.5.1 Розробка інтерфейсу

Титульна сторінка (головна) будь-якого сайту повинна максимально інформативно і в стислому об'ємі відображати необхідну користувачеві інформацію про сайт. На головній сторінці необхідно помістити логотип веб-сайту, основне меню сайту (для навігації по його структурі), форму аутентифікації (входу зареєстрованих користувачів), реєстраційне посилання (реєстрація нових клієнтів).

2.5.2 Розробка адміністративної частини сайту

Адміністративна частина повинна включати: інформацію про нових користувачів, розділ настройок сайту, можливість адміністрування магазину, можливість виведення статистики по магазину.Адміністративна частина сайту представлена у вигляді системи управління вмістом.

Система управління вмістом (англ. Contentmanagementsystem, CMS) – комп'ютерна програма, використовувана для управління вмістом чого-небудь (звичайний цей вміст розглядається як неструктуровані дані наочного

завдання в протилежність структурованим даним, СУБД, що зазвичай знаходяться під управлінням). Зазвичай такі системи використовуються для зберігання і публікації великої кількості документів, зображень, музики або відео.

Окремим випадком такого роду систем є системи управління сайтами. Подібні CMS дозволяє управляти текстовим і графічним наповненням веб-сайту, надаючи користувачеві зручні інструменти зберігання і публікації інформації.

Зараз існує безліч готових систем управління вмістом сайту, у тому числі і безкоштовних. Їх можна розділити на три типи, за способом роботи:

Генерація сторінок при редагуванні . Системи цього типу програми для редагування сторінок, які при внесенні змін до змісту сайту створюють набір статичних сторінок. При такому способі жертвуется інтерактивність між відвідувачем і вмістом сайту.

Змішаний тип. Як зрозуміло з назви, поєднує в собі переваги перших два. Може бути реалізований шляхом кешування — модуль уявлення генерує сторінку один раз, надалі вона в рази швидше підвантажується з кеша. Кеш може оновлюватися як автоматично, по закінченню деякого терміну часу або при внесенні змін до певних розділів сайту, так і вручну по команді адміністратора. Інший підхід – збереження певних інформаційних блоків на етапі редагування сайту і збірка сторінки з цих блоків при запиті відповідної сторінки користувачем.

На мою думку ,дуже важливим фактором є відкритість та гнучкість програмного забезпечення , тому Wordpress є фаворитом серед подібних собі. Основною перевагою CMS WordPress є максимально швидке ядро системи управління контентом WordPress, що дозволяє підключати до нього будь-яку кількість плагінів (модулів) для додаткових функцій.

Завдяки вбудованому візуальному редактору, який за всіма ознаками схожий за зручністю роботи з Microsoft Word, будь-яка задача з форматування тексту, вставки зображень чи відео, здійснюється всього в кілька кліків

мишею, доступ до якого відбувається через звичайний браузер. І відповідно, немає необхідності у закачуванні сторінки на сайт за допомогою спеціальної програми – FTP клієнта. Надалі можна легко змінювати існуючі сторінки і додавати нові.

При побудові сторінки за допомогою редактора WordPress, ви бачите її в тому ж самому вигляді, в якому вона буде представлена на сайті. Можна додавати малюнки і форматувати заголовки, при цьому бачити це все відразу ж у вікні редактора.ДляWordPress написано сотні плагінів, більшість з них безкоштовні і доступні для усіх. З їх допомогою можна досягнути дійсно вражаючих функціональних можливостей.

Також вражаючою є популярність даної системи серед користувачів інтернет-форумів. Це означає , що вирішення майбутніх проблем буде швидким та ефективним завдяки активному товариству.

Отже, на підставі поставленого технічного завдання була ,вибрана система управління вмістом CMS Wordpress, оскільки саме ця система відповідає всім вимогам технічного завдання, і є найбільш гнучким та відкритим інструментом для створення та керування веб-сайтів.

3 ПРОЕКТНА ЧАСТИНА

3.1 Загальна структура сайта

На рис. 3.1 представлена структурна схема сайта, яка враховує всю специфіку веб-сайта, що розробляється.

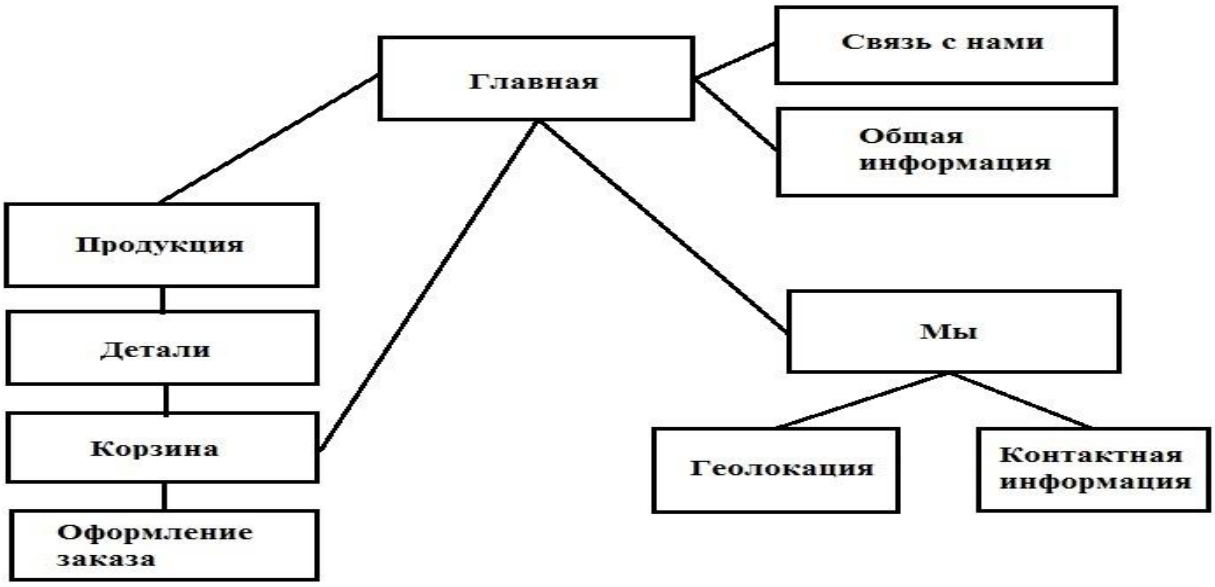

Рисунок 3.1–Структурна схема сайту

На головній сторінці відображається основна інформація про підприємство, контактна форма для звя’зку, контактна інформація, та перехід на інші сторінки сайту. Сторінка «Продукция» містить товари, які є в магазині.

Сторінка «Детали» містить повну інформацію про товар та його ціну.Також є можливість додати товар до корзини.

Сторінка «Корзина» містить всю необхідну покупцям інформацію про оплату товара та її доставку.Сторінка «Мы» містить повну контактну інформацію, та геолокаціюпидпрємства.

3.2 Дизайн веб-сайту

Відповідно до розробленої структури була спроектована головна сторінка сайту (рис. 3.2). Вона містить всі основні структурні елементи, перехід по яких здійснюється за допомогою гіперпосилань.

Рисунок 3.2 – Головна сторінка сайту

Як і планувалося на етапі постановки завдання, сайт містить всі необхідні структурні элементи:можливість перейти до контактної форми,сторінкипродукціі , корзини та інформаційної сторінки .При натисненні на посиланні «Продукция» виводиться первинна інформація про товари. Це набагато спрощує пошук продукції для клієнтів (рис.3.3).

Рисунок 3.3–Сторінка “Продукция”

При натисненні посилання “Детали” ми переходимо на сторінку товару, на якій ми маємо можливість познайомитися з характеристиками товару та додати його у корзину (рис. 3.4).

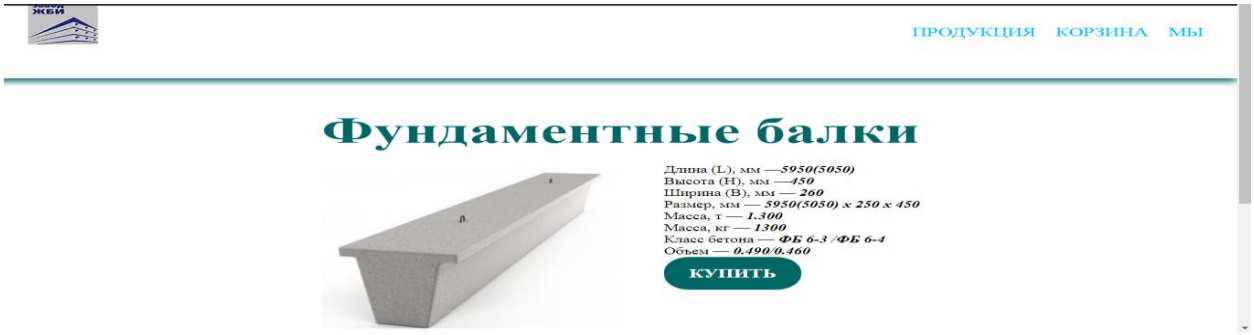

Рисунок 3.4– Сторінка “Деталі товару”

Отже, якщо ми натиснимо на посилання “Купить” , ми перейдемо до сторінки “Корзина” , на якій ми зможемо регулювати нашими ,доданими до корзини товарами.За допомогою кнопки “Пересчитать”, ми маємо можливість дізнатися суму усіх доданих товарів.На цій сторінці присутня функція оформлення замовлення за допомогою кнопки “Оформить заказ” (рис. 3.5).

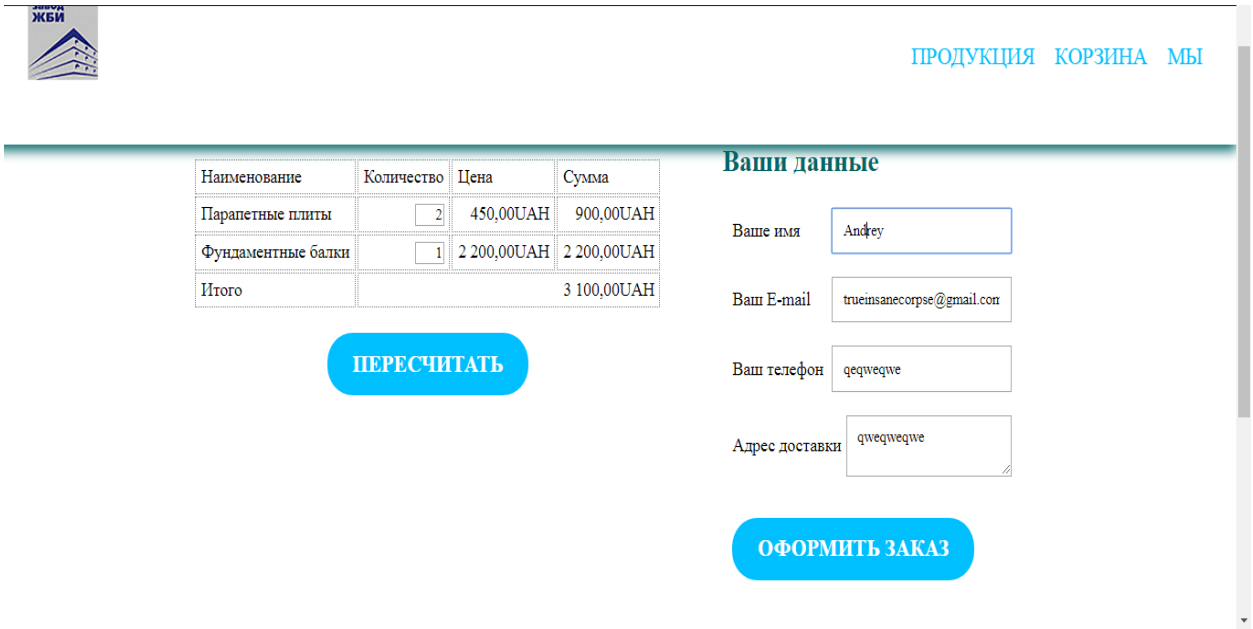

Рисунок 3.5–Сторінка “Корзина”

При натисненні на кнопку “Мы”, користувач переходить до сторінки на якій встановленний плагін геолокаціі, за допомогою якого ми можемо визначити точне містоположення підприємства використовуючи “ Google Мап и”.(рис. 3.6).

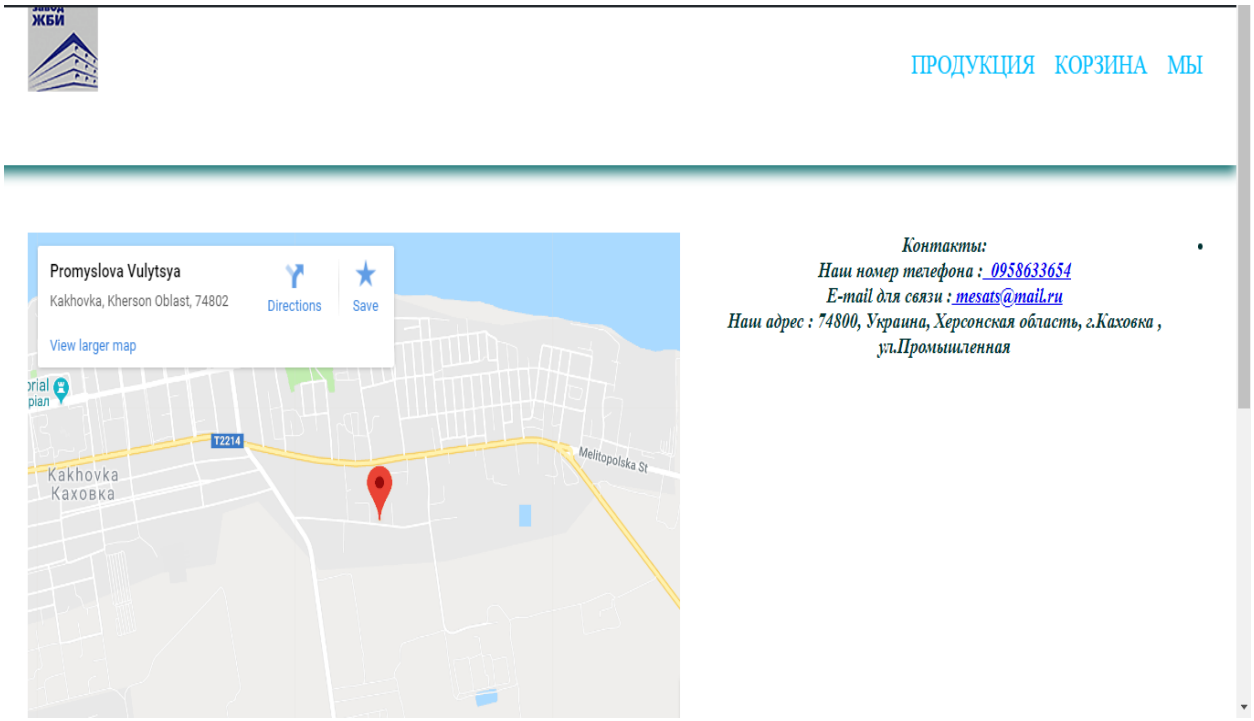

Рисунок 3.6–Сторінка “Мы”

При натисненні посилання “Связаться”, нас автоматично переносить до кінця головної сторінки , де розташована контактна форма для зв’язку із менеджером підприємства (рис. 3.7).

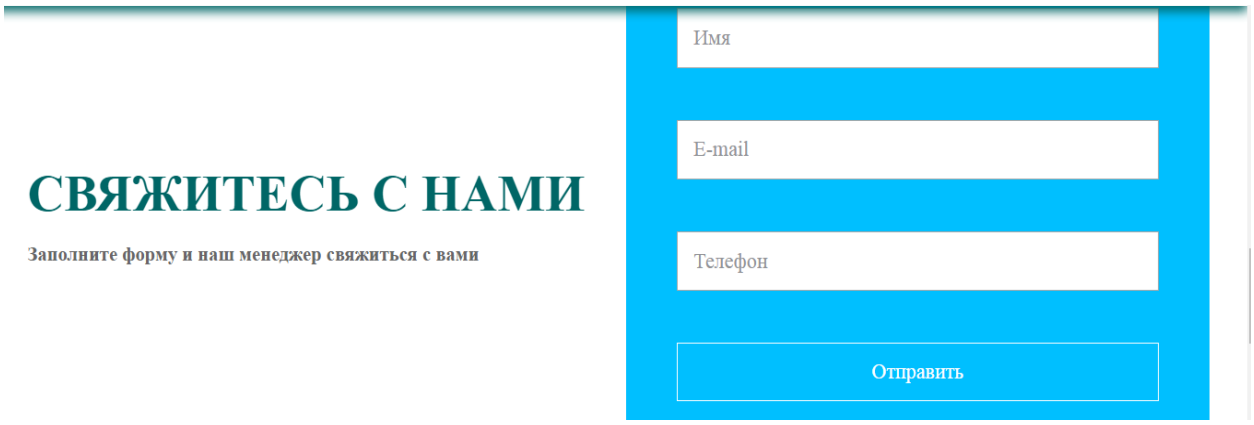

Рисунок 3.7–Контактна форма

3.3 Опис створення сторінок веб-сайта

Розробка веб-сайта велася в основному в програмном пакеті компанії Wordpressверії 4.9.10.

Вибір даного пакету обґрунтований тим, що його освоєння віднімає мало часу, під час розробки доступний попередній перегляд кожної окремо зробленої операції, доступна велика кількість функцій (організація інтерактивних елементів сайту, створення гіперпосилань в декілька натисків мишки і тому подібне), пакет підтримує велику кількість технологій (HTML, PHP, ASP,Java, XML, XSLT, CSS та інші).

При створенні сайту використовувалися стандартні шаблони, проте для їх модифікації під конкретні завдання, стандартні засоби Joomla застосовувати незручно через відсутність візуального редактора.

Отже,проаналізувавши предметну область,та перевіривши доступні аналоги я зупинився на тому щоби вибрати такі застосування для редагування шаблонів ,текстовий режакторNotepad ++ та інтернет-браузер GoogleChrome (рис. 3.8).

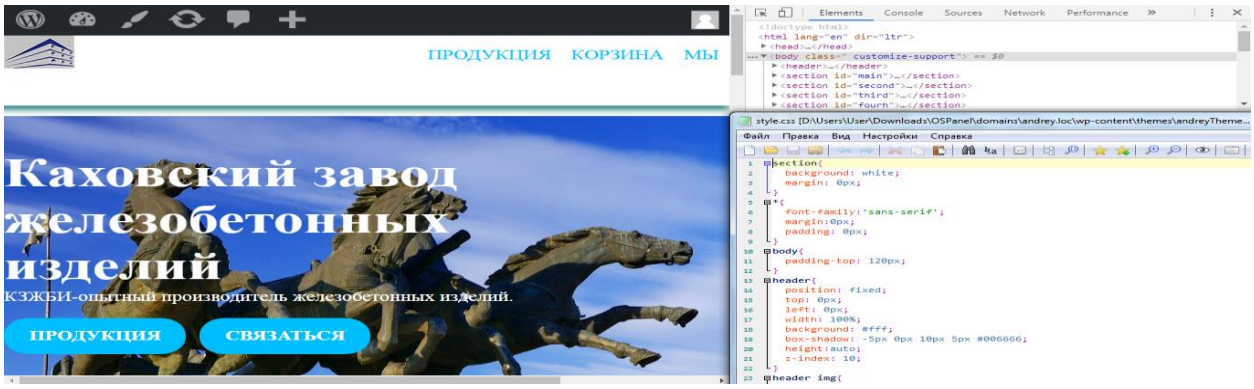

Рисунок 3.8–Розробка головної сторінки сайту

Будь-яка інформація (текстова або графічна) заноситься безпосередньо у текстові файли с розшиереннім .php. Таким чином, досягається структурованість сторіночки, що дозволяє найзручніше редагувати будь-який її елемент.

3.4 Розробка сайту на основі CMS

3.4.1 Опис встановлення на сервер CMS Wordpress

Для доступу до адміністративної панелі управлінняWordpress використовуйте адресу Вашого веб-сайту (або повний шлях до теки, в яку встановлена Wordpress) з додаванням в кінці шляху "/wp-admin". Наприклад, якщо адреса веб-сайта www.myweb.ua, то доступ до панелі управління можливий за адресою www.myweb.ua/wp-admin. Після правильного введення адреси, відкриється сторінка авторизації (рис. 3.9)[8][1)].

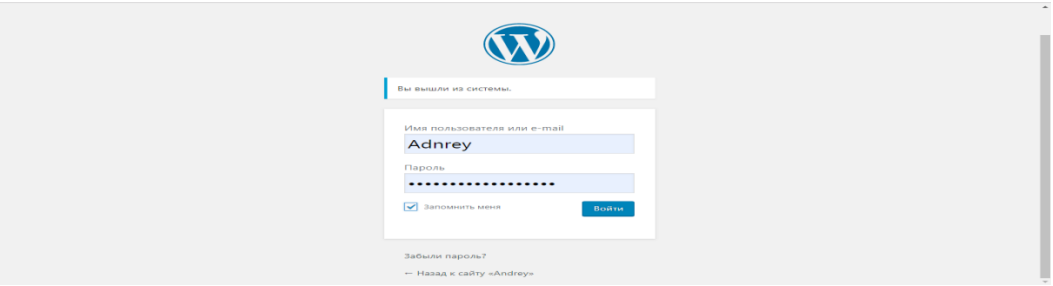

Рисунок 3.9–Сторінка входу в панель управління Wordpress

[1)] [8] Опис керування панеллю адміна.URL:https://wpkama.ru/cat/wordpress/adminka-wordpress.(дата звернення 03.04.2019).

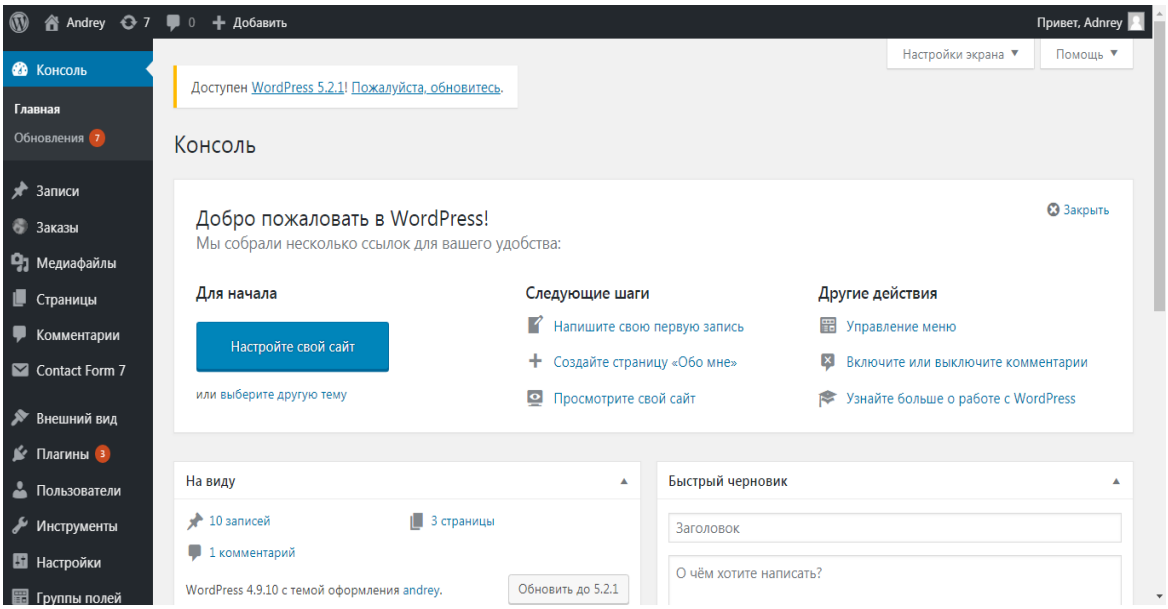

Рисунок 3.10–Головна сторінка панелі управління Wordpress

При правильному введенні імені користувача і пароля відкриється головна сторінка адміністрування системою, тобто буде відкрита панель управління Wordpress (рис. 3.10), яка надасть управління всіма функціями і можливостями Wordpress.

3.4.2 Створення сторінок сайту

Сторінка – це головний об'єкт в основі ієрархії структури вмісту. Для роботи з сторінками необхідно перейти у розділ "Страницы". Можна натиснути на головній сторінці панелі кнопку "Страницы" або у верхньому меню вибрати розділ “Добавить” і в наданому виборі , натиснути на кнопку “Страницы” (рис. 3.11).

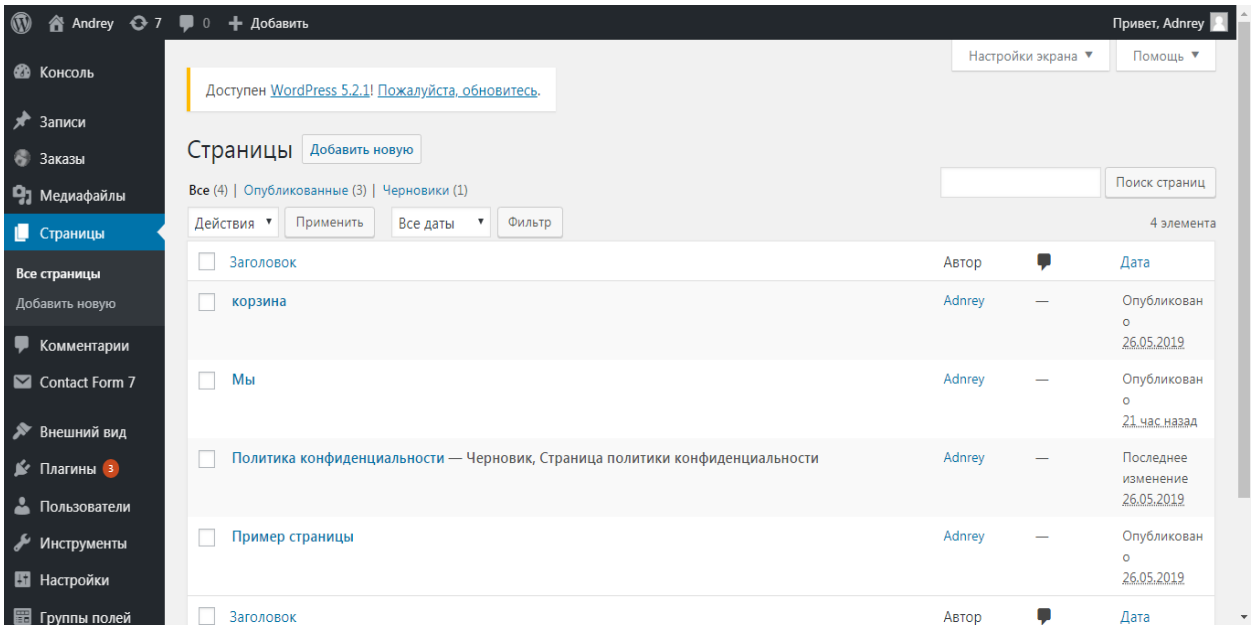

Рисунок 3.11–Сторінка управління розділом “Страницы”

На сторінці "Страницы", використовуючи кнопки на панелі інструментів, можна:

– за домогою форми “Поискстраниц” виконати пошук бажаної сторінки серед створених.
– створити нову сторінку. Натисніть кнопку "Добавить новую" у лівому діалоговому меню.
– змінити існуючу сторінку. Натисніть на назву сторіни або відзначте потрібну сторінку і натисніть кнопку "Изменить".
– видалити існуючу сторінку. Відзначте потрібну сторінку і натисніть кнопку "Удалить".
– дізнатися характеристики сторінки. Відзначте потрібну сторінку і натисніть кнопку "Свойства"(рис. 3.12).

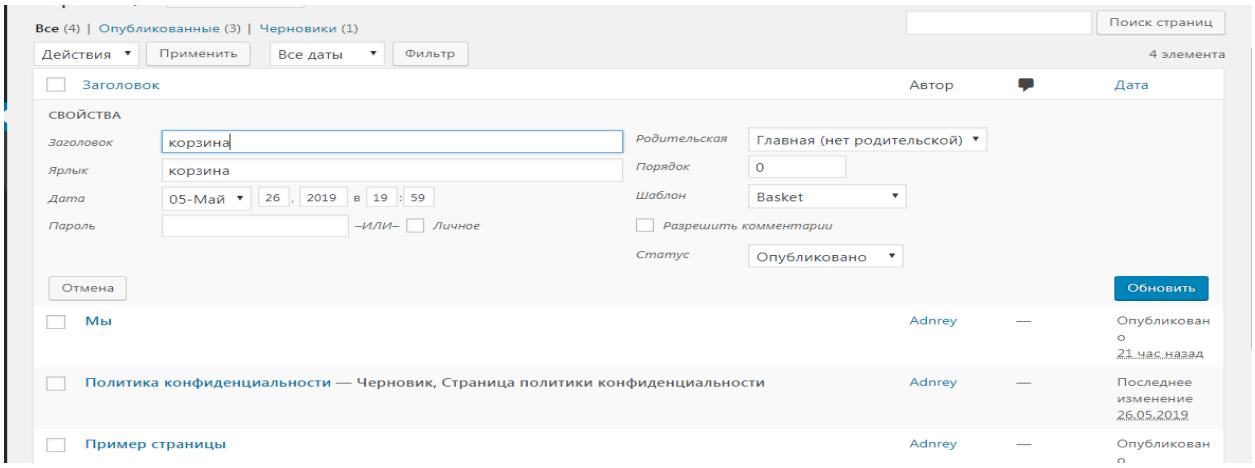

Рисунок 3.12– Характеристика сторінки

Також є можливість виділити та застосувати певню дію одразу до декількох сторінок.Дізнатися час публікації або останнього редагування сторінки. Для створення нової сторінки потрібно натиснути на кнопку "Добавить новую" у лівому діалоговому меню.Буде відкрита сторінка створення та редагування(рис. 3.13).

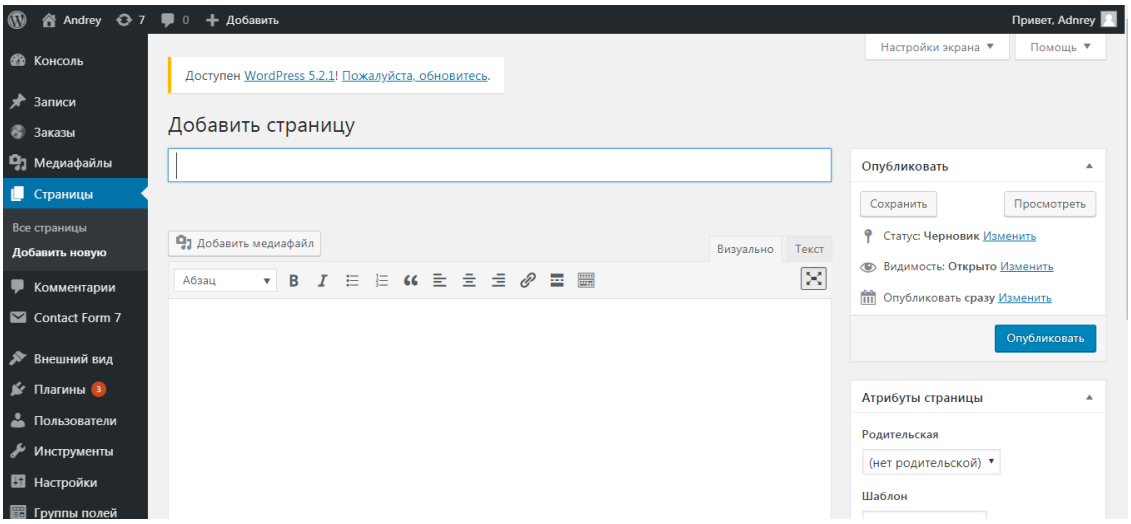

Рисунок 3.13– Панель створення та редагування

На сторінці редагування розділу за допомогою кнопок на панелі інструментів можна дати назву сторінці.Для цього пишемо бажану назву у

строку під “Добавить страницу” .Застосувати (зберегти) зміни, але залишитися на сторінці редагування. Натисніть кнопку "Обупликовать".Додати медіафайл.Для цього треба натиснути на кнопку “Добавить медиафайл” і слідувати інтсрукціям.Вибрати розмір та заданий пресет налаштувань тексту у полі під назвою “Абзац”.За допомогою кнопок “**B**” та “*I”* надати вибраному тексту характеристики: Жирний та Курсив.Промаркувати виділений текст , натиснувши на кнопки під назвою “Маркерованый список” та “Нумерованый список”.Розмістити текст по лівому,правому краях або по центру за домогою однойменних кнопок.Вибрати форму відображення чере кнопки “Визуально” та “Текст”. Зберегти створену або виправленнусторінку.Для цього потрібно натиснути на кнопку “Сохранить”.Для попереднього перегляду сторінку можна натиснути кнопку “Посмотреть”.Вибрати зображення сторінки.Для цього поторібно натиснути на посилання “Установить изображениестраницы” (рис. 3.14).Далі слідуючи інструкціям,налаштувати бажане зображення для сторінки.

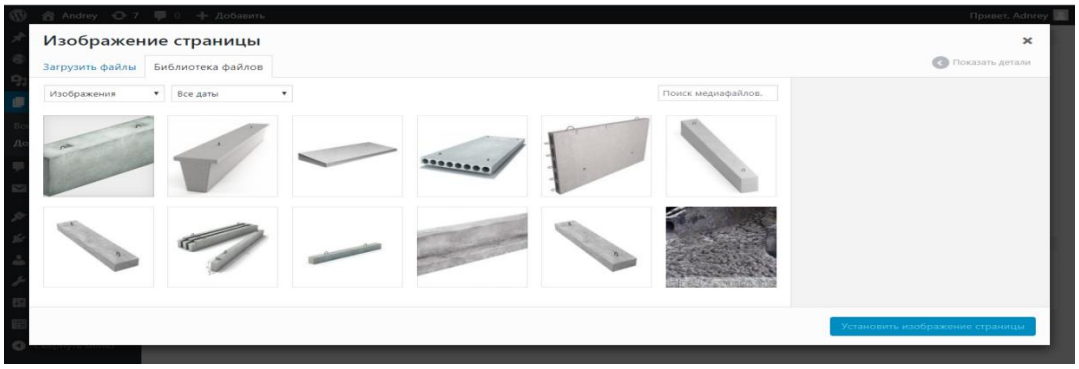

Рисунок 3.14– Панель додання зображення для сторінки

Увесь текст , або медіафайли які мі бажаемо бачити на створеній нами сторінці, ми додаємо у найбільше поле панелі редагування після натискання

кнопки “Визуально”.В меню “Текст” ми можемо вставляти код плагінів, та більш розширенно редагувати візуальну частину опису [9][1)].

3.4.3 Установка шаблону

Якщо ви самі безпосередньо встановлюєте і настроюєте Wordpress, то однією з перших дій з настройки є установка шаблону сайту для вашого проекту. Є багато вільно поширюваних шаблонів для Wordpress, і установка нового шаблону не є складною процедурою. Шаблони сайту для Wordpress зазвичай розповсюджуються (так само як і розширення) у вигляді .zip файлів (або .tar.gz файлу) .Для використання таких файлів необхідно зберегти файл шаблону і далі перейти на сторінку "Внешний вид" (рис. 3.15).

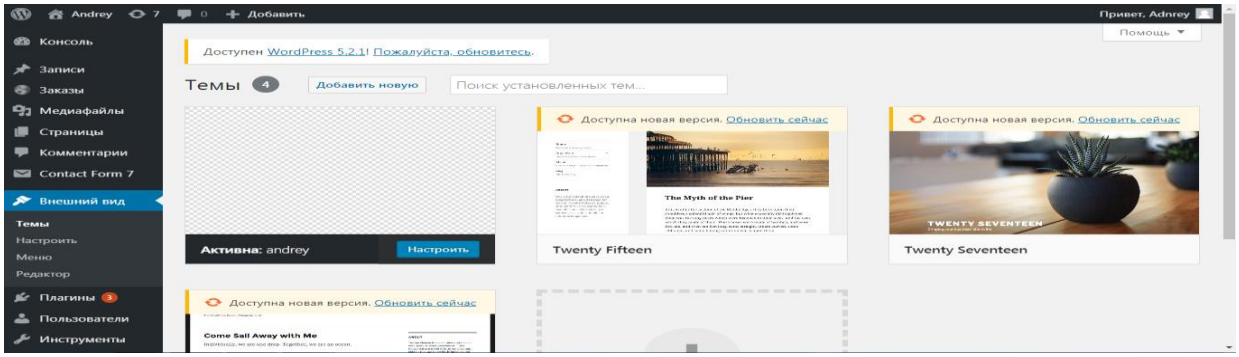

Рисунок 3.15 – Панель редагування темами

Використовуючи форму пошуку, можна знайти тему за назвою.

Натиснувши кнопку “Просмотреть”,ми маємо можливість переглянути тему до її активації. Для зміни теми потрібну навести курсор на бажану тему та натиснути на кнопку “Активировать”. Також є можливість власноруч

[1)][9] Веб-сайт wp-lessons.com.Опис змінитеми (шаблона). URL:https://wp-lessons.com/smena-temyi-shablona-wordpress(дата звернення 05.04.2019).

редагувати тему.Для цього потрібно активувати тему та натиснути кнопку “Настроить”(рис. 3.16).

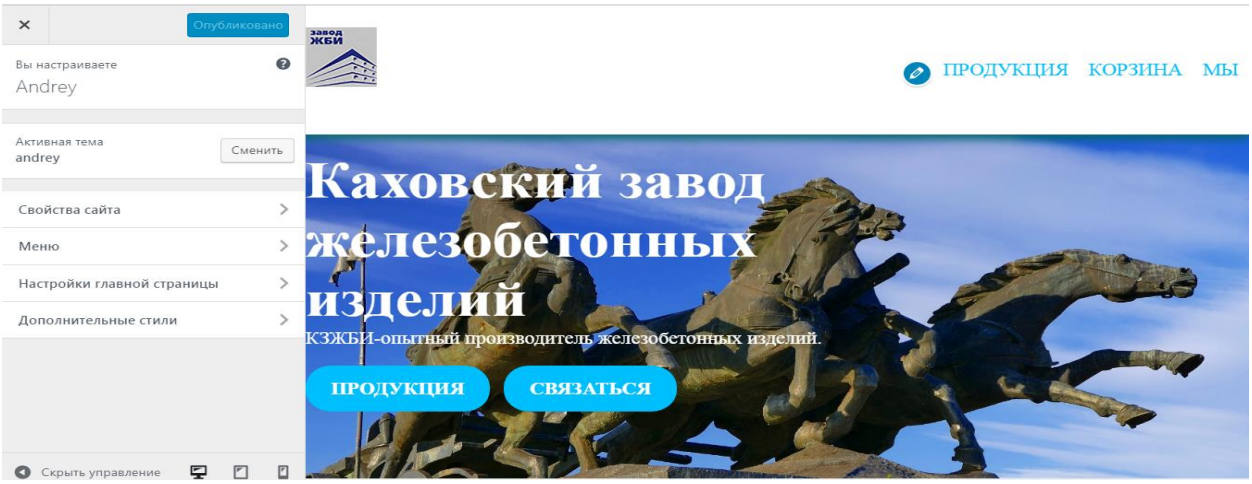

Рисунок 3.16 – Панель редагування теми

На даній сторінці ми маємо можливість редагувати тему за власним бажанням:

– змінити тему натиснувши на кнопку “Сменить” [10][1)].
– редагувати назву сайта, його опис, та поставити іконку розміром 512x512 можна за допомогою розділу “Свойствасайта”.
– створити, або редагувати існуюче меню веб-сайту.Для цього потрібно натиснути кнопку “Меню”.
– за допомогою меню “Настройка главнойстраницы” мы можемо вибрати, що саме буде висвітлено на головній сторінці.

[1)][10] Веб-сайт wp-lessons.comСменатемы (шаблона) WordPress. URL:https://wp-lessons.com/smena-temyi-shablona-wordpress (дата звернення 05.04.2019).

3.4.4 Встановлення плагінів

Використовуючи плагіни у системі Wordpress, ми можемо насичити веб-сайт за допомогою різноманітних функцій, які облегшують та покращують користування сайтом. Панель управління Wordpress надає можли вість відшукати на активувати бажані плагіни. Для цього потрібно натиснути на кнопку “Плагины” яка знаходиться у меню управління Wordpress в лівій частині сторінки (рис. 3.17).

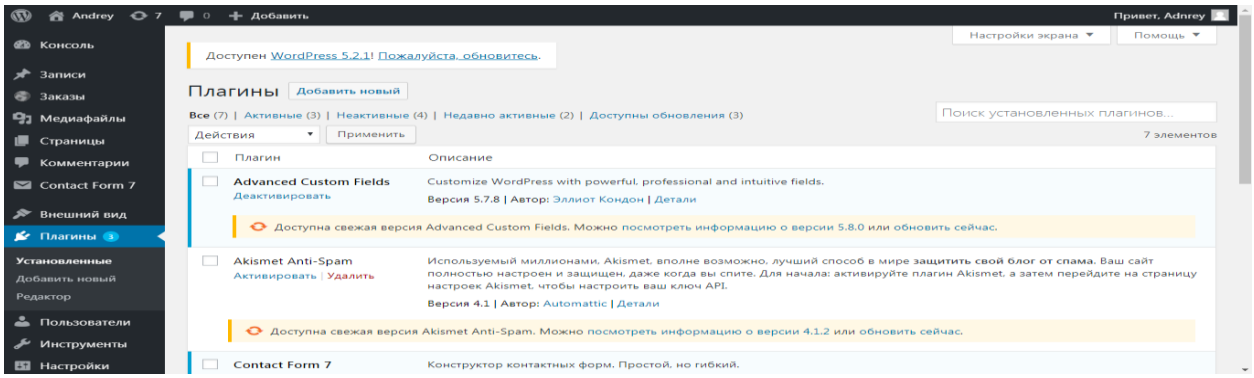

Рисунок 3.17 – Панель управління плагінами

На цій сторінці ми можемо взаємодіяти з вже встановленними плагінами.Для цього треба натиснути на кнопку “Активировать” .Якщо є потреба вимкнути один з плагінів,потрібно натиснути на кнопку “Деактивировать”.Це вимкне непотрібним нами додаток.Також присутні такі функції:

– відсортуватиплагіни.Для цього натискаємо на один із даних фільтрів.
– обновитиплагін.Якщо на момент використування , вийшла новіша версія плагіну , то ми можемо обновити його задля отримання нових функцій,або покращення старих.

– видалитиплагін.Післядеактиваціх плагіну він все одно залишається у панелі редагування.Якщо навести курсор на плагін,то з’явиться можливість видалити плагін назавжди.
– пошук додатків.За допомогою форми пошуку ми можемо знайти бажаний плагін за назвою.

Simple-basket.Завдяки цьому додатку була реалізована корзина для товарів, та система інтернет-магазину на цьому сайту(рис. 3.18)

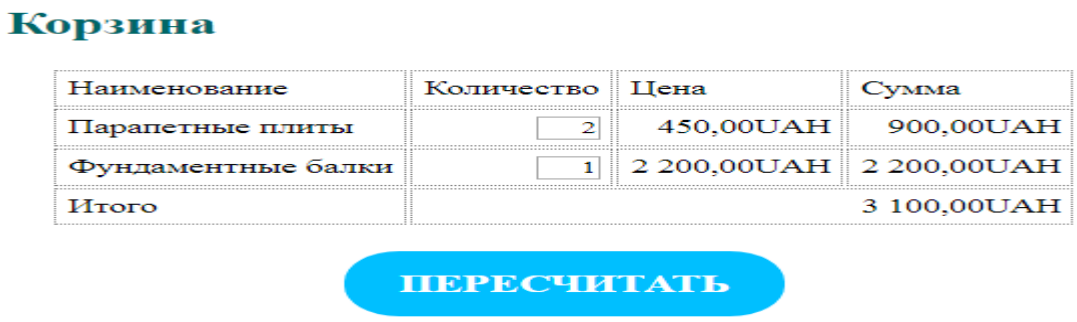

Рисунок 3.18 – Використання плагіну simple-basket

ContactForm 7.Цей плагін був використаний задля створення контактних форм для зв’язку менеджера ,або працівника відповідаючим за процесс спілкування з користувачем підприємства(рис. 3.19).

Рисунок 3.19 – Використання плагіну ContactForm 7

Також цей додаток я використав для створення форми створення замовлення у сторінці “Корзина”(рис. 3.20)

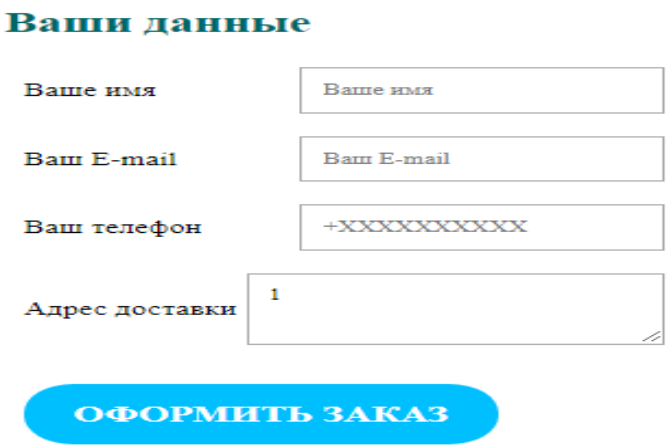

Рисунок 3.20 – Використання плагіну ContactForm 7 для створення формы створення замовлення

AdvancedCustomFields.Даний плагін був використан задля створення власних полів , наприклад таких як ціна товарів у власному інтернет-магазині (рис. 3.21)(рис.3.22).

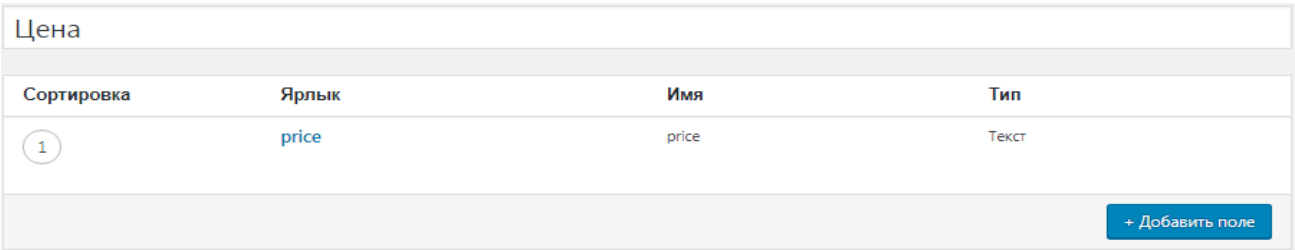

Рисунок 3.21 – Використання плагіну AdvancedCustomFields у CMS

Рисунок 3.22 – Зовнішній вигляд плагіну AdvancedCustomFields

3.4 Створення інтернет-магазину

3.4.1 Створення рубрики для інтернет магазину

На основі здобутих нами знань після аналізу та дослідження системи Wordpress було вирішено виконати інтернет-магазин у вигляді сторінки ,на яку було додано записи с назвами товару та підключенний плагін simple-basket.Отже створення сторінки було за таким алгоритмом:

– навівши курсор на меню “Записи” в панелі редагування Wordpress в лівій частині екрану, я вибрав пункт “Рубрики”.
– далі , викростивуючи меню створення категорії,надрукувавши ім’я ярлик та опис у задані поля, я натиснув на кнопку “Добавить новую рубрику’

Завдяки цим діям була створена відповідна до наших побажань рубрика(рис. 3.23).

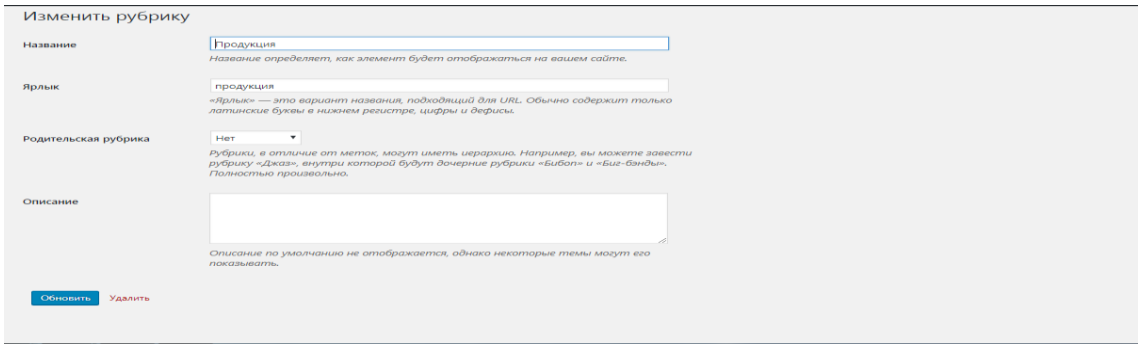

Рисунок 3.23 – Створена рубрика у панелі редагуваня

3.4.2 Створення товарів інтернет-магазину

Для стоверння товарів інтернет магазину ,я використав функцію CMSWordpress під назвою “Записи”.Натиснувши на дане посилання, я вибрав варіант “Добавить новую”. На відповідній сторінці я заповнив надані поля , інформацією замовника(рис. 3.24).

Рисунок 3.24 – Створення та редагування запису

Зображення для кожного товару було додано завдяки посиланню “Установить изображениестраницы”.Також у правій частині сторінки треба виділти потрібну рубрику.

Натиснувши на кнопку “Опубликовать” мы створюємо записи(товари) у вибрану рубрику.Результат можна побачити на відповідний сторінці веб-сайту(рис. 3.25).

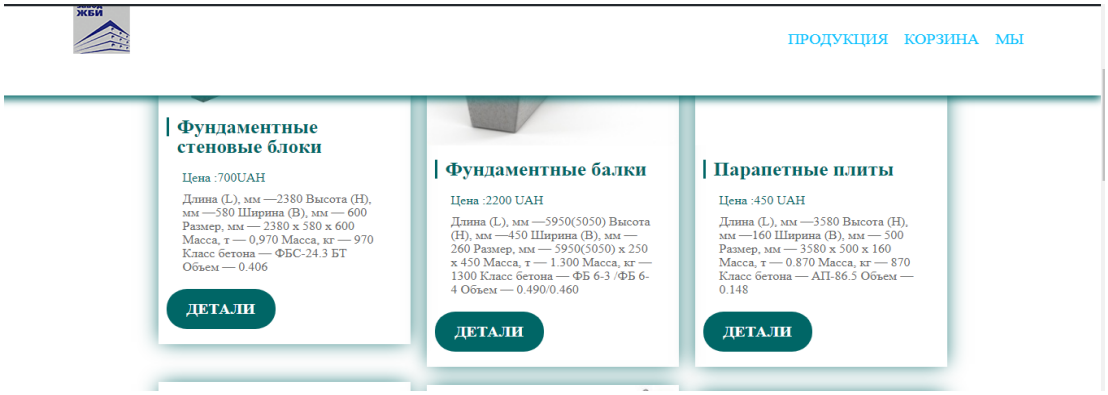

Рисунок 3.25 – Створені товари на відповідній сторінці

3.4.3 Створення корзини

Створення корзини виконуєтся за допомогою вбудованої функції у Wordpress під назвою “Страницы” .Отже натиснувши на меню “Страницы” та вибравши варіант “Добавить новую” я потрапив до панелі створення сторінкою.Заповнивши її потрібною інформацією я отримав відповідну сторінку на веб-сайті (рис. 3.26)

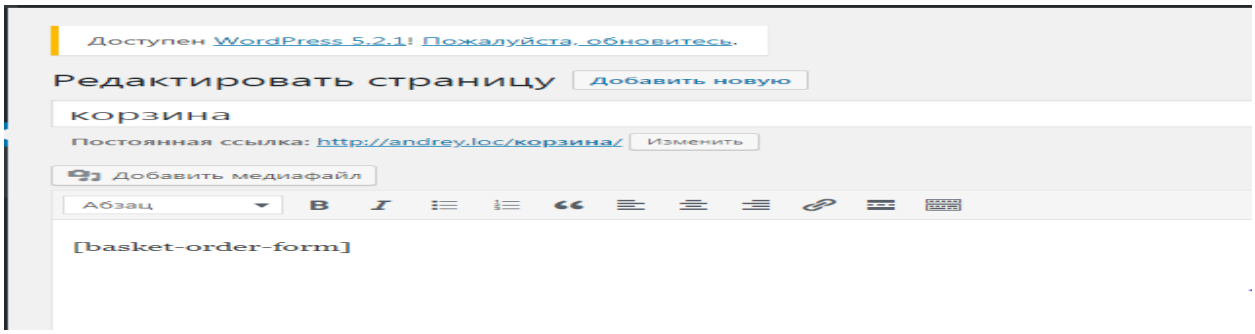

Рисунок 3.26 – Панель редагування та створення сторінки “Корзина”

У полі заповнення додатковою інформацією я підключив плагін simple-basket командою [basket-order-form].

Для того , щоб плагін корректнопрацював,потребуєтся створення та заповення кодом відповідний basket-tamplate.php файл. Цей файл є шаблоном сторінки із корзиною. Отже, завдяки виконаним нами діями вище, ми створили працюючу та оформлену сторінку “Корзина”.

3.4.4 Створення сторінки “Мы”

Сторінка “Мы” відповідає за надання географічної та контактної інформації користувачу.

Додання сторінки на веб-сайт відбувається аналогічним чином як і попередня.

Отже на панелі редагуванні,потрібно натиснути на меню “Текст” і вписати наступний код:

```
<div style="display: flex;">
<div style="overflow: hidden; width: 700px; position:
relative;">
<h4><iframe
src="https://maps.google.com/maps?width=700&amp;height=440&amp;h
l=en&amp;q=Ukraine%2CKakhovka%2CPromyslova%20Vulytsa+(%D0%9D%D0%
B0%D0%B7%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)&amp;ie=UTF8&amp;t=&amp;z
```

```
=14&amp;iwloc=B&amp;output=embed" width="700" height="440"
frameborder="0" marginwidth="0" marginheight="0"
scrolling="no"></iframe></h4>
```

Даний код відповідає за встановлення iframe з Google Мапами.

Завдяки веб-сайту https://embedgooglemaps.com/ru/[11][1)], ми визначаємо потрібну нам адресу, та отримуємо iframe код , який ми вставляємо у відповідне поле в панелі редагування сторінкою.

На самому веб-сайті це виглядає як Google Мапа з якою користувач може взаємодіяти(рис. 3.27).

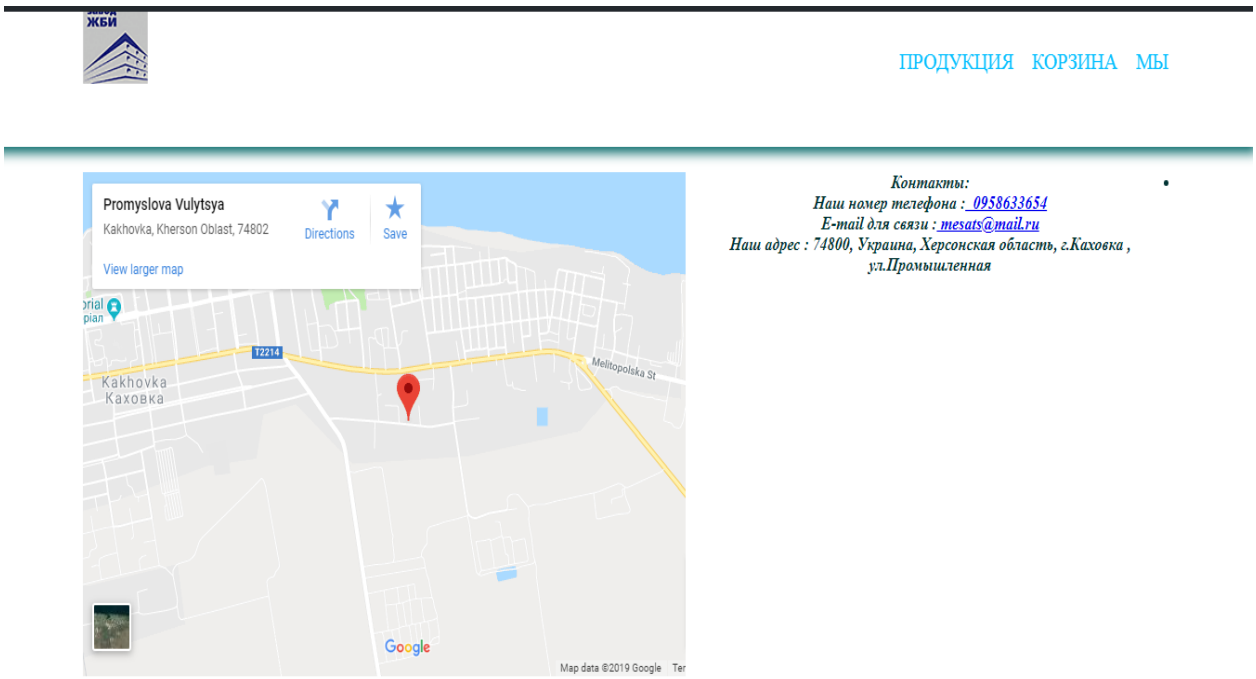

Рисунок 3.29 – Зовнішній вигляд iframe коду на веб-сайті

Також , на даній сторінці були додані розширені контактні данні для комфортнішого користування та швидкого зв’язку із представниками підприємства.

---

[1])[11] Embedgooglemaps. URL: https://embedgooglemaps.com/ru/(дата звернення 05.04.2019).

ВИСНОВКИ

Згідно технічного завдання було створено сайт інтернет-магазин на базі Wordpress. Був створений дизайн сайта в виді шаблона, який використовується в системі управління вмістом Wordpress. На головній сторінці розміщено логотип веб-сайта, основне меню сайту (для навігації по його структурі),форму зв'язку із представниками підприємства. Також сайт має початкові категорії товарів та початкові товари для відображення працездатності інтернет-магазина. Також сайт має сторінки, які мають інформативний вміст для користувачів і корисну інформацію для клієнтів інтернет-магазину.В ході виконання бакалаврьскої кваліфікаційної роботи був працюючий та функціональних веб-сайт, повністю готовий до застосування. Даний сайт орієнтований для широкого спектру застосування в on-line торгівлі. За його допомогою користувачі отримують можливістьпрацювати із необхідною інформацією про товар, а також можливість купівлі товару. При розміщенні його в веб- мережі географія розповсюдження зростає до всесвітніх маштабів.При розробці проектубули проаналізовані сучасні технології, що дозволяють створювати інтерактивні веб-сторінки. Найбільш відповідними для виконання поставленого завдання виявилися Wordpress 4.9.10, Notepad ++ та веб-браузер GoogleChrome.

Розроблений сайт задовольняє всім вимогам, поставленим на етапі постановки завдання. При розробці веб-сайта були використані готові контактних форм ,геолокації , on-line магазин. Дані модулі були допрацьовані з урахуванням специфіки веб-сайта і успішно упроваджені в його структуру.

За подальше вдосконалення веб-сайта представляється можливим розробка модулів доступу на веб-сайті. Також можливе доопрацювання

інтерфейсу сайту з метою подальшого підвищення його інформативності, привабливості і зручності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. University web developers.URL: http://cuwebd.ning.com(дата звернення 03.04.2019).
2. Основы XML для начинающих пользователей.URL:https://www.ibm.com/developerworks/ru/library/x-newxml/index.html (дата звернення 03.04.2019)
3. Офіціний сайт php.net URL: https://www.php.net/manual/ru/book.mysql.php (дата звернення 03.04.2019)
4. Веб-сайт .w3schoolsHTMLTutorials.Інформація про вивчення HTML.URL: https://www.w3schools.com/html/ (дата звернення (02.04.2019)..
5. Матеріал з Вікипедії вільній енциклопедії про Debian.URL: https://ru.wikipedia.org/wiki/Debian (дата звернення 02.04.2019).
6. Матеріал з Вікипедії — вільній енциклопедії про системи управління сайтом. http://ru.wikipedia.org/wiki/CMS (дата звернення 03.04.2019).
7. Веб-сайт joomla.ru.ИсторияJoomla. URL: http://joomla.ru/about/history
8. Опис керування панеллю адміна.URL: https://wp-kama.ru/cat/wordpress/adminka-wordpress.(дата звернення 03.04.2019).
9. Веб-сайт wp-lessons.com. Опис зміни теми (шаблона) Word-Press URL:https://wp-lessons.com/smena-temyi-shablona-wordpress (дата звернення 05.04.2019).

10. Веб-сайт wp-lessons.comСменатемы (шаблона) WordPress. URL:https://wp-lessons.com/smena-temyi-shablona-wordpress (дата звернення 05.04.2019).
11. Embed google maps. URL: https://embedgooglemaps.com/ru/(дата звернення 05.04.2019).

ДОДАТОК А

Лістинг програми

А.1 Код шаблону сайту

```
section{
background: white;
margin: 0px;
}
*{
font-family:'sans-serif';
  margin:0px;
padding: 0px;
}
body{
padding-top: 120px;
}
header{
position: fixed;
top: 0px;
left: 0px;
width: 100%;
background: #fff;
box-shadow: -5px 0px 10px 5px #006666;
height:auto;
  z-index: 10;
}
headerimg{
width: 70px;
}
```

```
.desctop{
display: block;
}
.mobile{
display: none;
}
nav a {
color:        #00BFFF;
font-size: 20px;
padding-bottom: 25px;
text-decoration: none;
text-transform: uppercase;
}
nav{
display: flex;
justify-content: space-between;
align-items: center;
height: 100%;
margin: 25px 0px;
}
navul{
display: flex;
align-items: center;
list-style: none;
padding-left: 0px;
margin: 0px;
}
navul li {
margin: 0px 10px;
}
.wrapper{
max-width: 1180px;
margin: 0 auto;
```

```
}
section#main{
min-width: 100vw;
min-height: calc(100vh - 80px);
background: url(maxresdefault.jpg);
  background-size:100% 100%;
}
#main h1{
font-size: 60px;
color: #fff;
}
#main p{
font-size: 20px;
color: #fff;
}
.mainText{
max-width: 550px;
}
button{
font-size: 20px;
padding: 15px 25px;
border-radius: 25px;
display: inline-block;
border: 0px solid #FFF;
color: white;
font-weight: 600;
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
box-sizing: border-box;
text-align: center;
background: #00BFFF;
text-transform: uppercase;
}
```

```
section{
padding: 50px 0px;
position: relative;
}
#second{
background:   #00BFFF;
}
#second *{
color: white;
}
#secondul{
list-style: none;
}
#secondul li{
background-image: url(http://west-berton-impex.com/wp-content/themes/berton/img/srtelka-min.png);
background-size: 100% 100%;
padding: 15px 65px 15px 55px;
margin: 20px 0px;
color: #003333;
}
.bigImg{
position: absolute;
right: 0px;
top: 0px;
height: 100%;
}
h2{
font-size: 50px;
margin: 0px 0px 20px;
font-weight: 600;
}
#third{
```

```
background:   #DCDCDC;
}
#third h2{
color: #006666;
}
.itemHolder{
display: flex;
align-items: center;
}
.itemHolder .item{
width: 50%;
margin: 25px 0px;
}
.itemHolder .itemimg{
display: block;
margin: 0 auto;
}
.itemul{
list-style: square;
padding-left: 25px;
}
.itemul li {
color: #006666;
margin: 20px 0px;
}
.itemul li span{
color: #000000;
}
.item h3{
color:#006666;
}
#fourh{
background: white;
```

```
}
#fourh .wrapper{
display: flex;
}
#fourh .item{
width: 50%;
display: flex;
align-items: center;
}
#fourh .item h2 {
color: #006666;
}
#fourh .item h3{
color: #666666;
}
#fourhform{
padding: 50px;
background-color: #00BFFF;
}
#fourhinput{
width: 100%;
box-sizing: border-box;
font-size: 20px;
background: #ffffff;
padding: 15px;
color: #929397;
margin: 25px 0px;;
}
#fourhinput[type="submit"]{
background-color: #00BFFF;
color:#fff;
border: 1px solid #fff;
}
```

```
footer{
background-color:  #00BFFF;
padding: 30px 0px;
}
footerul{
list-style: none;
color:#003333;
}
footerul a{
color: #003333;
text-decoration: none;
}
ul.contacts{
margin: 0px;
color:#003333
```

А.2 Код файлу header.php

```
<!DOCTYPE html>
<htmllang="en" dir="ltr">
<head>
<linkrel="stylesheet"
href="<?phpechoget_template_directory_uri()?>/css/style.css?r=<?
=time();?>">
<link
href="https://fonts.googleapis.com/css?family=Rubik:300,400,700,
900&display=swap" rel="stylesheet">
<metacharset="utf-8">
<title></title>
</head>
<body>
<header>
<div class="desctop">
<div class="wrapper">
<nav>
<a
href="/"><imgsrc="<?phpechoget_template_directory_uri()?>/Zbi.pn
g" alt="logo"></a>
<?phpwp_nav_menu( array( 'theme_location' => 'new-menu' ) ); ?>
</nav>
</div>
</div>
<div class="mobile">
<img src="http://west-berton-impex.com/wp-
content/themes/berton/img/Logo%20green.svg" alt="">
<div class="burger">
      </div>
<nav>
```

```
<ul>
<li></li>
</ul>
</nav>
</div>
<?phpwp_head();?>
</header>
<?php
```

А.3 Код файлу footer.php

```
     <footer>
<div class="wrapper">
<div class="col">
<imgsrc="<?phpechoget_template_directory_uri()?>/Zbi1.png"
width="100px" alt="">
<ulclass="contacts">
<li><a href="tel:+380958633654">0958633654</a></li>
<li><a href="mailto:mesats@mail.ru">mesats@mail.ru</a></li>
<li>74800, Украина, Херсонская область, г.Каховка</li>
</ul>
</div>
</div>
</footer>
<?phpwp_footer();?>
</body>
</html>
```

А.4 Код файлу front-page.php

```
<?phpget_header();?>
<sectionid="main">
<div class="wrapper">
<div class="mainText">
<h1>Каховский завод железобетонныхизделий</h1>
<p style="margin-bottom:20px;">КЗЖБИ-
опытныйпроизводительжелезобетонныхизделий.</p>
<buttontype="button" name="button"><a
href="http://andrey.loc/category/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%
D0%BA%D1%86%D0%B8%D1%8F/" style="text-
decoration:none;color:#Fff;">Продукция</a></button>
<buttontype="button" name="button" style="margin-left:10px;"><a
href="#fourh" style="text-
decoration:none;color:#Fff;">Связаться</a></button>
</div>
</div>
</section>
<sectionid="second">
<div class="wrapper">
<div style="max-width:650px;">
<h2>О НАС</h2>
<p>КЗЖБИ – опытныйпроизводительжелезобетонныхизделий с
многолетнейисторией .Успехнашегопроизводстваоснован на таких
концепциях:</p>
<div class="">
<ul>
<li>Наша репутация основана на богатойистории и лидерством на
рынке.</li>
<li>Продукциядоставляется по всейстране в каждую область.</li>
<li>100% надежность и понимание в переговорах междузаказчиком и
проедприятием.</li>
<li>Успешнаяреализациявсехбизнес-процессов.</li>
</ul>
</div>
</div>
</div>
<img src="http://west-berton-impex.com/wp-
content/uploads/2018/11/Sunflower-min.png" class="bigImg">
</section>
<sectionid="third">
<div class="wrapper">
<h2>Преимуществоработы с нами</h2>
<div class="itemHolder">
<div class="item">
<img src="http://west-berton-impex.com/wp-
content/themes/berton/img/quality-min.png" alt="">
</div>
<div class="item">
```

```
<h3>1.Поставка сертифицированнойпродукциивысокогокачества</h3>
<ul>
<li><span>Вся продукция , предоставляемаякомапнией,
сертефицирована на
соотвествиетребованияммеждународныхстандартов.</span></li>
<li><span>Компанияпредоставляетклиентамполный пакет
документов,включаясчет,упаковочныйлист,коносамент.</span></li>
</ul>
</div>
</div>
<div class="itemHolder">
<div class="item">
<h3>2.Опытная и проффесиональная команда всегда готова
предоставить:</h3>
<ul>
<li><span>Консультация и ответы на все вопросы.</span></li>
<li><span>Помощь в принятииверногорешения и выборлучшегопродукта
в соотвествии к вашим предпочтениям</span></li>
<li><span>Гараниюиндивидуальногоподхода и
качественогосервиса.</span></li>
</ul>
</div>
<div class="item">
<img src="http://west-berton-impex.com/wp-
content/themes/berton/img/Hands-min.png" alt="">
</div>
</div>
<div class="itemHolder">
<div class="item">
<img src="http://west-berton-impex.com/wp-
content/themes/berton/img/truck-min.png" alt="">
</div>
<div class="item">
<h3>3.Мы работаемразными видами транспорта</h3>
<ul>
<li><span>Дорожный транспорт</span></li>
<li><span>Железнодорожный транспорт</span></li>
<li><span>Речной и морскойтранспорты</span></li>
</ul>
</div>
</div>
</div>
</section>
<sectionid="fourh">
<div class="wrapper">
<div class="item">
<div class="">
<h2>СВЯЖИТЕСЬ С НАМИ</h2>
<h3>Заполните форму и наш менеджер свяжиться с вами</h3>
</div>
</div>
<div class="item">
<?phpechodo_shortcode('[contact-form-7 id="6" title="Contactform
1"]')?>
</div>
</div>
</section>
<?phpget_footer();?>
```