

Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій та механотроніки

(повна назва факультету)

Кафедра комп'ютерних та інформаційних технологій і систем

(повна назва кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(освітньо-кваліфікаційний рівень)

на тему

Розроблення веб-сайту автосалону

Виконав: студент 4 курсу, групи 402-ТН
спеціальності

122 Комп'ютерні науки

(шифр і назва напрямку)

Чернишенко Т.С.

(прізвище та ініціали)

Керівник

Руденко О.А.

(прізвище та ініціали)

Рецензент

_____ (прізвище та ініціали)

Полтава – 2021 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ
ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ТА МЕХАНОТРОНІКИ**

**КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
спеціальність 122 «Комп'ютерні науки»**

на тему

«Розроблення веб-сайту автосалону»

Студента групи 402-ТН Чернишенка Тараса Сергійовича

Керівник роботи
кандидат технічних наук,
доцент Руденко О.А.

Завідувач кафедри
кандидат технічних наук,
доцент Головка Г.В.

Полтава – 2021

РЕФЕРАТ

Загальний обсяг дипломного проекту містить 60 сторінок, 24 малюнків, 10 таблиць, 17 джерел, 3 додатків. Вона складається з переліку умовних позначок, скорочень, термінів, вступу, 3 розділів, висновка, списку використаних джерел, та 3 додатків.

У першому розділі проводиться аналіз предметної області, а також розміщене в ньому технічне завдання.

У другому розділі проводиться проектування, вказано етапи розроблення та розміщення сайту, створюється структура, вказується даталогічна модель системи, здійснюється опис дизайну сайту.

У третьому розділі вказано вимоги, що повинен містити заданий сайт, узагальнено поняття мов веб-програмування, вказано використання програмних продуктів, здійснено огляд та порівняння локальних серверів.

Об'єкт дослідження: інформаційна система Web-сайту автосалону.

Предмет дослідження: технологія створення Web-сайту засобами Web-програмування та систем CMS, СУБД.

Мета роботи: придбання теоретичних знань і практичних умінь з розробки сайтів, із застосуванням засобів Web-програмування та систем CMS, СУБД, а також введення в продаж доступних автомобілів.

Методи: пошукові, інформаційні, аналіз предметної області, використання технологій web-програмування, проектування web-сайту за допомогою CMS, СУБД.

Ключові слова: Web-сайт, web-сторінка, автосалон, автомобілі, база даних, web-сервер, MySQL, CMS, СУБД.

ANNOTATION

The total volume of the diploma project contains 45 pages, 24 figures, 10 tables, 17 sources, 3 appendices. It consists of a list of symbols, abbreviations,

terms, introduction, 3 sections, conclusion, list of sources used, and 3 appendices.

The first section analyzes the subject area, as well as the technical task placed in it.

In the second section the design is carried out,

the stages of site development and placement are indicated, the structure is created, the data model of the system is indicated, the site design is described.

The third section specifies the requirements that must contain a given site, summarizes the concept of web programming languages, indicates the use of software products, reviews and compares local servers.

Object of research: information system of the Motor Show Web-site.

Subject of research: technology of creating a Web-site by means of Web-programming and CMS systems, DBMS.

Purpose: acquisition of theoretical knowledge and practical skills in website development, using Web programming tools and CMS systems, DBMS, as well as the introduction of affordable cars.

Methods: search, information, analysis of the subject area, the use of web-programming technologies, web-site design using CMS, DBMS.

Keywords: Web-site, web-page, car showroom, cars, database, web-server, MySQL, CMS, DBMS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧОК, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	8
РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ	11
1.1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.2.ТЕХНІЧНЕ ЗАВДАННЯ.....	13
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА САЙТУ	14
2.1 КОНЦЕПТУАЛЬНЕ ПРОЕКТУВАННЯ.....	14
2.2 ДАТАЛОГІЧНА МОДЕЛЬ СИСТЕМИ	16
2.3 СТРУКТУРА РОЗРОБЛЮВАНОВОГО САЙТУ	19
2.4 РОЗРОБКА ДИЗАЙНУ САЙТУ.....	21
2.5 СТВОРЕННЯ БАЗИ ДАНИХ WEB-САЙТУ	22
2.6 ФІЗИЧНА СТРУКТУРА САЙТУ	24
2.7 ПРОЦЕС РОЗМІЩЕННЯ САЙТУ НА ХОСТИНГУ	25
РОЗДІЛ 3 ВИКОРИСТОВУВАНІ МОВИ ПРОГРАМУВАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	31
3.1.ВИМОГИ ДО РОЗРОБЛЮВАНОВОГО САЙТУ.....	31
3.2 МОВИ ПРОГРАМУВАННЯ.....	33
3.2.1 Поняття HTML.....	33
3.2.2 Поняття CSS.....	36
3.2.3 Поняття PHP.....	37
3.2.4 Поняття JavaScript.	39
3.2.5 Поняття реляційної системи управління базами даних MySQL.....	40
3.3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.....	41
3.3.1 Огляд та порівняння сучасних систем управління вмістом.....	41
3.3.2 Огляд та порівняння локальних серверів.....	47
ВИСНОВОК	53

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
ДОДАТОК А	57
ДОДАТОК В.....	58
ДОДАТКИ С.....	59

ПЕРЕЛІК УМОВНИХ ПОЗНАЧОК, СКОРОЧЕНЬ І ТЕРМІНІВ

HTML (HyperTextMarkupLanguage) – мова розміток гіпертексту.

CSS (CascadingStyleSheets) – мова каскадних таблиць стилів.

CMS (ContentManagementSystem) – система управління контентом.

Web-сайт – ресурс розміщений в мережі Інтернет, який складається з декілька Web-сторінок пов'язаних між собою посиланням.

Web-сторінка – це сторінка з розташованим на ній мультимедійним контентом та яка розміщена в мережі Інтернет.

Web-програмування – це створення сайтів, додатків та програмного забезпечення для роботи в мережі Інтернет.

DTD (DocumentTypeDefinition) – мова запису метаданих, що містять в собі синтаксичні правила розмітки тексту.

WWW(WorldWide Web, або Web) – всесвітня паутина, або інформаційний простір для обміну даними по всьому світу, поєдную в собі безліч мереж, що поєднані між собою.

HTTP(HyperTextTransferProtocol) – мережевий протокол передачі гіпертекстової інформації.

HOSTING – цеспільне розміщення Webсайту на обладнанні компанії.

DBMS (DatabaseManagementSystem) або **СУБД,СКБД**(Система управління базами даних) – набір взаємозв'язаних між собою баз даних та програм, які мають в собі можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до них.

ВСТУП

На теперішній час важко уявити життя без машин. Великі можливості в комерційній сфері дають широке представлення для формування та розширення будь-яких мереж автосалонів і магазинів по всьому світу. При середньому рівні достатку, продаж автомобілів стає дуже привабливим бізнесом, в якому кожен може реалізувати себе в різних областях зайнятості, зайняти свою нішу. В край тяжко мати можливості зводити автосалони при мінімальних затратах та матеріальних витратах, які повністю відповідають державним, тобто комерційним вимогам і стандартам, а також забезпечують зручність для покупців і ефективність при демонстрації автомобільної продукції в цілому [15].

Особливо в часи коронавірусної пандемії, створення таких онлайн сервісів стало пріоритетним завданням для виходу з фінансових проблем. Масове закриття будь-якого бізнесу завдали чималі збитки в економіці багатьох країн світу. У зв'язку зі закриттям метро і припиненням руху громадського транспорту, роль автомобіля багаторазово зросла. Автомобілі, як і раніше, вимагають регулярний догляд, ремонт та належне обслуговування, а ті, хто замовив новий автомобіль, або хоче його придбати чекають, коли реалізуються їхні бажання. Але де ж їх купити, перевірити та дізнатися як не в салонах та магазинах, що закрилися?

Під час карантину автосалони не приймають відвідувачів і не проводять в повсякденні покази та демонстрації продукції, не здійснюють очне консультування, а також зали всі зачинені. Замість цього, робиться все можливе, щоб клієнти могли отримати максимум інформації з питань вибору та придбання автомобілів онлайн. Працівники перейшли на дистанційну (віддалену) роботу онлайн. Тобто власникам автомобілів багатьох марок та потенційним покупцям не варто хвилюватися. В цей нелегкий час, для оточуючих потрібно бути мобільним в максимально безпечних умовах, тому робота дилерських мереж перейшла в такий

формат, що можна вибрати автомобіль і зробити покупку, не відвідуючи сам автосалон [16].

Становиться зрозумілим, що створення таких сервісів має цілком позитивний характер, в умовах пандемії та тотальної ізоляції. Таким чином, вище сказане підтверджує актуальність створення та розширення функціоналу онлайн додатків, веб-послуг як автосалон.

Оптимальним та найкращим рішенням стає створення таких Web-сайтів автосалону. Переваги та недоліки такого рішення:

Переваги :

- Високий рівень безпеки, неможливість захворюваності через коронавірусну пандемію.
- Віддаленне та зручне відвідування з будь-якого місця та в будь-який час.
- Пошук і перевірка наявності вибраної моделі.
- Швидкий зворотній зв'язок та онлайн підтримка.
- Віддалений тест-драйв
- Адресна доставка додому автомобіля

Недоліки :

- Зовнішній вид товару може відрізнитися від продукції, що дійсна в реальності.
- Не можливо відчутти на дотик та перевірити на зручність обрану модель, що може задовольнити потенційного клієнта.
- Не завжди має можливість продавати, встановлювати ціни на товар і таке інше, але добре демонструє наявність в ньому товару.

Актуальність теми: створення сайту автосалону надає можливість підвищити ефективність взаємодії між співробітниками автосалону та ймовірним клієнтом, надаючи необхідну інформацію з будь-якого зручного місця, та в будь-який час не відвідуючи сам автосалон.

Метакваліфікаційної роботи: придбання теоретичних знань і практичних умінь з розробки сайтів, із застосуванням засобів Web-програмування та систем CMS, а також введення в продаж доступних автомобілів.

Об'єкт дослідження: аналіз предметної області та процес створення веб-сайту.

Методи дослідження: пошукові, інформаційні.

Очікуваний результат: виконання поставлених завдань повинно бути впровадження створеного веб-сайту автосалону «VS-Motors» та його розміщення на хостингу в мережі Інтернет.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ

1.1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Будь-яка інформація, що зберігається в мережі інтернет, розташована на комп'ютерах так названих web-серверах, в яких, встановлено спеціальне програмне забезпечення, що допомагає користувачу шукати потрібну інформацію. Значна частина таких даних збережена у вигляді web-сторінок, які в свою чергу мають адреси та імена в Інтернеті. Для перегляду web-сайтів користувачем повинне бути встановлено програмне забезпечення, яке називають браузером. Це програмне забезпечення зображує в своєму вікні, ту частину інформації, що відповідає дійсності, зважаючи на те яка адреса задана в “адресному рядку” свого робочого вікна.

Будь-який web-сайт має в собі зв'язані між собою web-сторінки. Кожна така сторінка являє собою текстовий файл, що має розширення * .html, а також містить інформацію та набір спеціальних команд названим, як HTML-код відображає певну інформацію в браузері. Web-сторінки не мають в собі графічну, відео та аудіо інформацію, але включає посилання на файли, що містяться в HTML-коді.

Використання баз даних застосовується для зображення з певної предметної області інформації. Тобто, це така собі схованка яка існує в межах області проектування та має вид баз даних. В ній визначено яка саме інформація буде зберігатися і оброблятися в базі даних [1].

Всі web-сайти діляться на такі види як: **статичні** та **динамічні**.

Статичний сайт – має з набір незмінних HTML-сторінок, які зв'язані один з одним посиланнями. Такі сторінки вручну створюються, зберігаються і завантажуються на сервер, після цього при кожному зверненні до веб-сайту

представляються користувачеві в незмінному вигляді. А для зміни цих даних потрібно кожний раз особисто коригувати програмний код веб-сторінки вручну.

Переваги та недоліки такого статичного типу сайтів.

Переваги :

- мінімальне навантаження на сервері;
- швидкодія;
- дешева розробка;
- простота перенесення на новий хостинг [2].

Недоліки :

- складність оновлення сайту;
- внесення будь-яких змін;
- складність адміністрування сайтом (на рівні Web-програмування).

Динамічний сайт - передбачає можливість редагування вмісту веб-сторінок сайту, не використовуючи технології Web-програмування. Зовнішній вид сторінок таких Web-сайтів створюється за шаблонною структурою, яке динамічно заповнюється інформацією, що береться частіше з бази даних. І коли запит веб-сторінки користувачем виконується вибірка з бази даних, яка записується в шаблон, створивши в цей час нову Web-сторінку, і пересилається Web-сервером до клієнта в Web-оглядач (браузер), який і зображує її потрібним чином. Дозвіл на внесення змін має тільки обмежене коло людей, серед яких адміністратори та модератори. Також є виключення з таких правил для всіх інших користувачів, що не мають особливих прав. (наприклад, сайти-форуми)

Переваги та недоліки такого динамічного типу сайтів.

Переваги :

- спрощене редагування вмісту;
- гнучкість в керуванні;
- варіативність створення як вручну так із використанням CMS.

Недоліки :

- дають значне навантаження на сервер;
- більша затратність ресурсів сервера;
- більші вимоги до хостингу.

Особливістю CMS є використання вже готових модулів і компонентів, без можливості створювати їх заново кожного разу. Користуючись CMS, є можливість створювати будь-яку кількість динамічних веб-сайтів. У таких сайтах реалізовано розподіл змісту контенту і оформлення, що дає можливість, вчасно змінювати інформацію без необхідності редагувати програмні коди сторінок. Це є одним з найголовніших переваг [3].

1.2.ТЕХНІЧНЕ ЗАВДАННЯ

Метою кваліфікаційної роботи є придбання теоретичних знань і практичних умінь з розробки сайтів із застосуванням засобів веб-програмування CMS, а також створенню баз даних в середовищі MySQL.

Для реалізації мети дипломної роботи необхідно вирішити наступні завдання:

- Вивчити теоретичні аспекти створення Web-сайтів.
- Провести аналіз існуючого програмного забезпечення для створення Web-сайтів.
- Розглянути мови програмування, що використовуються в Web-технологіях.
- Провести аналіз засобів створення баз даних для Web-сайтів.
- Створити структуру і призначений для користувача інтерфейс сайту.
- Створити базу даних для сайту.
- Провести аналіз працездатності створеного сайту.

Результатом виконання поставлених завдань повинен стати інтернет-сайт «Автосалон».

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА САЙТУ

2.1 КОНЦЕПТУАЛЬНЕ ПРОЕКТУВАННЯ

Концептуальний дизайн - це побудова семантичної моделі тематичної області та її опису. Створений процес проектування не звертає увагу на якусь конкретну систему управління базою даних.

Структура БД - це модель даних, яка дає здатність описувати концептуальну схему предметної області. Модель даних являють собою графічний опис предметної області, використовуючи набір стандартизованих назв. Для вибраних СУБД зводиться логічна модель на основі структури БД.

Логічна модель відображає процес проектування моделі, заснована на створеній моделі даних, але не враховує певних СУБД та інші фізичні обмеження [9].

Перетворення логічної моделі даних у фізичну відбувається за рахунок таких чинників:

- Перетворення об'єктів у таблиці може здійснюватися лише у фізичній базі даних;
- Для перетворення атрибутів у поля необхідна фізична база даних, і для атрибутів потрібно вибрати типи даних;
- Будь-який тип унікального ідентифікатора стає первинним ключем, тобто воно не може бути нульовим полем.

Відносини в базі даних, або іншими словами, відносини - це графічно описана асоціація, встановлена між двома різними сутностями. Ця асоціація завжди є дуалістичною і може існувати між двома різними сутностями або між суттю та собою.

Основні елементи, що складають структуру БД:

- Об'єкти (сутності) предметної області;
- Прямий зв'язок між об'єктами;
- Атрибути (атрибути) об'єкта.

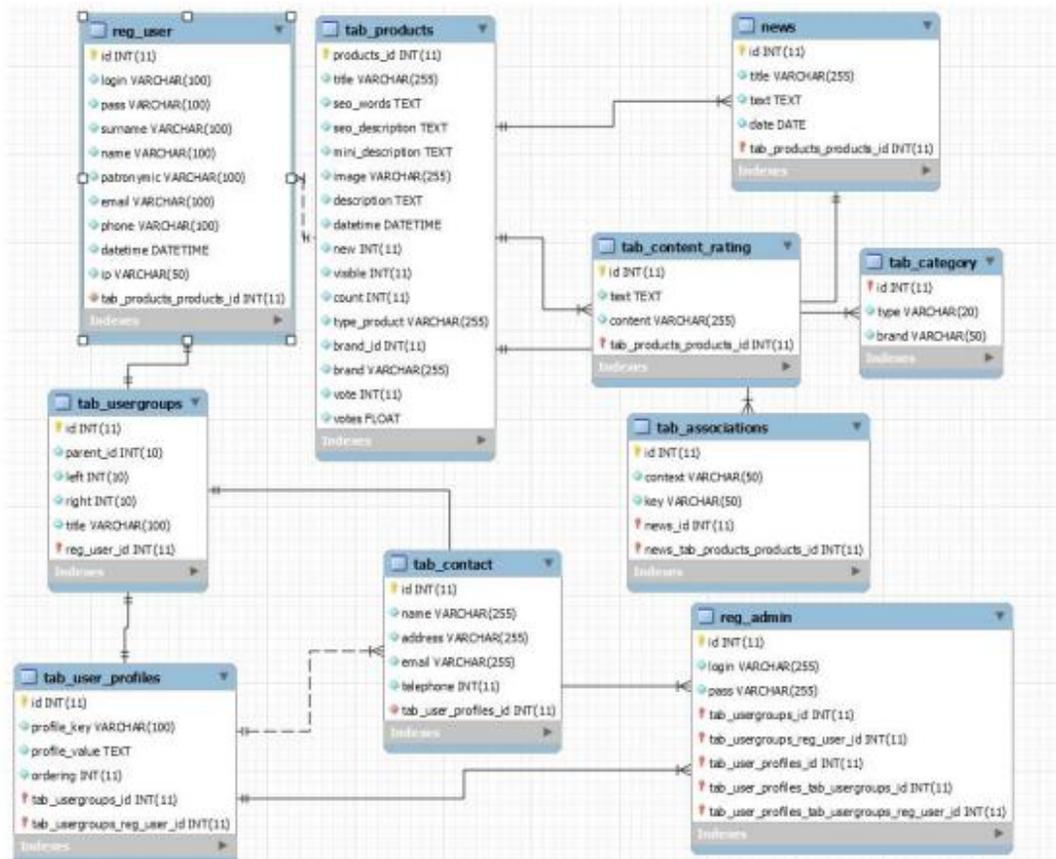
У кожній асоціації ви можете розрізнити два кінці (відповідно до паралельності асоціації) і встановити ступінь кінця зв'язку (кількість екземплярів асоціації сутності) на кожному кінці.

Моделювання зв'язків між об'єктами залежить від типу зв'язку та категорії.

Основні типи зв'язків:

- «Один до одного» (1: 1);
- "Один до багатьох" (1: ∞);
- "Багато-до-багатьох" (∞ : ∞).

Якщо всі об'єкти одного класу беруть участь у співвідношенні з об'єктами іншого класу, то він є обов'язковим таким клас, а якщо об'єкти одного класу не беруть участі в об'єктах іншого класу, то такий клас не є обов'язковим. Структура БД веб-сайту, що розробляється, показана на малюнку 2.1:



Малюнок 2.1 – Структура БД

2.2 ДАТАЛОГІЧНА МОДЕЛЬ СИСТЕМИ

База даних для веб-сайту «Автосалону» має в собі наступні таблиці:

Таблиця 2.2.1 – Контент

Ім'я поля	Призначення поля	Тип даних
Products_id	Первинний ключ	Лічильник
title	Назва	Текстовий
Mini_description	Міні опис	Текстовий
image	Посилання на зображення	Текстовий
description	Повний опис	Текстовий

Продовження таблиці 2.2.1

datetime	Дата добавлення	Дата/час
new	Новинки	Числовий
visible	Відображення на сайті	Числовий
count	Кількість переглядів	Числовий
Type_product	Тип	Текстовий
Brand_id	Номер категорії	Числовий
brand	Категорія	Числовий
vote	Відгук	Числовий

Таблиця 2.2.2 – Категорії

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
type	Тип	Текстовий
brand	Назва категорії	Текстовий

Таблиця 2.2.3 – Користувачі

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
login	Логін	Текстовий
pass	Пароль	Текстовий
surname	Прізвище	Текстовий
name	Ім'я	Текстовий
patronymic	Батьківство	Текстовий
email	Електронна пошта	Текстовий
phone	Телефон	Числовий
datetime	Дата	Дата/час

Таблиця 2.2.4 – Адміністратори

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
login	Логін	Текстовий
pass	Пароль	Текстовий

Таблиця 2.2.5 – Групи користувачів

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
Parent_id	Номер батьківської сторінки	Числовий
title	Назва	Текстовий

Таблиця 2.2.6 – Контакти

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
name	Ім'я	Текстовий
address	Адреса	Текстовий
email	Електронна пошта	Текстовий
telephone	Телефон	Числовий

Таблиця 2.2.7 – Новини

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
title	Назва	Текстовий
text	Опис	Текстовий
date	Дата	Дата/час

Таблиця 2.2.8 – Профілі

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
Profile_key	Ключ профілю користувача	Текстовий
Profile_value	Значення профілю користувача	Текстовий

Таблиця 2.2.9 – Зв'язки

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
context	Дані контактів	Текстовий
key	Ключ контактів	Числовий

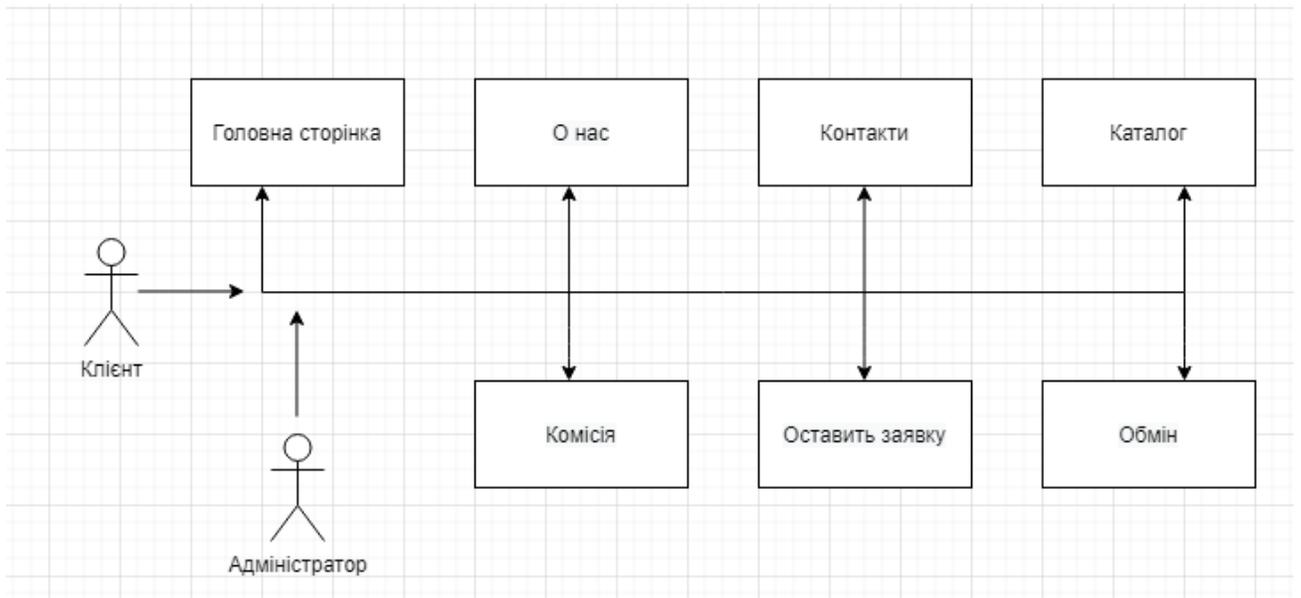
Таблиця 2.2.10 – Рейтинг

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
text	Атрибут	Текстовий
context	Значення атрибута	Текстовий

2.3 СТРУКТУРА РОЗРОБЛЮВАНОВОГО САЙТУ

Структура сайту здійснена за допомогою usecase– діаграм.

Головне меню позначено на малюнку 2.3.1



Малюнок 2.3.1 – Меню сайту

2.4 РОЗРОБКА ДИЗАЙНУ САЙТУ

Резерв (опис)

2.5 СТВОРЕННЯ БАЗИ ДАНИХ WEB-САЙТУ

База даних була створена за допомогою утиліти phpMyadmin. Щоб запустити цю утиліту, вводимо посилання `http://localhost/phpmyadmin/` в адресному рядку браузера та запускаємо пакет Denwer (запустити Denwer) (Малюнок. 2.5.1):

URL	Описание
https://subdomain.localhost/ssl.php	Проверка SSL
http://subdomain.localhost/	Проверка "не-Интернет" доменов второго уровня, а также SSI
http://test1.ru/	Проверка "Интернет"-доменов второго уровня: test1.ru (покаже отключите прокси-сервер!)
http://subdomain.test1.ru/	Проверка "Интернет"-доменов третьего уровня
http://localhost/Tests/phpnotice/index.php	Проверка перехода PHP Notice в Denwer
http://localhost/Tests/PHP5/index.php5	PHP5 information
http://localhost/Tools/phpMyAdmin	Проверка MySQL и phpMyAdmin
http://custom-host:8648	Проверка хоста с другим IP-адресом и портом (127.0.0.1:8648) <i>В Windows XP SP2 является ошибка, из-за которой данной части может не работать. Официальную "заплатку" от Microsoft можно здесь.</i>
http://localhost/Tests/sendmail/index.php	Проверка отладочной загрузки для sendmail

Малюнок 2.5.1 – Запуск утиліти phpMyadmin

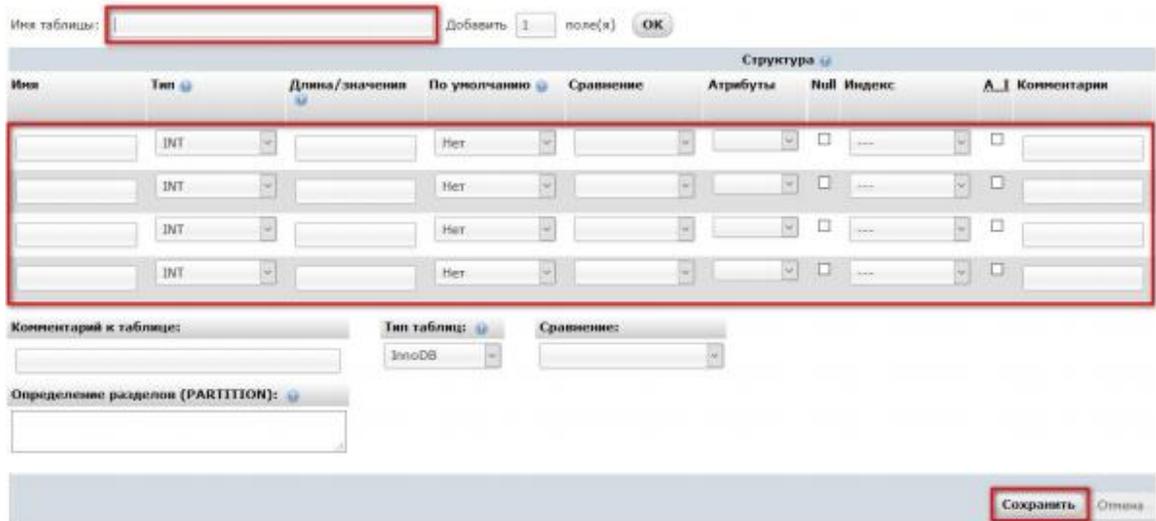
Далі, у вікні, яке з'явиться, вводимо назву бази даних та натискаємо кнопку Створити (Малюнок. 2.5.2):



Малюнок 2.5.2 – Створення бази даних

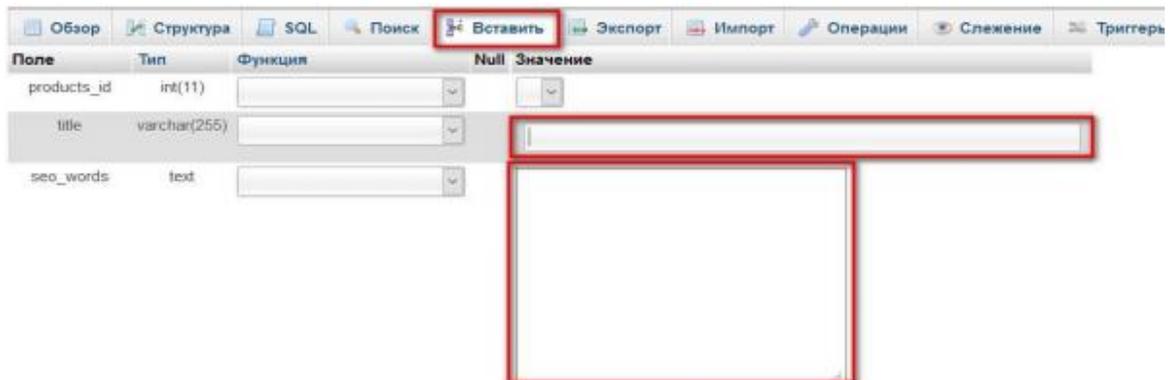
Після створення бази даних додаю таблиці, в яких буде зберігатися вся інформація, що використовується на сайті. Для цього натискаю кнопку «Створити таблицю», заповнюю ім'я та відповідні поля таблиці ідентифікатором та вибираю її тип, а потім натискаю кнопку «Зберегти» (Малюнок. 2.5.3). Кожен стовпець повинен отримати набір атрибутів, які можуть характеризувати сам стовпець та елементи, які будуть введені в цей стовпець таблиці. Іншими словами, кожному

стовпцю присвоюю ім'я, його тип, довжину та точність (якщо стовпець призначений для зберігання числових даних), а також обмеження, властиві елементам стовпця. На додаток до звичайних характеристик стовпців таблиць бази даних сховища, кожна таблиця повинна мати стовпець для ідентифікації унікальності об'єктів, зіставлених із таблицею зберігання.



Малюнок 2.5.3 – Створення таблиць бази даних

Після створення таблиці заповнюю дані. Для цього вибрано таблицю, та натискаю кнопку «Вставити», після чого з'являться безпосередньо заповнені поля (Малюнок. 2.5.4).



Малюнок 2.5.4 – Заповнення таблиці бази даних

2.6 ФІЗИЧНА СТРУКТУРА САЙТУ

Фізична структура сайту дозволяє змінювати дизайн та його функції за допомогою мови програмування php. Сайт містить такі основні документи:

- Index.php - файл домашньої сторінки сайту, містить ініціалізацію всіх компонентів сайту, завантажених структурою включення.
- New.php - файл на сторінці "Новий проект".
- About_us.php - файл сторінки "Про сайт".
- Зворотній зв'язок.php-файл форми зворотного зв'язку.
- Реєстрація.php-файл реєстраційного модуля.
- Файл модуля Search.php-Search.
- View_cat.php - файл, що використовується для відображення категорій сайтів.
- View_content.php - файл, який використовується для відображення детальної інформації про блок.
- Каталог "CSS" - Містить таблиці стилів для налаштування розташування елементів сайту.
- Каталог "Функції" - містить php-файли для використання функцій для спрощення роботи на сайті.
- Каталог "Картинки" - містить картинки, що використовуються для дизайну веб-сайтів.
- Каталог "Включити" - містить php-файли з компонентами сайту (верхній, нижній колонтитул, бічна панель, блок вмісту, модуль підключення до бази даних).
- Каталог "JS" - містить сценарії, що використовуються на сайті.
- Каталог "Reg" - містить додаткові php-файли, необхідні для реєстру.

- Каталог "Uploads_images" - містить зображення, що використовуються в базі даних.

2.7 ПРОЦЕС РОЗМІЩЕННЯ САЙТУ НА ХОСТИНГУ

Створений веб-сайт розміщують в Інтернеті на Web-сервері. Він в свою чергу виконує збереження, пошук і обмін файлами, та повідомленнями в WWW. Тобто, коли клієнт дає запит в WWW, браузер в свою чергу відправляє цей запит Web-серверу, на якому знаходяться дані. Сервер шукає дані в базі на своїй дисках і відправляє його на комп'ютер клієнта, від якого було здійснено запит. Обмін інформацією між клієнтом і Web-сервером відбувається відповідно до протоколу HTTP — загального протоколу, що відповідає за функціонування WorldWide Web.

Для вирішення цієї проблеми в виборі Web-серверу звертаються за допомогою сторонніх організацій, тобто фірми вашого провайдера чи спеціалізованих фірм, які займаються хостингом.

Хостинг— це спільне розміщення Web-сайту на обладнанні компанії. Ресурси серверу і лінії зв'язку використовуються спільно безліччю клієнтів.

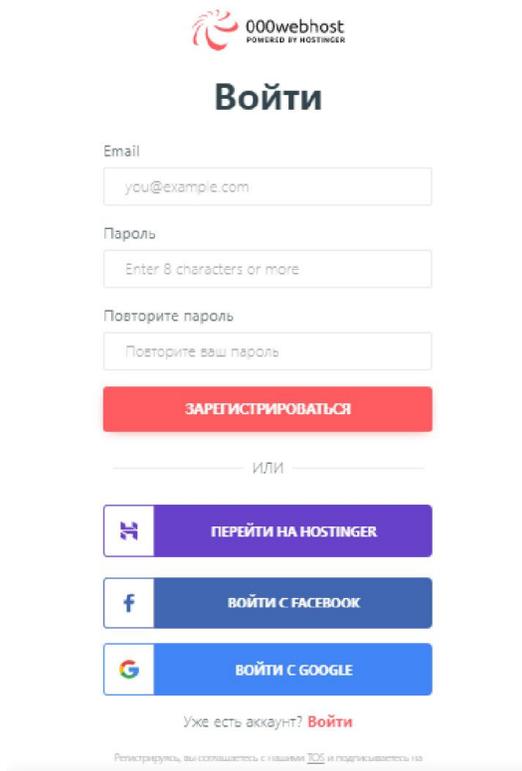
При виборі такого Web-серверу керуються двома основними критеріями, а саме ціною та продуктивністю. Мною було вибрано тарифний план безкоштовний, в такому тарифі можливості обмежені, але маю достатній простір на диску серверу – до 300Мбайт. На ньому я розміщую свій сайт автосалону.

Для цього розміщення я використовую сайт компанії hostinger.com.ua (Малюнок. 2.7.1).

Безкоштовний Хостинг	НАЙПОПУЛЯРНИШИЙ Простий Хостинг	Преміум Хостинг
Завжди безкоштовний	Найкраще рішення для початківця	Ідеальний пакет для особистих сайтів
0,00€ /місяць	47,00€ 29,00€ /місяць ЗНИЖКА 83%	33,00€ 44,00€ /місяць ЗНИЖКА 81%
Отримати безкоштовно	Додати	Додати
	59,00 €/міс. при продовжанні	79,00 €/міс. при продовжанні
<ul style="list-style-type: none"> ✓ 1 сайт ✓ 300 МБ місця на HDD ✓ ~300 Візитів/міс. 🌐 ✗ Намас email акаунтів ✗ Безкоштовний SSL ✗ Безкоштовний домен ✓ Обмежений трафік (3 ГБ) ✗ Керований WordPress ✗ Пришвидщення WordPress 	<ul style="list-style-type: none"> ✓ 1 сайт ✓ SSD-диск 30 ГБ ✓ ~10000 Візитів/міс. 🌐 ✓ 1 email акаунт ✓ Безкоштовний SSL-сертифікат (ціною 329,00 €) 🌐 ✗ Безкоштовний домен ✓ Трафік 100 ГБ ✓ Керований WordPress 🌐 ✓ Пришвидщення WordPress 🌐 ✓ Гарантія повернення коштів 30 днів 🌐 ✓ Баз даних: 2 ✓ Доступ до GIT 🌐 	<ul style="list-style-type: none"> ✓ 100 Сайтів ✓ SSD-диск 100 ГБ ✓ ~25000 Візитів/міс. 🌐 ✓ Безкоштовна пошта 🌐 ✓ Безкоштовний SSL-сертифікат (ціною 329,00 €) 🌐 ✓ Безкоштовний домен (ціною 224,00 €) 🌐 ✓ Необмежений трафік 🌐 ✓ Керований WordPress 🌐 ✓ Пришвидщення WordPress 🌐 ✓ Гарантія повернення коштів 30 днів 🌐 ✓ Необмежені бази даних ✓ Доступ до GIT 🌐 ✓ Доступ через SSH 🌐
Всі характеристики	Всі характеристики	Всі характеристики

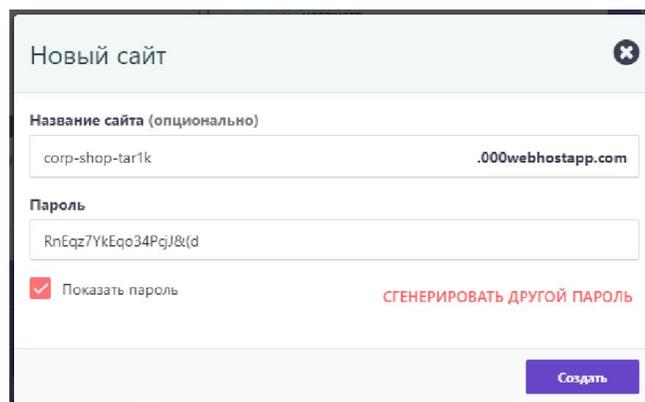
Малюнок 2.7.1 – Тарифні плани

При виборі тарифу, мене автоматично перенаправляє за посиланням на (ru.000webhost.com/besplatnaya-registraciya) , та реєструюся на хостінгу (Малюнок. 2.7.1).



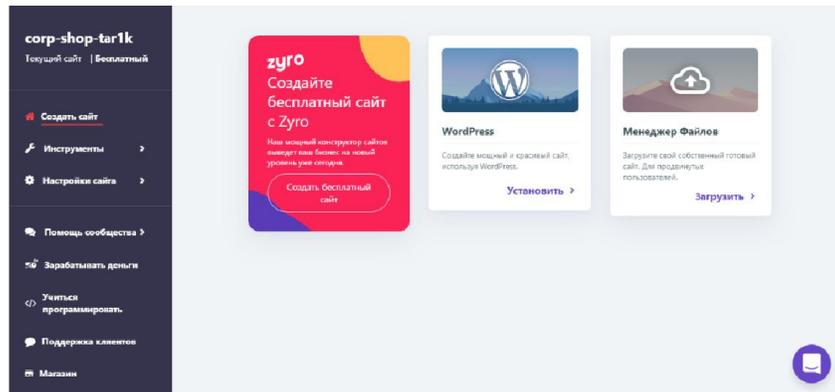
Малюнок 2.7.2 – Вікнореєстрації на хостінгу

Потім вибираю створити сайт, та записую дані майбутнього сайту (Малюнок. 2.7.3).



Малюнок 2.7.3 – Вікно реєстрації адмін панелі хостингу

Після реєстрації адмін панелі переходимо на саму панель (Малюнок. 2.7.4).

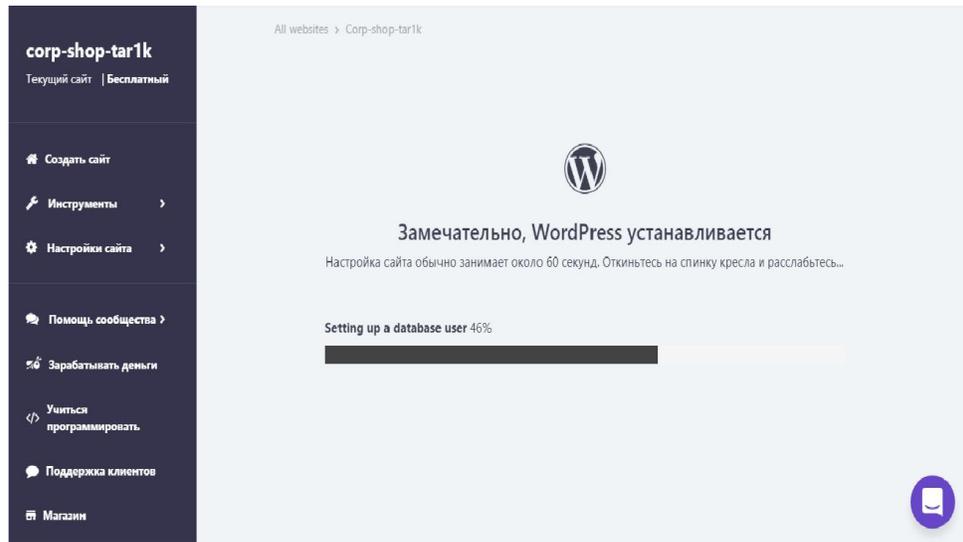


Малюнок 2.7.4 – Вікно адмін панелі хостингу

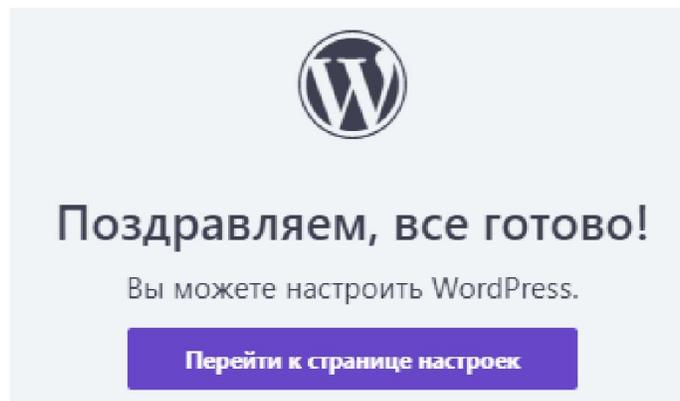
Потім вибираю для створення автосалону CMS WordPress (Малюнок. 2.7.5).

Малюнок 2.7.5 – Вікно встановлення WordPress

Етап встановлення та завершення такої системи зображено на малюнках 2.7.6 та 2.7.7

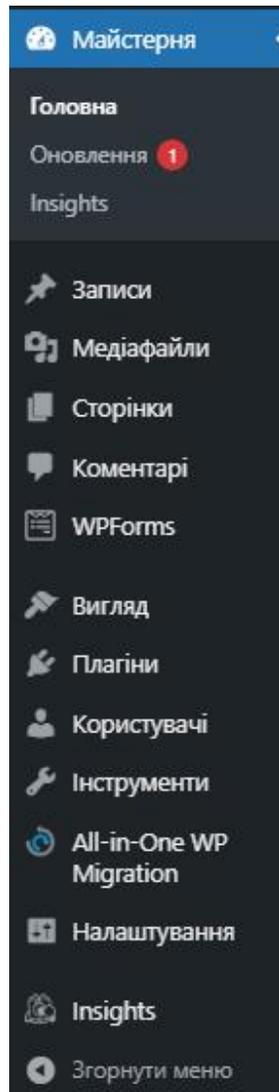


Малюнок 2.7.6 – Этап встановлення WordPress



Малюнок 2.7.7 – Завершения встановлення WordPress

Переходимо за посиланням (corp-shop-tar1k.000webhostapp.com/wp-login.php) в адмін панель нашого створеного сайту (Малюнок. 2.7.8).



Малюнок 2.7.8 – Адмін панель WordPress

РОЗДІЛ 3

ВИКОРИСТОВУВАНІ МОВИ ПРОГРАМУВАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

3.1.ВИМОГИ ДО РОЗРОБЛЮВАНОВОГО САЙТУ

Загальні вимоги до розроблюваного Web-сайту:

- висновок інформації щодо автосалону;
- розміщення контактної інформації;
- наявність вертикального і горизонтального меню;
- навігація по сайту забезпечується за допомогою текстових посилань

на його розділи;

- наявність форми зворотного зв'язку;
- наявність модуля здійснення пошукових запитів;
- наявність форми реєстрації.

Вимоги до структури сайту.

Структура розроблювального Web-сайту повинна передбачати подальший розвиток за допомогою додавання нових розділів, а також можливість їх зміни, впровадження додаткових блоків і модулів.

Вимоги до навігації сайту.

Графічний інтерфейс користувача повинен мати можливість забезпечувати наочне, інтуїтивно зрозуміле уявлення структури розміщеної на ньому інформації, забезпечувати швидкий перехід до вибраному розділу сайту. Меню навігації має бути зрозумілим для користувача, а саме: гіперпосилання на сторінки повинні мати зрозумілі заголовки, щоб користувач міг без праці перейти до потрібної йому

сторінці сайту. Для здійснення переходу між сторінками повинен бути реалізований блок, що складається зі списку гіперпосилань.

Вимоги до пошуку інформації на сайті.

Здійснення пошукових запитів повинно бути реалізовано за допомогою спеціального вікна введення запиту шляхом вибірки з бази даних. Використовувана система повинна дозволяти проводити пошук з використанням повнотекстового індексу сайту, що забезпечує морфологічний аналіз вводяться користувачем слів української мови і пошук всіх їх словоформ.

Вимоги до дизайну сайту.

Основні вимоги до дизайну сайту:

- графічні елементи сайту повинні бути розроблені з урахуванням специфіки подання інформації на Web-сайтах;
- основні елементи меню та інші елементи навігації повинні бути доступні з головної сторінки сайту;
- не повинно бути надлишкової інформації на сторінках сайту;
- перехід до обраної категорії повинен здійснюватися за допомогою бокового меню (сайдбар);
- дизайн сайту повинен бути сучасним і лаконічним;
- увагу користувача має концентруватися на обраних об'єктах.

При розробці дизайну сайту не повинні бути присутніми:

- елементи, які відволікали б увагу користувачів від перегляду інформації;
- складно структурована інформація, важка для сприйняття;
- розділи, що випадають із загального оформлення сайту.

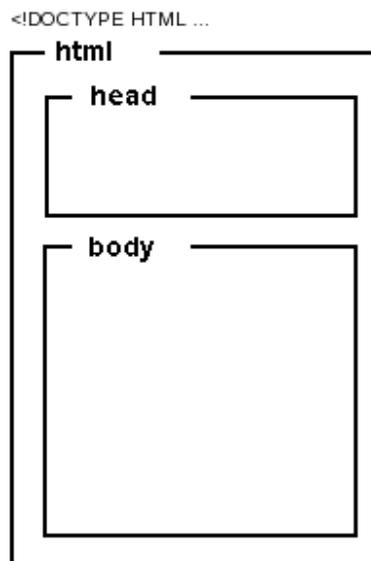
3.2 МОВИ ПРОГРАМУВАННЯ

3.2.1 Поняття HTML.

HTML - це мова розміток гіпертексту, що має набір послідовних правил розміщення та оформлення, включає в собі набір структурних елементів (тегів). HTML дозволяє описувати вміст презентацій логічних частин документів (заголовки, абзаци, списки тощо) та створювати веб-сторінки різної важкості.

Така мова включає в себе засіб структурування та форматування сторінок без прив'язки до засобів зображення. Сторінка, що складається з гіпертексту виглядає однаково на різних пристроях (крос-платформна мова).

Структура HTML-документа (Малюнок. 3.2.1) достатньо проста:



Малюнок. 3.2.1 – Структура HTML-документа

- Початок сторінки розпочинається з типу (секція DOCTYPE).
- Для опису сторінки використовують тег `<html>`. Вона в свою чергу розділена на 2 частини, а саме на теги `<head>` голову і `<body>` тіло.

- У голові (<head>) записують назву HTML-сторінки та інші параметри, за допомогою якого браузер буде зчитувати при відображенні сторінки.
- В тіло сторінки (<body>) записують по суті вміст HTML-сторінки. Тіло має в собі призначений для зображення текст і управляючу розмітку сторінки (теги), які користується браузером.

Присутність секції DOCTYPE дає право зазначати браузеру, який тип сторінки йому належить виконувати при обробці гіпертексту.

У голові розміщені метадані з описом веб-сторінки.

Секція <body> вміщує в себе те, що потрібно зобразити користувачеві: текст, мультимедійні об'єкти тощо.

DOCTYPE

Секція DOCTYPE демонструє браузеру яким буде тип документа і версію вказаної мови розмітки. А також записується назва та опис цієї мови і адреса файлу з розширенням *.dtd (documenttypedefinition).

Стандарт зобов'язує, щоб секція DOCTYPE була вказана в документі, тому як має можливість прискорити та підвищити обробку гіпертексту. браузер може не тратити часу на припущення про те, як потрібно інтерпретувати теги, а перевіряється відповідно до стандартним визначенням (файлом .dtd).

Приклади DOCTYPE наведено в додаткуВ.

Мета-теги

Мета-тег HTML – один з елементів розмітки, як такий, що описує властивості документа (метадані). Позначення мета-тега зазначається набором його атрибутів, які задані в тезі <meta>. Розміщені в блоці <head> ... </ head> веб-сторінки. Вони не завжди є обов'язковими елементами.

Приклад опису метаданих наведено в додаткуС.

Теги

Тег (html-тег, тег розмітки) - керуюча символна послідовність, яка задає спосіб відображення гіпертекстової інформації. Вони складаються з імені та можуть мати необов'язковий список атрибутів. Всі теги (разом з атрибутами) зображені в кутові дужки (<>):

<Імя_тега [атрибути]>

Зазвичай, теги є парними і поділяються з початкового та кінцевого тегів. Ім'я кінцевого тега подібне з іменем початкового, але перед іменем кінцевого ставиться коса риска / (<html> ... </html>). Кінцеві теги ніколи не можуть містити атрибутів, а також не всі теги не мають кінцевий елемент, прикладом якого є тег . Регістр символів немає значення.

Одні теги можуть бути записані всередині других, при цьому форматування внутрішнього є пріоритетним перед зовнішнім. Використанні вкладені закривають, розпочинаючи з самого останнього та рухаючись до першого:

Приклади тегів наведено в додаткуС.

Атрибути

Атрибути – доповнюють представлення заданого тегу по формулі:

<Тег атрибут = "значення"> ... </тег>

Атрибути записують в початковому тегу, та розділяють один від одного пропусками, якщо їх декілька. Значення атрибута, може бути як словом так і числом, розташовують після знаку рівності, що слідує за іменем атрибута. Немає значення в якому порядку записані атрибути. Якщо їх декілька то такі значення необхідно взяти в лапки.

Приклади атрибутів та їх значень наведено в додаткуС.

Гіперпосилання

Гіперпосилання - це такий елемент, який має властивість переходу. Такими елементами можуть бути текст або мультимедія, при натисканні якого здійснюється перехід від однієї сторінки в іншу та на певне слово у фрагменті тексту. Позначається тегом <a>. Також посилання діляться на абсолютні та відносні.

- Абсолютні виконують перехід на зовнішній ресурс.
- Відносні виконують перехід на внутрішні веб-сторінки сайту.

3.2.2 Поняття CSS.

Мова каскадних таблиць стилів використовується для опису зовнішнього вигляду web-сторінки, написаного зі спільним використанням мови гіпертекстової розмітки. Широко використовується розробниками Web-сайтів для установки шрифтів, їх розташування на сторінці, квітів, використовуваних в ході розробки, і інших принципів зовнішнього вигляду блоків Web-сторінки.

Головним завданням, що вирішується в ході створення CSS, є поділ вмісту, що, в свою чергу, дозволяє підвищити доступність документа, а також забезпечити більшу гнучкість і можливість керування поданням Web-сторінки і знизити складність структури її вмісту. Крім цього, каскадні таблиці стилів дають можливість представити одну і ту ж Web-сторінку в різноманітних стилях відображення, наприклад, екранне уявлення, читання голосом і т.д. CSS складається з набору правил. У свою чергу, кожне правило включає в себе один або декількох перемикачів, які розділені комами, і блок визначень, який полягає в фігурні дужки ({}), в складі якої є набір властивостей та їх значень [6].

При безпосередньому зображенні Web-сторінки таблиця стилів може бути взята з різних джерел. Ієрархія стилів складається з:

- авторські стилі (інформація стилів, що надається розробником сторінки);
- призначені для користувача стилі;

- стилі браузера.

На додаток до цього, каскадні таблиці стилів надають можливість працювати зі шрифтовим оформленням документа на більш високому рівні, ніж стандартний HTML-код, уникаючи при цьому обважнення сторінок графікою.

3.2.3 Поняття PHP.

PHP - це мова сценаріїв загального призначення, яка широко використовується для розробки веб-додатків. В даний час він взаємодіє з великою кількістю постачальників послуг хостингу та є однією з найкращих мов програмування для розробки веб-сайтів. Ця мова програмування призначена для веб-розробки і може бути безпосередньо вбудована в програмний код веб-сторінки.

Основна відмінність цієї мови сценаріїв від сценаріїв, написаних на C ++ або Perl, полягає в тому, що це не програма, яка генерує HTML-код, а програмний код, створений за допомогою декількох вбудованих команд PHP. Для того, щоб вбудувати PHP-код, він відокремлюється спеціальними тегамі відкриття та закриття. З допомогою цих тегів процесор PHP визначає початок і кінець частини HTML-коду, що містить команди PHP[4].

Мова програмування PHP містить багато вбудованих функцій, таких як: функції обробки файлової системи, функції обробки протоколу HTTP, функції обробки дат та часу, функції обробки рядків і масивів тощо. Кожна використовує написане програмне забезпечення PHP всі вироби містять набір спеціальних конструкцій. Такими конструкціями може бути будь-який елемент, що використовується в PHP-кодї, наприклад, цикли, оператори, функції тощо. **Головною особливістю мови** є те, що вона добре взаємодіє з усіма сучасними веб-технологіями та підтримує велику кількість сучасних веб-протоколів, а саме: протокол прикладного рівня для доступу до пошти IMAP, протокол FTP, протокол

POP, протокол SNMP тощо. Крім того, PHP дуже підходить для обробки баз даних. Більшість сучасних СУБД підтримують мову сценаріїв PHP.

Сценарій, написаний на PHP, може містити 10000 рядків коду і складатися з одного рядка - все залежить від того, яку проблему вам потрібно вирішити за допомогою цього сценарію. Початок коду PHP повинно бути розпочате спеціальними символами (тегами), він ініціює ряд дій (<?) і продовжить ці дії до останнього тегу (?>).

Для розробників PHP пропонує гнучкі та ефективні засоби захисту, які можна приблизно розділити на дві категорії: інструменти на системному рівні та інструменти на рівні програми. Механізм захисту, реалізований в PHP, контролюється адміністратором, і якщо він правильно налаштований, він може забезпечити максимальний захист. Крім того, у мові сценаріїв PHP реалізований так званий безпечний режим, за допомогою якого безпечний режим обмежує можливість використання певними типами користувачів коду. Наприклад, використовуючи безпечний режим, ви можете обмежити максимальний час, необхідний для виконання сценаріїв, і обсяг використаної пам'яті, оскільки неконтрольоване споживання пам'яті може негативно позначитися на роботі веб-сервера. Крім того, адміністратори можуть обмежити використання каталогів для перегляду користувачами та виконання PHP-скриптів, а також можуть використовувати PHP-скрипти для перегляду конфіденційної інформації на веб-сервері.

PHP включає безліч потужних механізмів шифрування. Крім того, сценарії, написані на PHP, сумісні з програмами багатьох незалежних компаній. Це полегшує інтеграцію із захищеною технологією електронної комерції [5].

Ще однією **перевагою** цієї мови є те, що вихідний текст PHP-сценарію неможливо переглянути в браузері, оскільки він компілюється безпосередньо перед відправкою запиту користувача. Як результат – таке виконання сценарію

заважає користувачам красти текст оригінального коду команди PHP. PHP є гнучким та легко налаштується для задоволення потреб розробників.

Оскільки PHP не містить коду, який фокусується на певному веб-сервері, він підходить для сучасних серверів, а саме: Microsoft IIS, NetscapeEnterprise Server, Apache, Stronghold і т.д.

3.2.4 Поняття JavaScript.

JavaScript - це мова програмування, яка використовується для управління сценаріями веб-перегляду. Головною особливістю цієї мови є те, що при її використанні ви можете змінювати властивості середовища відображення під час перегляду веб-сайту, не перезавантажуючи веб-сторінку. Наприклад, використовуючи JavaScript, ви можете замінити колір фону веб-сторінки або замінити зображення, вбудоване у веб-сторінку, а також можна створити нове вікно відображення або відобразити різні сповіщення.

JavaScript є об'єктно-орієнтованою мовою програмування, але оскільки вона використовує прототипування (швидка реалізація основних функцій для аналізу роботи системи), у порівнянні з традиційними об'єктами, є багато можливостей, пов'язаних з безпосереднім виконанням мовногокодоорієнтованого програмування мову. Крім того, мова JavaScript має перелік властивостей, властивих іншим функціональним мовам, а саме: функції як об'єкти першого типу, об'єкти як списки та анонімні функції.

Хоча JavaScript синтаксично схожий на C, він все одно має багато **відмінностей:**

- Використовувати об'єкти з можливостями самоаналізу;
- Функція як першокласний об'єкт;
- Автоматичне лиття типу;
- Анонімна функція.

Недоліки мови JavaScript такі:

- Неможливість регулювати окремі території;
- Відсутність як такого інтерфейсу;
- Використання стандартних інтерфейсів для доступу до веб-серверів та баз даних.

3.2.5 Поняття реляційної системи управління базами даних MySQL.

MySQL є однією з найнадійніших, найшвидших, якісних та відомих систем серед усіх існуючих сучасних систем управління базами даних. Основною причиною цього є те, що його вихідний код розповсюджується вільно, а інша причина полягає в тому, що MySQL є досить швидкою СУБД.

Головною особливістю MySQL є використання мови структурованих запитів-SQL як основної мови для управління базами даних, тобто створення або видалення таблиць у базі даних, вибір із бази даних та безпосереднє заповнення таблиць даними.

MySQL дуже стабільний, високошвидкісний, простий у налаштуванні та використанні. Вихідний код сервера компілюється на декількох платформах, тому СУБД MySQL є дуже підходящим рішенням для невеликих програм[7].

Основні особливості MySQL:

- Надайте необмеженій кількості користувачів можливість працювати одночасно з базою даних.
- Кількість рядків у таблиці може досягати 50 мільйонів.
- Швидке виконання команд користувача.
- Проста та ефективна система безпеки.

Ще однією **перевагою** перед іншими СУБД є те, що MySQL може використовувати мову SQL у стандарті ANSI 92, і встановлено безліч розширень стандарту, які не передбачені в жодній іншій системі управління базами даних.

Недоліком цієї СУБД є відсутність підтримки вкладених запитів, таких як `SELECT * FROM table_1 WHERE id IN (SELECT id FROM table_2)`. Крім того,

підтримка транзакцій не реалізована, а також не забезпечує підтримку тригерів та збережених процедур.

3.3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

3.3.1 Огляд та порівняння сучасних систем управління вмістом.

В даний час розроблено велику кількість готових систем управління контентом, як платних, так і безкоштовних. Всі ці системи можна розділити на три категорії:

1. Створення сторінки відповідно до вимог користувача. Системи цієї категорії працюють за таким шляхом: "Редагувати модуль → База даних → Демомодуль". Коли користувач запитує, модуль перегляду використовує інформацію з бази даних для безпосереднього заповнення вмісту сторінки. Для внесення змін до бази даних використовується модуль редагування. Недоліком цієї категорії є те, що кожного разу, коли користувач запитує сторінку, створюється сторінка, яка навантажує ресурси веб-сервера. Але за допомогою інструментів кешування можна значно зменшити навантаження на сервер.

2. Створення сторінки в режимі редагування. Системи цієї категорії діють як програми для редагування коду веб-сторінок, відтворюючи набір статичних сторінок після внесення змін. Недоліком цієї категорії є те, що в такому порядку створення сторінки взаємодія між користувачем та вмістом сайту буде втрачена.

3. Змішаний тип, або гібридний. Ця категорія поєднує в собі переваги перших двох. Системи цієї категорії також можуть працювати по шляху кешування, тобто коли механізм презентацій створює веб-сторінку один раз, а наступного разу, коли користувач запитує її, він використовує кеш для швидшого завантаження. Кеш можна автоматично оновити через певний час, або під час редагування певних частин веб-сайту, або вручну оновити за допомогою спеціальних команд. Іншим методом є збереження певних інформаційних блоків на етапі зміни структури веб-сайту та створення сторінок безпосередньо з цих блоків, коли користувач це запитує.

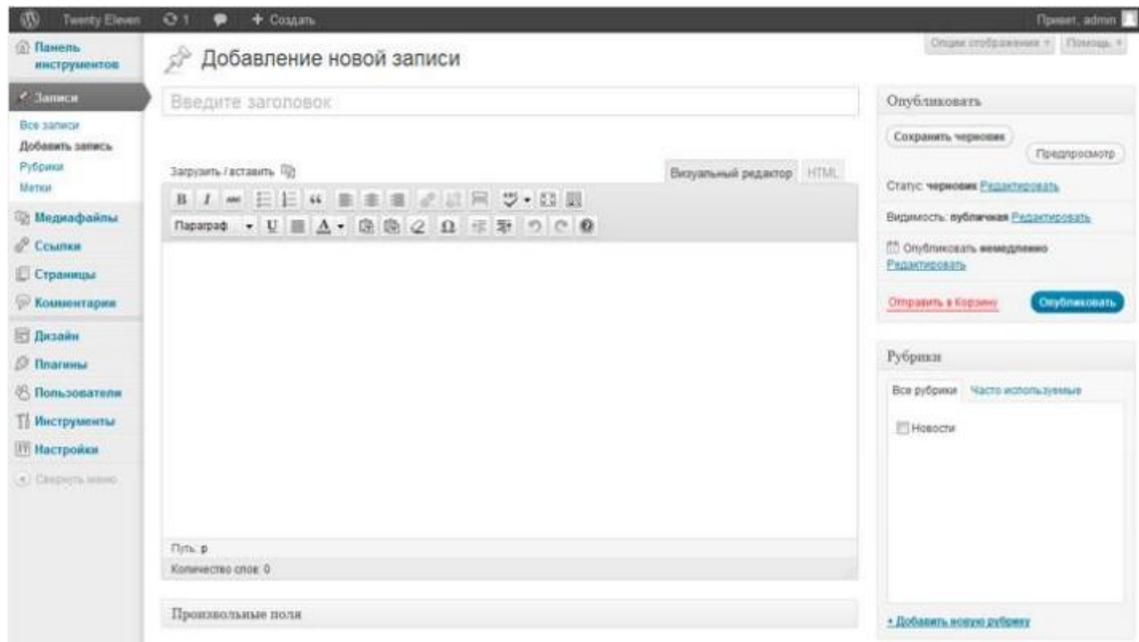
WordPress - одна з найпопулярніших безкоштовних CMS у світі. Її головне призначення - створення та реалізація веб-сайту в блозі. CMS WordPress дуже простий і простий у встановленні, майже за кілька кліків. Відразу ж потрібно звернути увагу на велику кількість існуючих тем та шаблонів дизайну для зовнішнього вигляду веб-сайту. Для створення проекту з цією системою управління вмістом не потрібні спеціальні або додаткові знання, що ідеально підходить для людей, які вперше стикаються із створенням веб-сайту.

CMS WordPress має дуже інтуїтивно зрозумілий інтерфейс панелі управління.



Малюнок 3.3.1– Панель адміністрування CMS WordPress

Для додавання матеріалів вбудований спеціальний візуальний редактор для оновлення сайту.



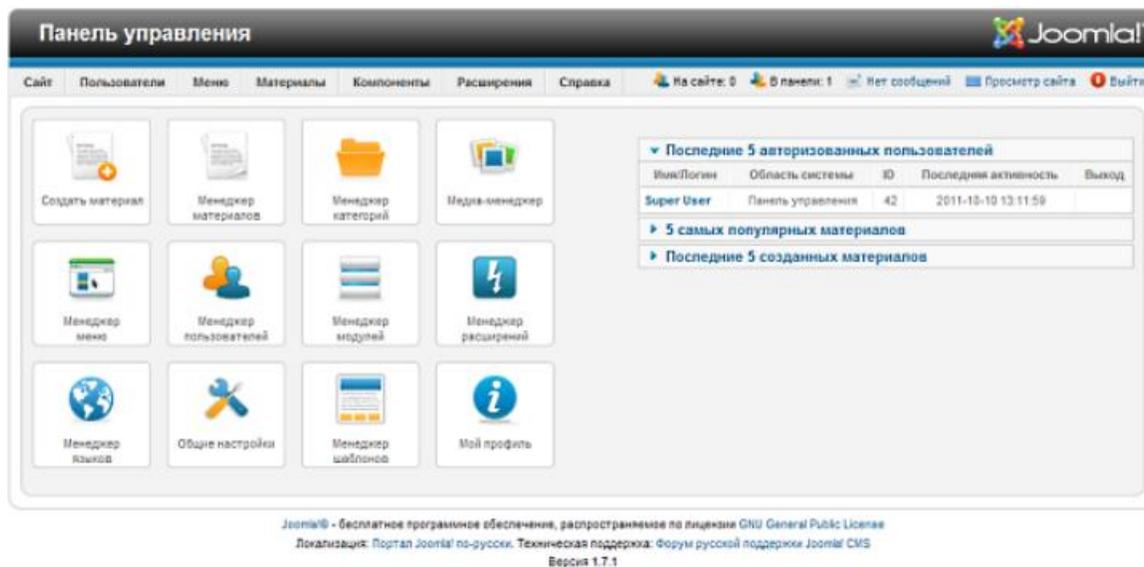
Малюнок 3.3.2 – Форма для додавання новин до CMS WordPress

Основною метою WordPress є створення та підтримка веб-сайтів блогів, але варто зазначити, що ви також можете створювати різні журнали, магазини, каталоги статей або просто сайти на цій системі управління вмістом, що однаково зручно та швидко. Оскільки існує тисяча існуючих доповнень, написаних спеціально для цієї CMS, і розроблені для значного розширення функціональних можливостей, можна використовувати різні функції.

Недоліками цієї системи можна пояснити те, що вона спеціалізується на створенні та підтримці блогів, і порівняно з іншими більш загальними системами створення сайтів з різними категоріями може бути більш складним завданням.

Joomla - це дуже популярна та гнучка безкоштовна система управління вмістом. На основі цієї системи було створено велику кількість малих та надвеликих проєктів, які мають величезні функції. Встановлення системи Joomla автоматизовано і дуже просто. Велику кількість різноманітних тем можна легко встановити з панелі управління. Всі необхідні налаштування також здійснюються

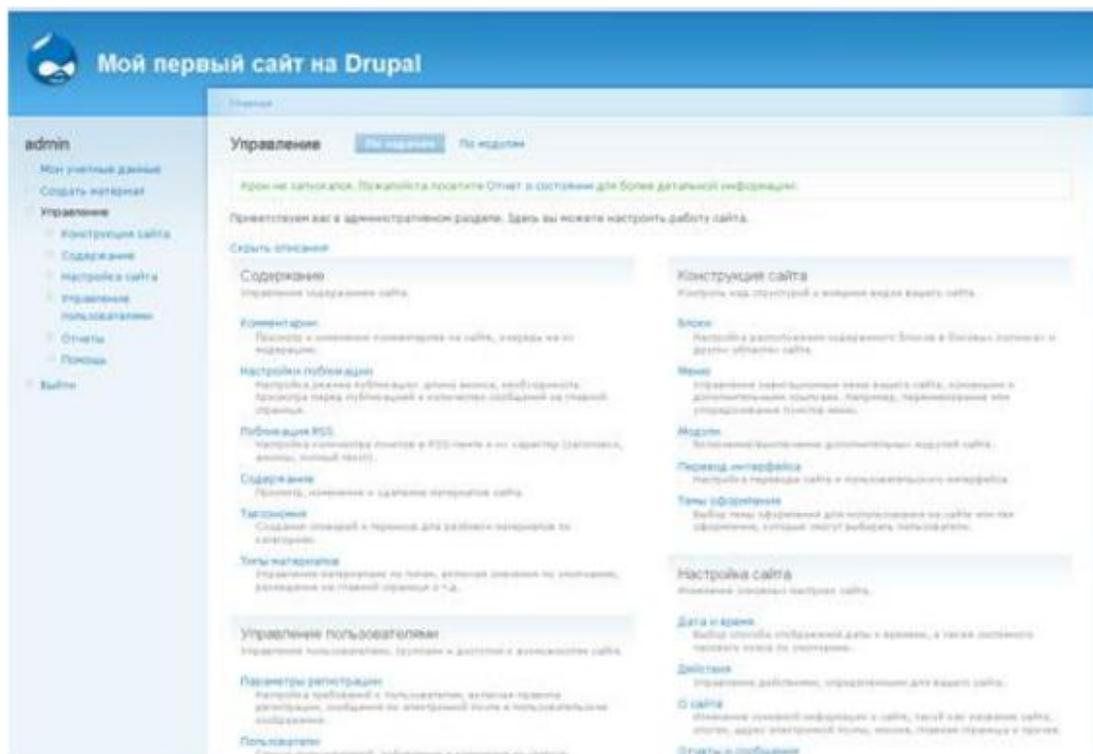
безпосередньо з панелі управління, яка має простий та інтуїтивно зрозумілий інтерфейс.



Малюнок 3.3.3 – Панель адміністрування CMS Joomla

Система управління вмістом також має велику кількість розширень, які можна легко встановити за допомогою панелі управління. Однак слід зазначити, що використання Joomla для створення високоякісного веб-сайту вимагає великих знань мови розмітки HTML та таблиць стилів CSS. Великих проблем з управлінням проектами немає. Всі необхідні операції виконуються з панелі управління. Для того, щоб реалізувати додавання нових матеріалів, використовується візуальний редактор статей та завантажувач зображень. Завдяки великій кількості існуючих розширень, Joomla дуже гнучка. Ви можете використовувати ці розширення для створення різноманітних порталів новин, відео- та фотогалерей, каталогів нерухомості, магазинів, багатомовних сайтів, соціальних мереж, дощок оголошень та інших тематичних сайтів. **Недоліки** цієї системи можна пояснити тим, що, незважаючи на всі зручності, створення проекту все ж вимагає великих знань у галузі CSS та HTML. Слід також зазначити, що дуже ретельно потрібно вибирати розширення та надавати пріоритет тим, хто має найпопулярніші, найвищі якості та найменше навантаження на сервер [8].

Drupal - це також популярна система управління вмістом для створення веб-сайтів порталів. Установка системи дуже проста і не викличе жодних проблем. Але варто зазначити, що новачкам потрібно трохи часу, щоб з'ясувати та звикнути до цієї CMS.



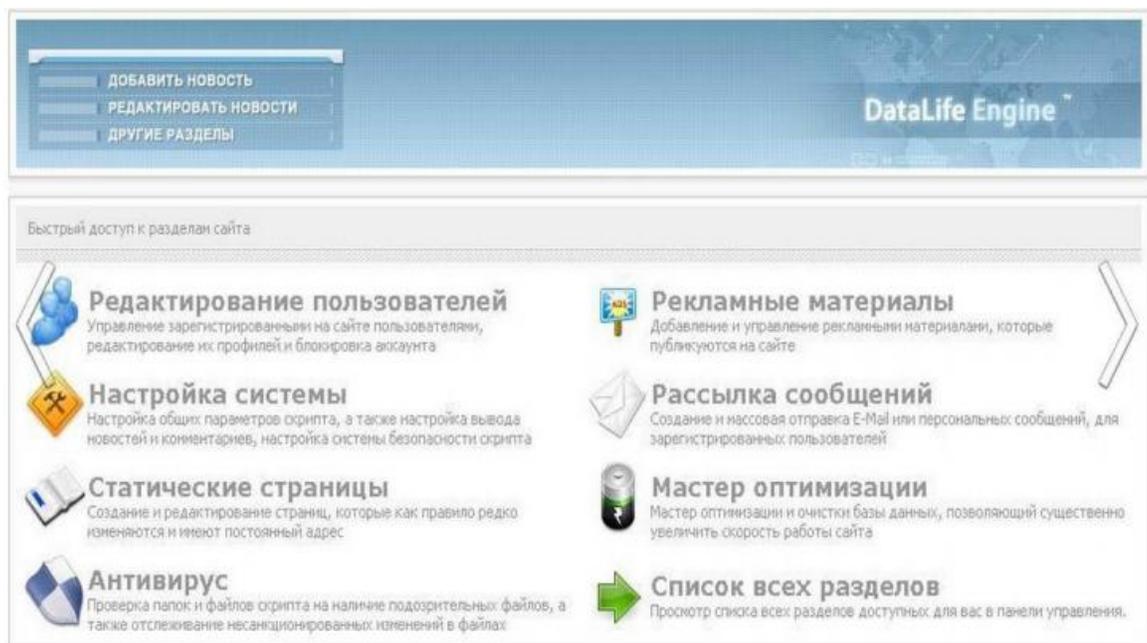
Малюнок 3.3.4 – Панель адміністрування CMS Drupal

Для інтеграції шаблонів вам потрібні деякі попередні знання з програмування. Управління системою не викликає особливих труднощів, але вимагає конкретних знань. Щоб додати матеріал на сайт, ви можете скористатися візуальним редактором, який потрібно встановити окремо, оскільки він не входить в основу цієї системи управління вмістом.

Цей базовий пакет CMS також не містить вставлених зображень чи інших зображень, але цю проблему також можна вирішити за допомогою доповнень. Слід зазначити, що Drupal - це гнучка система, яка, як і інші системи, що розглядаються, має велику кількість різних доповнень, що дозволяють створювати сайти на різні теми - соціальні мережі, галереї, форуми, новинні сайти. **Недоліки**

цієї системи, насамперед, полягають у тому, що не кожен може швидко створити якісний веб-сайт.

Система управління вмістом DLE (DatalifeEngine) найчастіше використовується для розробки різноманітних новинних ресурсів. Однак зараз випущено велику кількість різноманітних доповнень та розширень, що дозволяє використовувати DLE для вирішення будь-яких проблем. Ця система управління вмістом вважається дуже якісною, і її знає майже кожен веб-майстер. За допомогою цієї системи було створено велику кількість різноманітних ресурсів. Різниця між DLE та системою управління, про яку говорилося вище, полягає в тому, що вона не є безкоштовною, але всі інвестиції будуть цілком розумними, оскільки система управління вмістом може надати більше переваг, ніж інші системи. DLE має дуже зрозумілий та зручний інтерфейс. Жодна інша система не є такою простою, як DLE. Інтерфейс CMS надзвичайно простий у читанні та не задає зайвих питань. Для цього рівня системи найпростіший доступ до інтерфейсу.



Малюнок 3.3.5 – Панель адміністрування CMS DLE

Не менш важливо, що система управління висуває мінімальні вимоги до хостингу, що забезпечує підтримку продуктивності сайту на високому рівні за

будь-яких умов. Через цю функцію рекомендується використовувати DLE для створення новин, оскільки сайт може легко приймати тисячі відвідувачів одночасно. Що стосується функціональності, тут все також на високому рівні. Для повноцінної журналістики тут передбачені всі необхідні функції, а також додаткові розширення, що забезпечують найшвидшу публікацію статей, а також дуже просте та легке редагування даних.

За допомогою шаблонів, модулів та інструментів, розроблених системою управління DLE, ви можете створювати різні проекти: Інтернет-магазини, соціальні мережі або форуми. Але найкраще використовувати цю систему управління вмістом для створення порталу новин. До **недоліків** цієї системи можна віднести те, що система управління вмістом є платною, і хоча безкоштовна версія надає багато функцій, вона має обмеження на кількість матеріалів, які можна опублікувати.

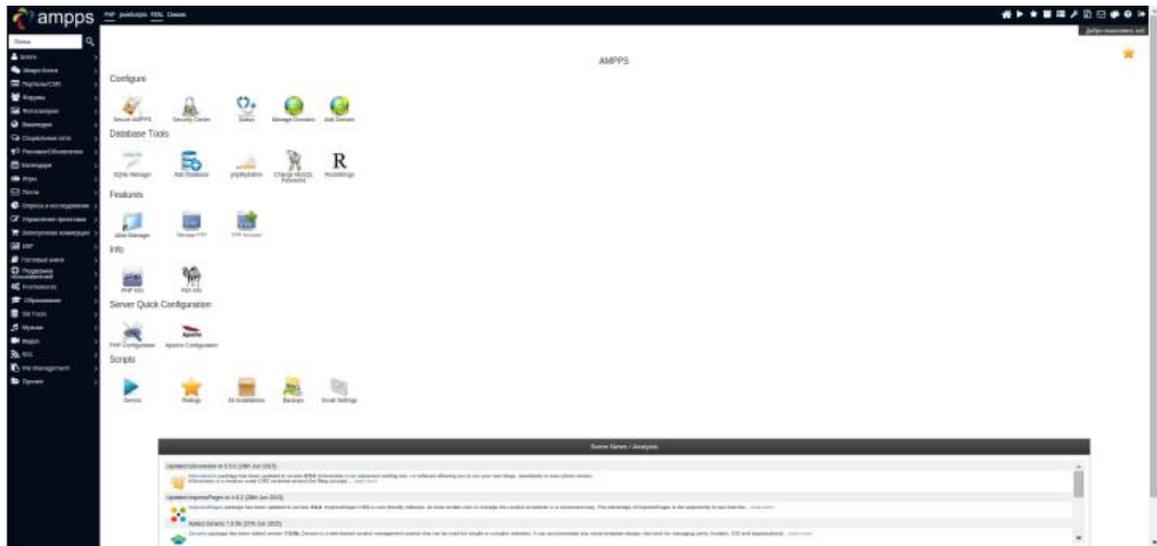
3.3.2 Огляд та порівняння локальних серверів.

Локальний сервер - це набір програм, які можуть створювати симуляції хостів безпосередньо на вашому комп'ютері та дозволяють створювати та тестувати динамічні сайти за допомогою серверних мов програмування та баз даних.

Локальний сервер AMPPS

Набір програм, які можуть створити повну бібліотеку для веб-майстрів. Кожну програму можна встановити, протестувати та використовувати для власних цілей, і ви можете зробити це дуже швидко.

Цей компонент має **переваги** перед іншими компонентами. Варто зазначити підтримку загальних систем управління вмістом, онлайн-форумів, платформ блогів, бібліотек зображень, дошки оголошень тощо. Кількість доступних програм перевищує 200.

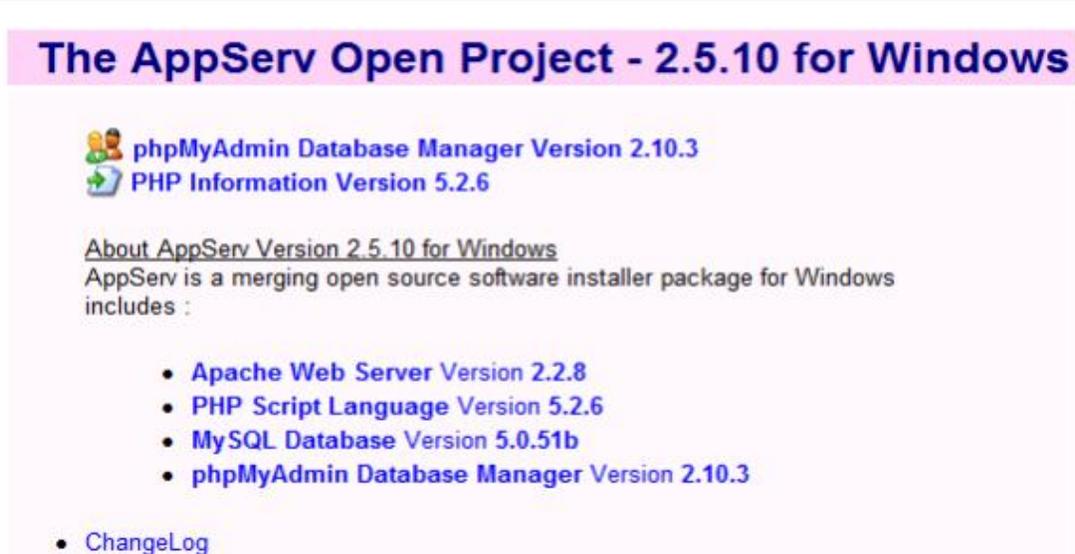


Малюнок 3.3.6 – Панель керування локального сервера AMPPS

Серед **недоліків** можна помітити частину українського перекладу, яка існує лише в каталозі скриптів. Ще одним недоліком є відсутність портативності.

Локальний сервер AppServ

Сукупність програм, встановлених як послуга. **Недоліком** є те, що немає програми управління, є лише проста сторінка дизайну та посилання на PhpMyAdmin. Ядром AppServ є графічний інстлятор Apache, PHP, MySQL, не більше того.



Малюнок 3.3.7 – Панель керування локального сервера AppServ

Локальний сервер WampServer

Вільно побудована платформа веб-розробки Windows. Асамблея включає все програмне забезпечення, необхідне для створення динамічного веб-сайту, а саме мову програмування PHP, СУБД MySQL та інше допоміжне програмне забезпечення. **Основними перевагами** WampServer є: проста установка, невеликий розподільчий пакет та стабільна робота. **Основним недоліком** WampServer є те, що WampServer вимагає певної початкової настройки.



Малюнок 3.3.8 – Панель керування локального сервера WampServer

Локальний сервер Denwer

Головною особливістю Denwer є віддалена робота над кількома незалежними проектами одночасно в комфортних умовах та можливість розміщення їх на флешці. Крім того, за допомогою емулятора веб-сервера Denwer можна створити власний хост і використовувати мову SQL для створення бази даних для веб-сайту.

URL	Описание
https://subdomain.localhost/ssl.php	Проверка SSL
http://subdomain.localhost/	Проверка "не-Интернет" доменов второго уровня, а также SSI
http://test1.ru/	Проверка "Интернет"-доменов второго уровня: test1.ru (значител отключите прокси-сервер!)
http://subdomain.test1.ru/	Проверка "Интернет"-доменов третьего уровня
http://localhost/Tests/phpnotice/index.php	Проверка перехвата PHP Notice в Денвере
http://localhost/Tests/PHP5/index.php5	PHP5 information
http://localhost/Tools/phpMyAdmin	Проверка MySQL и phpMyAdmin
http://custom-host:8648	Проверка хоста с другим IP-адресом и портом (127.0.0.2:8648) <i>В Python XP SP2 имеется ошибка, из-за которой данный хост может не работать. Официальную "заплатку" от Microsoft качайте здесь.</i>
http://localhost/Tests/sendmail/index.php	Проверка отладочной загрузки для sendmail

Малюнок 3.3.9 – Панель керування локального сервера Denwer

PHP-редактор

В якості редактора коду було обрано PhpDesigner - це платний редактор коду (+ IDE), створений спеціально для розробки веб-додатків, за допомогою нього ви можете створювати PHP, CSS та HTML-проекти та збільшувати свої переваги від вбудованої бібліотеки функцій. Також варто відзначити автоматичне налаштування виразів та частин коду, підказки коду та посилання на функції PHP [11].

phpDesigner - потужний програмний продукт, що використовується для створення, редагування, налагодження, аналізу та публікації веб-сторінок у PHP. Програма підходить не тільки для початківців програмістів, але і для професіоналів, що займаються веб-розробкою. **Видатною особливістю** PhpDesigner є те, що крім PHP, він також підтримує JavaScript, HTML, MySQL, CSS, VBScript, XML, JAVA, C, Python та Ruby. Крім того,

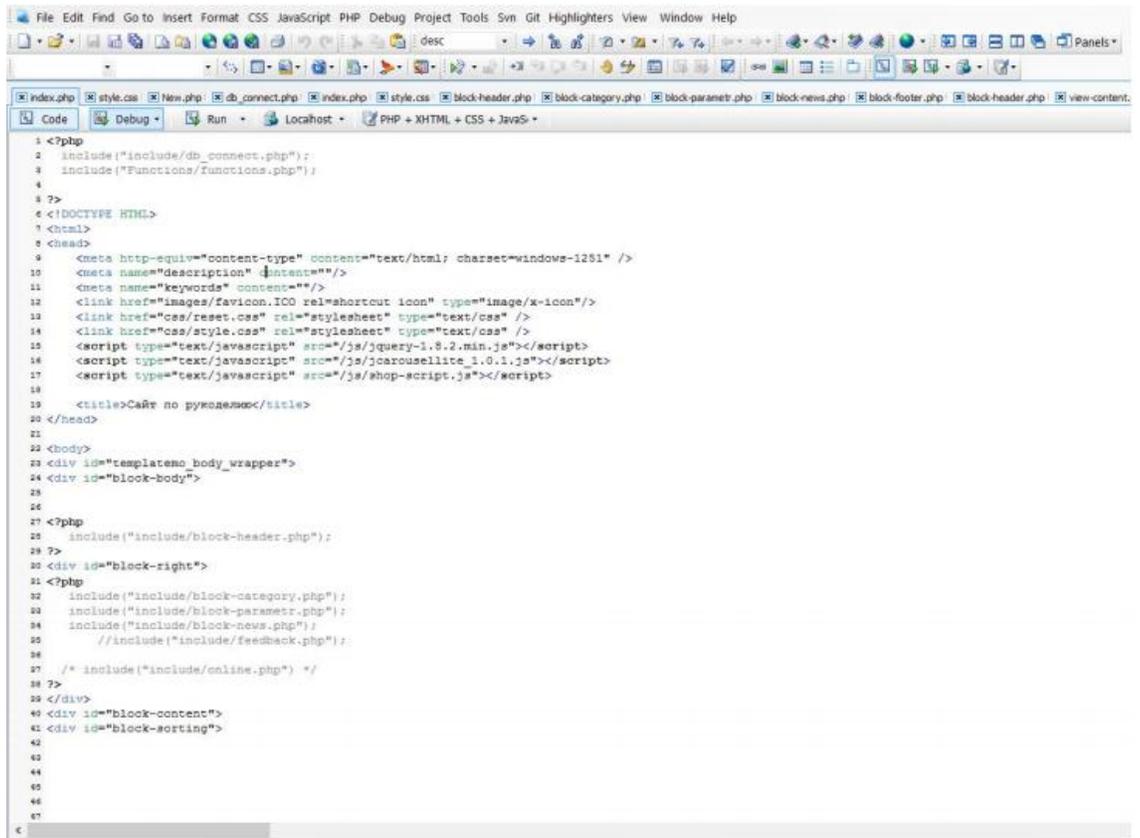
програма також містить бібліотеку великої ємності, що містить понад 3000 функцій, до яких можна легко отримати доступ під час програмування.

Основні особливості phpDesigner:

- підтримка виділення коду під час програмування;
- підтримка перевірки граматики PHP, CSS, HTML;
- підтримка автоматичного створення закриваючих тегів для PHP, CSS, JavaScript, HTML;
- підтримка бібліотеки JavaScript;
- підтримка оперативної системи під час написання коду;
- підтримка переходу до будь-якого блоку коду;
- перевірка коду підтримки.

Ви також можете використовувати локальний сервер і перевіряти результати в різних браузерах безпосередньо з інтерфейсу програми.

Основне вікно phpDesigner показано на малюнку 3.3.10:



```
1 <?php
2 include("include/db_connect.php");
3 include("Functions/Functions.php");
4
5 ?>
6 <!DOCTYPE HTML>
7 <html>
8 <head>
9     <meta http-equiv="content-type" content="text/html; charset=windows-1251" />
10    <meta name="description" content="" />
11    <meta name="keywords" content="" />
12    <link href="images/favicon.ico" rel="shortcut icon" type="image/x-icon" />
13    <link href="css/reset.css" rel="stylesheet" type="text/css" />
14    <link href="css/style.css" rel="stylesheet" type="text/css" />
15    <script type="text/javascript" src="/js/jquery-1.8.2.min.js"></script>
16    <script type="text/javascript" src="/js/jcarousellite 1.0.1.js"></script>
17    <script type="text/javascript" src="/js/shop-script.js"></script>
18
19    <title>Сайт по рукоделам</title>
20 </head>
21
22 <body>
23 <div id="template_body_wrapper">
24 <div id="block-body">
25
26
27 <?php
28 include("include/block-header.php");
29 ?>
30 <div id="block-right">
31 <?php
32 include("include/block-category.php");
33 include("include/block-parametr.php");
34 include("include/block-news.php");
35 //include("include/feedback.php");
36
37 /* include("include/online.php") */
38 ?>
39 </div>
40 <div id="block-content">
41 <div id="block-sorting">
42
43
44
45
46
47
48
```

Малюнок 3.3.10 – Головне вікно програми phpDesigner

ВИСНОВОК

У першому розділі проведено аналіз предметної області.

У другому розділі проведено проектування, вказано етапи розроблення та розміщення сайту, створено структуру і призначений для користувача інтерфейс сайту, вказана даталогічна модель системи, здійснено опис дизайну сайту, проведено аналіз засобів по створенню бази даних для Web-сайту, проведено аналіз працездатності створеного сайту, розглянуто різні методи і способи публікації різної інформації в Інтернеті.

У третьому розділі вказані вимоги, що повинен містити заданий сайт, розглянуто мови програмування, що використовуються в Web-технологіях, проведено аналіз існуючого програмного забезпечення, вказано використання програмних продуктів, здійснено огляд та порівняння локальних серверів, розглянуто та застосовувано основні правила та пропозиції щодо розробки та створення веб-сайтів;

Набуто теоретичні знання, використано практичні уміння з розроблення сайтів, із застосуванням засобів веб-програмування, було вибрано CMS WordPress.

В результаті дослідження обрано мову веб-програмування php, розроблено таблицю стилів CSS, підключено базу даних MySQL і створено сучасний веб-сайт "автосалону" з багатим вмістом.

Розроблений веб-сайт має багато відмінних особливостей:

- Сучасний дизайн;
- Гнучка система індивідуального відображення інформації;
- Сумісність браузера з веб-розробником;
- Розроблено модуль реєстрації та пошуку.
- Розроблено модуль зворотного зв'язку.

Спираючись на виконану роботу, можу впевнено сказати, що запропоноване у вступі завдання повністю вирішено. Мета кваліфікаційної, дипломної роботи досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Барысов Р. И. Постройте профессиональный сайт сами. – СПб., 2013. – 304 с.
2. Томсон Л., Веллинг Л. Разработка Web-приложений на PHP и MySQL. ДиаСофтЮП, 2013. – 672с.
3. Веллинг Л., Томсон Л. Разработка веб-приложений с помощью PHP и MySQL. - М.: Вильямс – 2014. - 848 с.
4. Дронов В.А. Разработка современных Web-сайтов. – СПб.: БХВПетербург, 2013. – 414 с.
5. Зольников Д.С. PHP 5. Как самостоятельно создать сайт любой сложности. 2 изд. – М.: ИТ Пресс, 2014. - 272 с.
6. Кожемякин А. А. HTML и CSS в примерах. Создание Web-страниц – М.: Альтекс-А, 2014. - 416 с.
7. Федорчук Д. А. Разработка WEB приложений на PHP и MySQL – СПб. : Корона-принт, 2013. – 340 с.
8. Колисниченко Д.Н. Joomla! Руководство пользователя. – М.: Диалектика, 2013. - 256 с.
9. Яргер Р.Дж., Риз, Дж. Кинг. MySQL и mSQL: Базы данных для небольших предприятий и Интернета. - СПб: Символ-Плюс, 2013. – 340 с.
10. Прохоренок Н. М. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера, 2013. – 912 с.
11. Мариков Ф. Ф. Самоучитель PHP 5. - М.: Вильямс, 2013. – 420 с.
12. Морозов Б. С. MySQL в связке с PHP. – СПб.: Корона-принт, 2014. – 310 с.
13. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. – СПб.: Питер - 2011. – 496 с.

14. Печников В.Н. Создание Web-страниц и Web-сайтов. Самоучитель. – М.: Триумф, 2013. – 470 с.

15. Актуальность строительства автосалонов.[Электронный ресурс]: статья. – Режим доступа: <https://liksinvestroy.ru/articles/Aktualnost-stroitelstva-avtosalono-vo-tekhnologii-byistrovozvodimyykh-zdaniy/>

16. Як працюють найбільші автосалони і сервіси в період карантину.[Електронний ресурс]: стаття. – Режим доступу: <https://fra.org.ua/uk/an/publikatsii/novosti/iak-pratsiuiut-naibilshi-avtosaloni-i-servisi-v-period-karantinu>

17. Введение в html.[Электронный ресурс]: статья. – Режим доступа: <https://iit-web-lectures.readthedocs.io/ru/latest/www/html.html>

ДОДАТОК А

Резерв для вихідного коду.

ДОДАТОК В

Приклади DOCTYPE:

```
<! DOCTYPE HTML PUBLIC "-// W3C // DTD HTML 4.01 Frameset // EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

Гіпертекстовий документ в форматі HTML 4.01, що містить фрейми.

```
<! DOCTYPE HTML PUBLIC "-// W3C // DTD HTML 4.01 // EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

Гіпертекстовий документ в форматі HTML 4.01 із суворим синтаксисом (тобто не використані застарілі і не рекомендовані теги).

```
<! DOCTYPE HTML PUBLIC "-// W3C // DTD HTML 4.01 Transitional // EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

Гіпертекстовий документ в форматі HTML 4.01 з нестрогим («перехідним») синтаксисом (тобто використані застарілі або які не рекомендовані теги і атрибути).

```
<! DOCTYPE HTML> Ще не стандартизоване оголошення для документів  
HTML5.
```

ДОДАТКИ С

Приклади опису метаданих:

<Head>

<Metaname = "author" content = "рядок"> - автор веб-документа

<Metaname = "date" content = "дата"> - дата останньої зміни веб-сторінки

<Metaname = "copyright" content = "рядок"> - авторські права

<Metaname = "keywords" content = "рядок"> - список ключових слів

<Metaname = "description" content = "рядок"> - короткий опис (реферат)

<Metaname = "ROBOTS" content = "NOINDEX, NOFOLLOW"> - заборона на індексування

<Metahttp-equiv = "content-type" content = "text / html; charset = UTF-8"> - тип і кодування

<Metahttp-equiv = "expires" content = "число"> - управління кешуванням

<Metahttp-equiv = "refresh" content = "число; URL = адреса"> - перенаправлення

</ Head>

Приклади тегів:

<Html> ... </ html> - контейнер гіпертексту

<Head> ... </ head> - контейнер заголовка документа

<Title> ... </ title> - назва документа (те, що відображається в заголовку вікна браузера)

<Body> ... </ body> - контейнер тіла документа

<Div> ... </ div> - контейнер загального призначення (структурний блок)

<HN> ... </ hN> - заголовок N-ного рівня (N = 1 ... 6)

<P> ... </ p> - основний текст

<a> ... - гіперпосилання
 ... </ ol> - нумерований список
 ... </ ul> - маркований список
 ... </ li> - елемент списку
<Table> ... </ table> - контейнер таблиці
<Tr> ... </ tr> - рядок таблиці
<Td> ... </ td> - елемент таблиці
 ... </ img> - зображення
<Form> ... </ form> - форма
<I> ... </ i> - відображення тексту курсивом
 ... </ b> - відображення тексту напівжирним шрифтом
 ... </ em> - виділення (курсивом)
 ... </ strong> - посилення (напівжирним шрифтом)

 - примусовий розрив рядка

Приклади атрибутів та їх значень:

style = "описаніє_стілей" - локальні стилі

src = "адреса" - адреса (URI) джерела даних (наприклад картинки або скрипта)

align = "left | center | right | justify" - вирівнювання, за замовчуванням left (по лівому краю)

width = "число" - ширина елемента (в пікселях, піках, поинтах і ін.)

height = "число" - висота елемента (в пікселях, піках, поинтах і ін.)

href = "адреса" - гіперпосилання, адреса (URI) на який буде виконаний перехід

name = "ім'я" - ім'я елемента

id = "ідентифікатор" - унікальний (в межах веб-сторінки) ідентифікатор елемента

size = "число" - розмір елемента

class = "ім'я_класу" - ім'я класу у вбудованій або пов'язаній таблиці стилів

title = "рядок" - назва елемента

alt = "рядок" - альтернативний текст