

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ  
Кафедра Комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Аліна САВЧЕНКО

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

# КВАЛІФІКАЦІЙНА РОБОТА

(ДИПЛОМНА РОБОТА, ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

“МАГІСТРА”

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ “ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ  
СИСТЕМИ ТА ТЕХНОЛОГІЇ”

**Тема:** «Система управління web-сайтом»

**Виконав:** студент групи УС-211М Янголь Василь Євгенійович

**Керівник:** доцент Моденов Юрій Борисович

**Нормоконтролер** Ігор РАЙЧЕВ

Київ – 2023

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук та технологій

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Аліна САВЧЕНКО

"\_\_" \_\_\_\_\_ 2023 р.

## ЗАВДАННЯ

на виконання кваліфікаційної роботи студента

**Янголя Василя Євгенійовича**

(прізвище, ім'я, по батькові)

- Тема роботи:** «Система управління WEB-сайтом», затверджена наказом ректора від “29” вересня 2023р. за № 1976/ст.
- Термін виконання роботи :** з 02 жовтня 2023р. по 31 грудня 2023р.
- Вихідні дані до роботи:** діючі системи управління контентом, шаблон програмного продукту системи управління, інструкції для користувачів, літературні джерела з досліджуваної проблеми..
- Зміст пояснювальної записки:** теоретичні аспекти методики та засобів оновлення web-сайту, оцінка якості сучасних систем управління оновленням web-сайтів, проектування моделі системи на основі шаблонізаторів, проектування системи управління web-сайтом.
- Перелік обов'язкового ілюстративного матеріалу:** Відмінності в підтримці між відкритими і закритими системами управління контентом», «Загальний рейтинг CMS. Характеристики систем управління», «Відображення шаблонів у браузері», «Адміністративна частина розробленої системи».

## 6. Календарний план-графік:

№ п/п	Завдання	Термін виконання	Підпис керівника
1	Аналіз і опрацювання літератури	02.10.2023 – 10.10.2023	
2	Провести консультацію з науковим керівником щодо розділів дипломної роботи	10.10.2023 – 15.10.2023	
3	Підготовка та написання розділу 1	15.10.2023 – 18.10.2023	
4	Підготовка та написання розділу 2	18.10.2023 – 24.10.2023	
5	Підготовка та написання розділу 3	24.10.2023 – 06.11.2023	
6	Оформлення пояснювальної записки	06.11.2023 – 19.11.2023	
7	Оформлення графічної частини роботи	19.11.2023 – 27.11.2023	
8	Подати дипломну роботу керівнику	11.12.2023	
9	Підготовка до захисту дипломної роботи	12.12.2023 – 20.12.2023	

7. Дата видачі завдання: 02.10.2023р.

Керівник дипломної роботи \_\_\_\_\_ Юрій МОДЕНОВ  
(підпис керівника)

Завдання прийняв до виконання \_\_\_\_\_ Василь ЯНГОЛЬ  
(підпис випускника)

## РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Система управління Web-сайтом»: 102 сторінки, 42 рисунки, 9 таблиць, 4 додатки, 7 використаних бібліографічних посилання. Список бібліографічних посилань складається з 4х найменувань.

**Ключові слова:** WEB-САЙТ, CMS, JOOMLA, NETCAT, 1С-БІТРИКС, WORDPRESS, DRUPAL, PHP, HTML, SQL, СКЕДИТОР, ШАБЛОНІЗАТОР, ZENDVIEW, СИСТЕМА УПРАВЛІННЯ КОНТЕНТОМ.

**Об'єкт дослідження:** відомі, широко використовувані системи для управління веб-сайтами.

**Предмет дослідження:** система управління з використанням існуючих шаблонізаторів.

**Мета роботи:** розробити просту у використанні, але високоефективну систему управління контентом веб-сайту з використанням HTML, MySQL, PHPMyAdmin, PHP та інших технологій для створення веб-додатків.

**Метод дослідження:** аналітичний огляд ринку CMS та порівняння за основними критеріями оцінки якості.

Потреба та обґрунтування кастомної системи полягає у простоті управління всією інформацією на сайті. Архітектура програмного забезпечення є відкритою і дозволяє подальше розширення функціоналу.

Матеріал дипломної роботи рекомендується для використання при проектуванні та розробці нових програмних продуктів, а також при аналізі вимог до програмних продуктів.

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ</b>	7
<b>ВСТУП</b>	9
<b>1. ТЕОРЕТИЧНІ АСПЕКТИ СИСТЕМИ УПРАВЛІННЯ</b>	
<b>WEB-САЙТОМ</b>	13
1.1.Огляд сучасних засобів інформаційних Web-технологій	13
1.2.Динамічний Web-сайт	19
1.3.Основи використання CMS. Створення сайтів з допомогою CMS	20
1.3.1.Призначення систем управління	20
1.3.2.Схема роботи систем управління	22
1.3.3.Класифікація систем управління	24
1.3.4.Переваги CMS	29
1.4.Огляд існуючих систем управління контентом	32
1.5.Висновки до розділу	43
<b>2. ОЦІНКА ЯКОСТІ СУЧАСНИХ СИСТЕМ УПРАВЛІННЯ</b>	
<b>WEB-САЙТОМ</b>	44
2.1.Основні вимоги до систем	44
2.2.Критерії вибору систем управління	47
2.3.Порівняння CMS на основі вимог та критеріїв	50
2.4.Висновки до розділу	56
<b>3. ПРОЕКТУВАННЯ МОДЕЛІ СИСТЕМИ НА ОСНОВІ</b>	
<b>ШАБЛОНІЗАТОРІВ</b>	57
3.1.Загальне поняття та теоретичні аспекти використання шаблонізаторів.	57
3.2.Класифікація шаблонізаторів для створення CMS	59
3.3.Текстовий візуальний редактор SKEditor	60
3.4. Вибір програмної та апаратної платформи	62
3.4.1.Вибір СУБД	62

3.4.2.Вибір мови програмування	66
3.4.3.Вибір веб-сервера	67
3.4.4.Вибір апаратної платформи	68
3.5. Висновки до розділу	71
<b>4. ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ WEB-САЙТОМ</b>	<b>72</b>
4.1.Створення веб-сайту	72
4.2.Принципи роботи системи	74
4.3.Доступ та захист до адміністративної панелі сайту	75
4.4.Інструкції по користування текстовим редактором CKeditor	76
4.5.Інформаційне наповнення та оновлення матеріалів	80
4.5.1. Створення та редакція пунктів меню	80
4.5.2. Створення та редакція основних текстових матеріалів	82
4.5.3. Створення та редакція статей	84
4.5.4. Основні налаштування та вихід з системи	86
4.5.5. Оптимізація веб-сайту через систему управління контентом	87
4.6.Висновки до розділу	89
<b>ВИСНОВКИ</b>	<b>90</b>
<b>СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ</b>	<b>92</b>
<b>ДОДАТОК А</b>	
<b>ДОДАТОК Б</b>	
<b>ДОДАТОК В</b>	
<b>ДОДАТОК Г</b>	

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Скорочення	Пояснення
<b>Apache</b>	Вільний веб-сервер, кросплатформне ПО, підтримує операційні системи Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.
<b>API</b>	Інтерфейс програмування додатків - набір готових класів, процедур, функцій, структур і констант, що надаються додатком для використання у зовнішніх програмних продуктах.
<b>ASP</b>	Технологія створення веб-додатків корпорації «Майкрософт».
<b>DHTML</b>	Спосіб створення інтерактивного веб-сайту, який використовує поєднання статичної мови розмітки HTML, що вбудовується і скриптової мови JavaScript, CSS і DOM.
<b>DOM</b>	Document Object Model - програмний інтерфейс для доступу до документів.
<b>LAMP</b>	Набір серверного програмного забезпечення, широко використовуваний у Всесвітній павутині. До набору входять: Linux, Apache, MySQL, PHP.
<b>MyISAM</b>	Одна з основних систем зберігання даних в СУБД MySQL. Вона ґрунтується на коді ISAM.
<b>MSXML</b>	Набір служб, який дозволяє додаткам, написаним на JScript, VBScript і засобах розробки Microsoft, будувати додатки для Windows, засновані на XML.

<b>MVC</b>	(«Модель-подання-контролер») – схема використання декількох шаблонів проектування, за допомогою яких модель даних програми, користувацький інтерфейс і взаємодія з користувачем розділені на три окремих компонента так, що модифікація одного з компонентів надає мінімальний вплив на інші.
<b>POP3</b>	(англ. Post Office Protocol Version 3 – протокол поштового відділення, версія 3) – стандартний Інтернет-протокол прикладного рівня, використовуваний клієнтами електронної пошти для вилучення електронного повідомлення з віддаленого сервера по TCP / IP-з'єднанню.
<b>SQL</b>	(англ. Structured Query Language – «мова структурованих запитів») – універсальна комп'ютерна мова, застосовувана для створення, модифікації й керування даними в реляційних базах даних. SQL ґрунтується на обчисленні кортежів.
<b>URL</b>	(англ. Uniform Resource Locator) – стандартизований спосіб запису адреси ресурсу в мережі Інтернет.
<b>WYSIWYG</b>	(What You See Is What You Get) – це середовище, в якому користувач відразу бачить результат своєї роботи.



## ВСТУП

Донедавна при створенні будь-якого веб-сайту веб-майстер створював певну кількість веб-сторінок і розміщував їх в Інтернеті. Якщо були потрібні зміни або доповнення, власник сайту знову звертався до веб-майстра, і той виконував необхідну роботу.

Сьогодні на зміну таким статичним сайтам прийшли динамічні, в яких кожна сторінка генерується безпосередньо в момент доступу. Інформація на всіх сторінках зберігається в базі даних, а власник сайту оснащений спеціальними інструментами для додавання нових даних або виправлення раніше внесених.

При цьому зовнішній вигляд веб-сторінок і структура їх взаємодії задаються фахівцем під час створення сайту, а введення і редагування контенту може здійснюватися людиною, яка не знайома з версткою веб-сторінок і програмуванням.

Систему, яка керує динамічним веб-сайтом, часто називають системою управління контентом або CMS, що є аббревіатурою від Content Management System (система управління контентом). Такі системи зазвичай використовуються для зберігання та публікації широкого спектру контенту. Вміст веб-сайту, тобто контент, обробляється CMS і відображається відвідувачеві веб-сайту в певній формі.

CMS-системи дозволили знайти компроміс між низькою гнучкістю статичних і високою вартістю динамічних веб-сайтів і, як наслідок, підвищили зручність управління і знизили витрати на адміністрування ресурсів.

Система управління контентом – це автоматизований редакційний комплекс, який використовується для управління контентом і структурою веб-ресурсу.

Через обмеження, що накладаються на логічну структуру контенту, дизайн і функціональність динамічних веб-сайтів, використання таких систем різко знижує трудомісткість розробки й підтримки.

За допомогою таких систем досвідчений користувач може легко створити базовий веб-проект (без додаткових трудовитрат), а веб-інтегратор – реалізувати масштабний проект за розумний проміжок часу (модульність таких систем розширює їх функціональність).

У часи, коли системи управління контентом практично не використовувалися, розробка нового веб-проекту передбачала створення коду розмітки для кожної сторінки або складне програмування та інтеграцію графічного дизайну в кожну сторінку. Цей процес займав величезну кількість часу, а результат виконував обмежену кількість функцій і важко піддавався оновленню.

Тому сьогодні вибір системи управління контентом є досить важливим кроком, якому слід приділити належну увагу. Існує чотири групи CMS-систем: комерційні системи вартістю від кількох сотень до кількох тисяч доларів, недорогі комерційні системи вартістю близько ста доларів або трохи більше, безкоштовні CMS-системи з відкритим вихідним кодом та системи, які використовуються лише розробниками.

Дорогі комерційні системи можна використовувати для створення складних веб-сайтів для великих комерційних організацій, але для малого та середнього бізнесу їхня ціна часто занадто висока. Крім того, дорогі системи часто занадто складні і потребують постійної підтримки фахівців, а також мають вищі вимоги до хостингу. Як наслідок, підтримка веб-сайтів, як і їх створення, виявляється досить дорогим задоволенням. Важливо пам'ятати, що десятикратна ціна не означає десятикратної кількості функцій, і часто дорога система не набагато випереджає свої дешеві аналоги.

У більшості випадків дешеві CMS не можуть конкурувати з безкоштовними системами. Щоб створити повноцінну систему, потрібно багато працювати, і спільнота вільної системи, яка налічує тисячі користувачів, може зробити більше, ніж кілька розробників, які створюють систему на продаж. Якщо є велика команда розробників, що працює над комерційною системою, продавати її за сотню доларів не вигідно. Звичайно, якщо буде створено систему, яка буде настільки хорошою,

що більшість веб-майстрів будуть її купувати, можна встановити і низькі ціни, але доки такої системи не створено, і немає жодних ознак того, що вона буде створена в найближчому майбутньому. Фактично, до цієї ж категорії можна віднести системи, якими користуються тільки розробники. Навряд чи можливо виділити достатньо вільного часу під час веб-розробки, щоб створити якісну CMS.

У будь-якому випадку, взявши за основу будь-яку безкоштовну систему і модифікувавши її, можна отримати кращий результат при відносно невеликих зусиллях.

При створенні сайтів для малого та середнього бізнесу рекомендується використовувати безкоштовні CMS-системи з відкритим кодом. Ці системи можна модифікувати та налаштовувати відповідно до потреб. До речі, суттєвим недоліком комерційних систем є неможливість за потреби змінити основний код системи, тобто відсутність гнучкості. Звичайно, для модифікації систем з відкритим кодом потрібен певний обсяг знань, але в результаті буде створено якісний веб-сайт за короткий час і за низьку ціну.

Хоча в Інтернеті можна знайти сотні безкоштовних CMS, чітке визначення завдань для сайту допоможе значно звужити коло. Наприклад, якщо створюється блог, рекомендується для цього використовувати WordPress.

Якщо ж потрібно створити сайт на основі стандартного шаблону і розмістити його в Інтернеті, можна скористатися системою UcoZ. Однак суворі обмеження в цій системі не дозволяють створювати будь-які сайти, які подобаються, тому вона не рекомендується в більшості випадків.

Серед досить популярних систем можна виділити Drupal, Joomla, Wordpress, кількість користувачів яких в останні роки стрімко зростає, і вибір тієї чи іншої в кінцевому підсумку залежить від особистих уподобань.

На питання «Яка CMS краще?» не можна відповісти однозначно. Відповідь залежить від ряду факторів, головним з яких є цілі і завдання, поставлені при створенні сайту, необхідний функціонал, а також готовність боротися з труднощами освоєння тієї чи іншої CMS.

Якщо потрібно налаштувати сайт швидко і просто, для більшості сайтів підійдуть Wordpress або Joomla. Wordpress підійде для блогу, простого інформаційного сайту, а Joomla – для порталу з різними додатковими функціями.

Для відповідального завдання – створення комерційного сайту, складного проекту з великою кількістю відвідувачів, можна вибрати комерційну CMS, вона буде зручніше і надійніше, хоча складний комерційний сайт можна зберегти і зробити на Joomla.

Рішення про використання CMS стає все більш популярним, що в основному пов'язано зі зростаючими потребами суспільства, бажанням мінімізувати відходи при регенерації інформаційного контенту і швидко вносити зміни в існуючі ресурси.

Метою даної дипломної роботи є дослідження систем управління веб-контентом та розробка власної системи під функціонал конкретного веб-сайту.

Виходячи з поставленої мети, були поставлені наступні завдання:

1. Проаналізувати наявні відомі CMS системи для управління динамічними веб-сайтами.
2. Визначити основні критерії для порівняння різних CMS систем.
3. Виявити переваги й недоліки систем управління за визначеними критеріями.
4. Розробити власну систему для конкретного веб-сайту, обґрунтувати її переваги та зручність.
5. Написати інструкцію з використання та експлуатації.

Впровадження запропонованої системи дозволить будь-якому користувачеві, без навичок програмування та дизайну, безперешкодно оновлювати інформацію на сайті, змінювати структуру та вдосконалювати інтернет-проект відповідно до власних побажань.

## РОЗДІЛ 1

### ТЕОРЕТИЧНІ АСПЕКТИ СИСТЕМИ УПРАВЛІННЯ WEB-САЙТУ

#### 1.1 Огляд сучасних засобів інформаційних Web-технологій

Об'єктивною реальністю сьогодення є широке впровадження сучасних інформаційних технологій у сфері життєдіяльності особистості, суспільства і держави та реалізація на їх основі локальних і глобальних інформаційних систем для прискорення обміну інформацією та доступу до різноманітних інформаційних ресурсів.

Розглядаючи рівень розвитку інформаційних технологій та телекомунікацій як один з критеріїв рівня розвитку країни та інструмент інтенсифікації всіх процесів у суспільстві, Україна, як і будь-яка інша країна, зацікавлена в розвитку своєї інформаційної інфраструктури, в тому числі шляхом використання можливостей, які надають глобальні мережі передачі даних в усіх сферах економічної діяльності.

Інтернет є надзвичайно привабливим засобом комунікації, що може пояснити його безпрецедентне зростання. Новітні засоби комунікації дозволили об'єднати різні системи зв'язку в глобальну мережу. В результаті люди отримали можливість обмінюватися інформацією по всьому світу, незалежно від кордонів і відстаней. Інтернет є однією з ознак переходу суспільства від індустріальної до інформаційної стадії розвитку.

За більш ніж 30 років Інтернет перетворився з секретної, закритої технології, що використовувалася військовими організаціями та навчальними закладами, які працювали на підтримку військових і розвідувальних програм, на загальнодоступну, відкриту технологічну систему, яка почала впливати на розвиток всієї планети.

Кафедра КІТ (47)				НАУ 23.26.51 000 ПЗ			
<i>Виконав</i>	<i>Янголь В.Є.</i>			<b>ТЕОРЕТИЧНІ АСПЕКТИ СИСТЕМИ УПРАВЛІННЯ WEB- САЙТУ</b>	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Моденов Ю.Б.</i>					13	31
<i>Консульт.</i>					<b>УС-211М 122</b>		
<i>Н-контрол.</i>	<i>Райчев І.Е.</i>						

На сьогоднішній день міжнародна інтернет-політика зосередила свої зусилля на чотирьох пріоритетних завданнях:

- покращення підтримки бізнесу, в сенсі надання гарантій для створення належних умов для розвитку ринку нових товарів та послуг;
- інвестиції в майбутнє, в сенсі збільшення фінансування освіти;
- перенесення уваги на людський фактор для адекватного відображення наслідків розвитку інформаційного суспільства на регіональному рівні;
- Встановлення глобальних правил поведінки в усьому світі для правильного розуміння, впровадження та розвитку глобальної економіки.

Основними інструментами веб-технологій є HTML, CSS, DHTML, JavaScript, PHP, ASP, XML, XSLT, Ajax.

Всесвітня павутина складається з веб-сторінок, створених мовою HTML (*HyperText Markup Language*, «мова розмітки гіпертексту»).

HTML – це фундаментальна, основоположна технологія Інтернету. HTML – це не мова програмування, а мова розмітки тексту, яка використовує спеціальні оператори – *теги* або *дескриптори* – для розмітки текстового документа. Ці теги вказують, як текст або інший елемент буде відображатися у вікні браузера. HTML дозволяє створювати текстові блоки на веб-сторінці, додавати до них зображення, організовувати таблиці, керувати передачею кольору, додавати звук до дизайну веб-сторінки, організовувати гіперпосилання на інші розділи сервера або доступ до інших веб-ресурсів, а також пов'язувати всі ці елементи між собою. Документи, створені за допомогою HTML, мають лише розширення *.htm* або *.html*. Однією з основних функціональних можливостей HTML, від якої він і отримав свою назву, є гіперпосилання. Гіперпосилання – це основний функціональний елемент HTML-документа, який реалізує зв'язок певного об'єкта веб-сторінки з іншим об'єктом. Гіперпосиланням може бути фрагмент тексту або графічний об'єкт, а саме гіперпосилання може створюватися як між об'єктами на одній сторінці, так і між об'єктами на різних сторінках в Інтернеті.

HTML – це лише інтерпретована мова, тому для виконання коду її не потрібно компілювати. Інтерпретатор мови вбудований у браузер і «компілює» код безпосередньо при відкритті документа. Якщо в кодї сторінки виявлено помилку, інтерпретатор зазвичай не видає попередження, а просто ігнорує весь «помилковий» рядок, що може зіпсувати зовнішній вигляд завантаженої сторінки. Це важливо для розробників, тому слід бути уважним при написанні HTML-коду і ретельно тестувати результати роботи.

CSS (*Cascading Style Sheets – каскадні таблиці стилів*) – це технологія опису зовнішнього вигляду документа, створеного за допомогою HTML, XML і XHTML. За допомогою CSS задаються кольори, шрифти, розташування елементів сторінки тощо. До появи CSS зовнішній вигляд веб-сторінок визначався безпосередньо в HTML-кодї сторінки. Однак з появою CSS стало можливим принципово розділити зміст і представлення документа. Це нововведення дозволило легко застосовувати один стиль оформлення до кількох сторінок сайту і швидко змінювати цей дизайн.

DHTML (*Dynamic HTML*) – це набір інструментів, які реалізують інтерактивність веб-сторінки без доступу до сервера. Це означає, що певні дії відвідувача можуть викликати зміни зовнішнього вигляду і вмісту сторінки. DHTML базується на моделі DOM (*Document Object Model*), яка розширює традиційний статичний HTML-документ. DOM забезпечує динамічний доступ до вмісту, структури та стилів документа. У DOM кожен елемент веб-сторінки є об'єктом, який може змінюватися. DOM не визначає нові теги або атрибути, а лише надає можливість програмно керувати всіма тегами CSS, атрибутами і каскадними таблицями стилів.

JavaScript – це мова для написання сценаріїв для інтерактивних HTML-сторінок. JavaScript не пов'язана з мовою Java. Java була розроблена компанією SUN, а JavaScript – компанією Netscape Communication Corporation. Початкова назва була «LiveScript», але через високу популярність Java назву «LiveScript» було змінено на «JavaScript» з комерційних міркувань. JavaScript не призначений для розробки автономних додатків. Код JavaScript записується безпосередньо в текст

HTML-документа і інтерпретується браузером при завантаженні документа. За допомогою JavaScript можна динамічно змінювати текст завантаженого HTML-документа і реагувати на події, пов'язані з діями відвідувача або зміною стану документа чи вікна. Важливою особливістю JavaScript є його об'єктна орієнтація. Програміст має доступ до низки об'єктів, таких як документи, гіперпосилання, форми, фрейми тощо. Об'єкти характеризуються описовою інформацією (властивостями) і можливими діями (методами).

PHP (*Personal Home Page*) – це скриптова мова, яка вже давно переросла свою назву. Перша версія PHP була створена Расмусом Лердорфом у 1994 році і була інструментом для відстеження поведінки відвідувачів веб-сайтів. З часом PHP перетворився з інструментарію на повноцінну мову програмування, а його назву було змінено на PHP *Hyper Text Preprocessor*.

PHP – це серверна мова. Конструкції PHP, вставлені в текст HTML, виконуються сервером при кожному відвідуванні сторінки. Результат обробки цих конструкцій передається браузеру разом із звичайним HTML-текстом.

Основними конкурентами PHP є ASP (*Active Server Pages*) від Microsoft та ColdFusion від Allaire. У порівнянні з ними PHP має багато переваг, зокрема:

- Висока продуктивність. PHP-додатки працюють швидше, ніж ASP.
- Функціональність. Розробку PHP-додатків можна відокремити від власне створення сайту, що спрощує роботу як розробників, так і дизайнерів.
- Ціна. Мова PHP є абсолютно безкоштовною.
- Простий у використанні. Програмісти з досвідом програмування на популярних мовах швидко зрозуміють синтаксис PHP.
- Переносимість. Один і той самий PHP-код можна використовувати як на Windows, так і на UNIX платформах.

ASP (*Active Server Pages*) – це технологія, схожа на JavaScript і PHP. Для того, щоб створити інтерактивний веб-сайт за допомогою макромови ASP, в код необхідно вбудувати відповідний скрипт, який віддалено нагадує Java і C#. Скрипт інтерпретується і виконується безпосередньо на сервері, після чого браузеру



надсилається готовий HTML-документ з результатами роботи ASP-скрипта. У випадку сторінок, що містять конструкції ASP, не має значення, яке програмне забезпечення встановлено на комп'ютері користувача, але першочергове значення має тип сервера, який повинен підтримувати технологію.

XML фокусується на даних. Тут структурна розмітка даних і їхнє представлення строго розділені. Основні причини створення XML:

- Спроба надати потужні інструменти форматування та структурування даних для широкого кола розробників.
- Потреба в стабільній реалізації мови структурування документів, в якій допоміжні інструменти можуть бути легко створені й доступні для звичайних користувачів.

XML – це метамова, тобто спеціальна мова, за допомогою якої можна повністю описати клас інших мов, якими створюються документи. XML містить набір правил, які дозволяють створювати унікальні додатки та підмножини даних.

Багато розробників мають певні труднощі з абстрактною природою XML і довільним використанням його методів. Насправді, XML є досить логічною і добре структурованою структурою, з чітким синтаксисом, який змушує суворо дотримуватися певних правил. Вивчення XML слід починати із застосування знань про HTML. XML, як і HTML, використовує теги та атрибути. Як мова розмітки веб-документів, XML має свої особливості:

- Гнучкість. XML дозволяє обробляти унікальні дані і, незалежно від характеру даних, надає відповідні методи для їх зберігання та обробки.
- Персоналізація. Гнучкість XML безпосередньо пов'язана з можливістю визначати користувацькі дескриптори, які потрібні в процесі вирішення проблем.
- Узгодженість. XML відрізняється узгодженістю синтаксису та строгою структурою.

Майже всі сучасні браузері підтримують XML. Він може повністю замінити HTML як спосіб розмітки веб-сторінок, хоча вивчити XML складніше, ніж HTML.

Мова XSLT (*eXtensible Stylesheet Language Transformations*) – це транслятор, який дозволяє вільно змінювати вихідний текст. XSLT відіграє ключову роль у становленні XML як універсальної мови для зберігання та передачі даних. XSLT широко використовується у всьому, від електронної комерції до бездротової всесвітньої павутини.

Фактична збірка результуючого документа відбувається, коли вихідний документ і таблиця стилів XSLT передаються синтаксичному аналізатору XSLT (XSLT-процесору).

При використанні XSLT в Інтернеті синтаксичний аналіз може виконуватися на стороні користувача (в браузері) або на стороні сервера.

XSLT-перетворення базується на шаблонах. XSLT-процесор аналізує вихідний документ і намагається знайти відповідний XSL-шаблон. Якщо такий шаблон знайдено, виконуються інструкції, що містяться в ньому.

Ajax (*Asynchronous Javascript And XML*) – це підхід до розробки веб-додатків з використанням наступних технологій:

- Уніфікована презентація з використанням XHTML та CSS.
- Динамічне відображення та взаємодія з користувачем за допомогою DOM.
- Обмін та обробка даних у форматах XML та XSLT.
- Широке використання мови сценаріїв JavaScript.
- Асинхронні запити з використанням об'єкта XMLHttpRequest.

У стандартному веб-додатку всю інформацію обробляє сервер, а браузер відповідає лише за взаємодію з користувачем, надсилання запитів і відображення отриманих даних у форматі HTML. В Ajax-додатку між користувачем і сервером з'являється ще один посередник – програмний движок Ajax. Він визначає, які запити можуть бути оброблені на стороні клієнта, а які повинні бути виконані на сервері.

## 1.2 Динамічний веб-сайт

Веб-сайт – це інформація, представлена в певній формі, яка знаходиться на веб-сервері і має власне ім'я (адресу). Для перегляду веб-сторінок на комп'ютері користувача використовуються спеціальні програми – браузері. Веб-сайт складається з пов'язаних між собою веб-сторінок. Веб-сторінка – це текстовий файл з розширенням \*.html, який містить текстову інформацію і спеціальні команди – HTML-коди, що визначають спосіб відображення цієї інформації у вікні браузера. Вся графічна, аудіо- та відеоінформація не міститься в самій веб-сторінці, а представлена окремими файлами з розширеннями \*.gif, \*.jpg (графіка), \*.mid, \*.mp3 (аудіо), \*.avi (відео). HTML-код сторінки містить лише вказівники на такі файли.

Кожна сторінка веб-сайту також має власну веб-адресу, яка складається з адреси веб-сайту та імені файлу, що відповідає цій сторінці. Таким чином, веб-сайт – це інформаційний сайт, що складається з взаємопов'язаних гіпертекстових документів (веб-сторінок), розміщених на веб-сервері та має індивідуальну адресу.

Обмін інформацією в Інтернеті здійснюється за допомогою протоколів прикладного рівня, які реалізують певний прикладний сервіс. Однією з найпопулярніших веб-служб є Всесвітня павутина (WWW), заснована на протоколі передачі гіпертексту (HTTP). У гіпертекстових документах, представлених у Всесвітній павутині, користувачі не обмежені одним документом, і вони не обмежені одним комп'ютером.

Динамічний веб-сайт – це веб-сайт, сторінки якого генеруються за допомогою програмного коду. Однак він може містити окремі статичні сторінки, флеш-заставки, анімацію тощо.

Позитивні моменти:

- динамічний веб-сайт – це «живий веб-сайт», що, безсумнівно, є його найважливішою перевагою;
- легке управління сайтом через інтерфейс адміністратора;

- мінімальні витрати на управління інформацією;
- миттєве відображення введених даних (створення нових продуктів, користувачів, повідомлень тощо)

Негативні аспекти:

- підвищені вимоги до хостингової платформи, що впливає на вартість послуг хостингу;
- за вартістю розробки динамічний веб-сайт є одним з найдорожчих веб-сайтів, так само як і флеш-сайт.

Проаналізувавши переваги та недоліки динамічного веб-сайту, можна зробити висновок, що динамічний веб-сайт вимагає великих витрат на розробку, але ці витрати швидше окупаються за рахунок зменшення витрат на редагування сторінок та розвиток інформаційної системи сайту. Динамічний сайт працює довше і ефективніше, ніж статичний, і охоплює всі основні функції веб-сайту.

Динамічні сайти, веб-сторінки яких створюються або генеруються з даних, що зберігаються на сервері в базі даних.

Для створення динамічних веб-сайтів зараз використовуються різні веб-додатки. Для створення динамічних веб-сайтів використовуються різні технології, що дозволяють створювати веб-додатки.

Динамічні веб-сайти здатні реагувати на дії користувача, тобто вони можуть бути інтерактивними.

Тому динамічні веб-сайти зазвичай інтерактивні.

### **1.3. Основи використання CMS. Створення сайтів за допомогою CMS**

#### **1.3.1. Призначення систем управління**

CMS (*Content Management System*) – це комп’ютерна програма або система, що використовується для забезпечення та організації спільного процесу створення, редагування та управління контентом веб-сайту (текстовими, графічними або мультимедійними елементами) [1]. Зазвичай CMS розглядає контент як набір

неструктурованих тематичних даних, на відміну від структурованих даних веб-сайту, якими керує СУБД.

CMS має панель керування, яка є лише частиною системи, але її достатньо для керування сайтом. Існують різні системи управління контентом, створені за різними технологіями, в тому числі платні та безкоштовні.

Основні завдання CMS:

1. Збір та інтеграція в єдине ціле різних джерел знань та інформації, доступних як в організації, так і за її межами.
2. Персонал, робочі групи та проекти взаємодіють з базами знань, інформацією та даними, які вони створили, таким чином, щоб їх було легко знаходити, отримувати та повторно використовувати у звичний для користувача спосіб.

Більшість сучасних систем управління контентом реалізовані у вигляді візуального редактора (WYSIWYG) – програми, яка створює HTML-код зі спеціальних спрощених тегів, що дозволяють користувачеві легко форматувати текст.

Існує багато причин для широкого використання систем управління контентом (CMS). Найголовніша з них – зростаюча складність сучасних веб-сайтів, оскільки навіть пересічний власник сайту хоче, щоб на його сайті був блог, форум і файловий архів. А потужні компанії потребують ще більшої функціональності для своїх сайтів.

Очевидно, що писати щоразу новий програмний модуль нерозумно, тому розробники створили спеціальні бібліотеки корисних функцій, а згодом і спеціалізовані рішення, зокрема форуми, на основі яких почали з'являтися універсальні системи управління контентом.

Іншим важливим фактором стало спрощення CMS-систем. Сьогодні навіть розробник-початківець може встановити і використовувати CMS, якщо перед ним не стоять специфічні завдання. Крім того, більшість хостингових сервісів

пропонують встановлення готових до використання CMS, як у безкоштовних, так і в комерційних версіях.

Хорошим прикладом є спільнота з відкритим вихідним кодом, яка поширює безкоштовні системи управління контентом. Вони підійдуть для малого бізнесу, який не може дозволити собі дорогий комерційний продукт.

### **1.3.2. Схема роботи систем управління**

Центральним елементом будь-якої CMS є зберігання інформації. У сучасних системах управління контентом це реляційна база даних. Слово «*реляційна*» означає, що база даних складається з таблиць зі зв'язками між ними.

Якщо CMS потрібно зберігати певну інформацію, вона зберігає її в базі даних. Кожній сутності в базі даних відповідає окрема таблиця. Наприклад, таблиця, в якій зберігається вміст веб-сторінок. Окрім тексту сторінки, вона зберігає назву матеріалу, дату створення та інформацію про автора. Поле «автор» посилається на таблицю користувачів, яка містить їхні логіни, паролі та дозволи. Встановивши такі модулі, можна побудувати гнучку та надійну систему зберігання інформації.

Движок бази даних обирається залежно від платформи. Якщо використовується платформа Windows, то це MSSQL, якщо UNIX, то MYSQL.

Після того, як базу даних обрано, необхідно вирішити, як найкраще запрограмувати CMS для роботи з нею. Найкращий підхід – створити абстрактний рівень для роботи з базою даних. Це може бути реалізовано як спеціальний клас або як набір функцій. В ідеалі, основний код CMS повинен бути однаковим для кожної бази даних, а змінюватися повинен лише код, що наповнює базу даних.

Оскільки інформація для відображення (наприклад, текст статті) отримується CMS з бази даних, для відображення інформації у форматі HTML використовується механізм шаблонів.

Шаблон – це файл дизайну сторінки, створений за допомогою спеціальної мови. Як правило, це HTML-код, розмічений певним чином, який вказує, де вставити заголовок сторінки, де вставити основний текст, а де меню або інші елементи, взяті з бази даних. Найпростіший варіант – створити шаблон на PHP, але існують і більш складні рішення.

Движок шаблонів має власну досить просту мову, з якою може впоратися програміст, який не знайомий з PHP. Шаблони конвертуються в PHP-файл, а потім просто виконуються інтерпретатором PHP. Для підвищення продуктивності шаблон можна скопіювати на PHP, оскільки він буде рідко змінюватися. Таким чином, робота верстальника і програміста розділяється: один створює шаблони, а інший пише код програми.

Іншою частиною системи є система користувачів та їхніх ролей. Роль користувача – це певний набір дій, які може виконувати користувач. Ролі можна порівняти з групами користувачів у Windows. У сучасних CMS-системах ролі користувачів можна створювати і налаштовувати відповідно до задуму розробника. Зазвичай визначають кілька ролей користувачів: адміністратор, модератор, автор, користувач і гість. Кожному користувачеві може бути призначена певна роль, і призначення ролей здійснюється автоматично або вручну адміністратором. Перший варіант часто використовується на форумах, де новий статус автоматично присвоюється користувачеві при досягненні певної кількості повідомлень.

Як взаємодіють між собою компоненти CMS. Припустимо, що користувач відвідав сайт, яким керує CMS, і виконав певну дію, наприклад, поставив запитання або відправив заповнену форму.

Перш за все, CMS повинна визначити, як реагувати на таку дію. Потім модуль, відповідальний за обробку запиту користувача, збереже інформацію в базі даних (наприклад, збереже нове повідомлення на форумі), якщо запит користувача був пов'язаний з цим.

Система надсилає запит до бази даних, щоб отримати інформацію для відображення або генерує повідомлення програмно, наприклад, у відповідь на нове

повідомлення. Інформація передається модулю шаблонів, який бере поточну тему і вставляє в неї всю необхідну інформацію. Отриманий HTML-код надсилається користувачеві.

### 1.3.3 Класифікація систем управління

Для того, щоб краще зрозуміти предметну область і встановити порядок, був створений метод, який називається класифікацією. Класифікація – це поділ загальної множини на окремі відповідно до обраних критеріїв, або просто поділ на групи на основі спільних характеристик. Класифікація також необхідна у сфері систем управління, де вона повинна чітко розмежовувати системи.

На цьому етапі розглянемо два найбільш очевидні типи класифікації:

1. класифікація за «ступенем відкритості»;
2. класифікація за «розробником системи».

Що таке ступінь відкритості? Багато користувачів вже спробували такі продукти, як OpenOffice, Firefox, Linux, а також інші відомі програми та системи, які поширюються не тільки безкоштовно, але й з відкритим вихідним кодом (в рамках рухів OpenSource, GNU, FSF та інших). Останнє означає, що будь-хто може вносити зміни до вихідного коду таких програм і навіть поширювати їх у зміненому вигляді. Однак проекти з відкритим кодом – це не просто програмне забезпечення і набір вільних ліцензій, це ще й соціальне явище, засноване на принципах взаємодопомоги, а також рух, який об'єднує своїх учасників на основі ідеологічних інтересів.

Які унікальні можливості відкривають проекти з відкритим вихідним кодом для різних людей і в чому полягають переваги продуктів з відкритим вихідним кодом? Розглянемо найбільш очевидні переваги та можливості.

- Головною перевагою відкритих продуктів є їх доступність – відкритий продукт можна не тільки вільно використовувати, але й швидко оновлювати, оскільки нові версії таких продуктів виходять із



регулярністю, а виправлення помилок не передбачає очікування патчів або сервіс-паків, як це зазвичай буває у випадку з комерційними рішеннями. Продукти з відкритим вихідним кодом забезпечують свободу вибору і роботи, а також можливість міграції з одного рішення на інше з мінімальними витратами, якщо існуюче не відповідає вимогам замовника.

- Продукти з відкритим кодом, запозичуючи сторонній відкритий код і підтримуючи один одного, іноді розвиваються швидше, ніж їхні комерційні конкуренти. Завдяки свіжим ідеям та невтомній підтримці спільноти, продукти з відкритим кодом здатні не тільки конкурувати з комерційними рішеннями, але й займати домінуючі позиції на ринку. Варто зазначити, що комерційні розробники приєдналися до ініціативи і багато з них вже включили рішення з відкритим кодом у свої продукти.
- Учасником відкритого проекту може стати будь-хто, від звичайного користувача до програміста, адже великі міжнародні відкриті проекти завжди потребують локалізаторів (перекладачів), тестувальників та інших спеціалістів.
- Розробник або організація, що обирає за основу рішення з відкритим вихідним кодом, позбавляється необхідності адаптуватися до чужої корпоративної «чорної скриньки». Крім того, з відкритого продукту можна запозичувати нові та цікаві ідеї, що не тільки не забороняється, але й вітається. Проекти та продукти з відкритим кодом майже завжди відповідають стандартам, а не суперечать їм. Це значно спрощує процес розробки продукту і гарантує сумісність з іншими продуктами, які підтримують стандарти.
- Будь-який фахівець у своїй галузі, отримуючи вихідний код відкритого програмного забезпечення, має можливість проаналізувати продукт і зробити власні висновки щодо його надійності, безпеки, сумісності тощо.

- Відкритий продукт можна «зібрати» відповідно до власних вимог і таким чином максимізувати результат.

Повертаючись до класифікації за «ступенем відкритості», всі продукти можна розділити на дві групи. До першої групи належать усі відкриті продукти, а до другої – інші продукти (інші включають комерційні та вільні продукти) (рис. 1.1).

Класифікація за «розробником системи». Розробник системи, як правило, є автором і супроводжувачем системи. Розробником може бути як відкрита спільнота з її ідейними лідерами, так і ціла комерційна організація з серйозним штатом співробітників. Для зручності будемо відносити проекти спільнот до першої групи, а проекти організацій – до другої. Для того, щоб зрозуміти, чії продукти кращі, варто розглянути типові сценарії появи нових CMS (хоча це стосується не лише CMS). Почнемо з систем управління першої групи. Як виникають спільноти і як з'являються продукти спільнот?

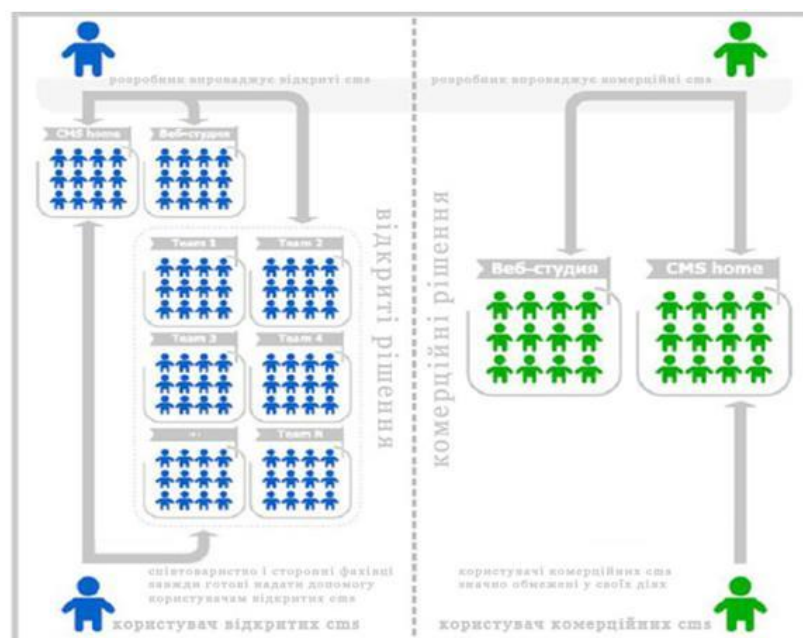


Рис. 1.1. Відмінності в роботі між відкритими та закритими системами управління контентом

У світі досить багато професіоналів-фрілансерів (висококваліфікованих програмістів, дизайнерів тощо), які цінують свою роботу і свободу. Вони ніколи не змиряються з роллю експлуатованої ланки, вважаючи за краще самостійно шукати завдання і працювати без посередників (або навіть створювати цілі команди для великих проектів), і в результаті отримувати гідну зарплату за свою роботу. Більшість з них – ідейні лідери, які відмовляються миритися зі статусом кво. Так чи інакше, їм доводиться використовувати різні «рішення» (готові ідеї та продукти) у своїй практиці. З часом вони розуміють, що існуючі рішення мають низку недоліків (обмежений функціонал, закритість тощо) або не підходять у новому технологічному середовищі, і тоді в них з'являється сміливість створити щось нове і більш доступне.

Так народжується нова ідея. Коли ідея реалізується, народжується «рішення», нехай примітивне, але вже перспективне. Потім автор вирішує зробити всі рішення публічними – так народжується новий проект. Це призводить до того, що незабаром у проекті з'являються прихильники – формується спільнота. Саме завдяки спільноті та її потужній підтримці розвиватимуться всі наступні «рішення» (при тому, що ідея залишається незмінною).

В результаті, через кілька років повністю готове і стабільне «рішення» почне витісняти комерційні продукти, а новий дизайн стане ресурсом спільноти. До другої групи належать комерційні системи. Системи цього класу в основному розробляються веб-студіями та великими інтеграторами.

Слід також зазначити, що веб-студії бувають різних типів – як команди з 2-3 осіб, так і досить серйозні організації можуть претендувати на звання веб-студії. Причому в першому випадку така веб-студія може бути найпримітивнішим «сайтобудівним» полігоном і не більше.

Сучасна CMS повинна дозволяти, не вдаючись до додаткового програмування, здійснювати редагувати вміст сторінок, включаючи додавання/видалення графіки; додавати нові сторінки; змінювати структуру сайту і різні метадані; налаштовувати реєстраційні форми; керувати опитуваннями,

голосуваннями і форумами; відображати статистику відвідувань; розподіляти права управління сайтом між користувачами [2].

CMS можна класифікувати за багатьма критеріями, ось деякі з них.

CMS-системи поділяються на групи за типом розробки. Зазвичай виділяють три класи систем: модульні, спеціалізовані та кастомізовані.

- Модульні системи управління контентом призначені для широкого спектру веб-проектів. Спочатку власники сайтів створюють їх на основі базової конфігурації «движка». Пізніше, коли потреби конкретизуються, можна придбати і підключити певний додатковий модуль, щоб розширити можливості управління контентом сайту. З часом клієнти отримують движок, адаптований під їхній проект. Це головна перевага модульних систем. Ще однією перевагою є поетапна оплата за CMS – якщо потрібна певна функція, можна просто купити необхідний модуль.
- Спеціалізовані системи управління контентом – це повноцінні CMS-системи, орієнтовані на один тип веб-проектів. Спеціалізовані системи управління контентом охоплюють всі аспекти бізнесу в одній галузі, включаючи дуже специфічні, які не реалізовані в модульних CMS. Спеціалізовані системи надають максимум можливостей для веб-розробників, залишаючись при цьому значно дешевшими за індивідуальні системи.
- Пристосовані до потреб клієнта. CMS-системи розробляються з урахуванням цілей, завдань і побажань клієнта. Це найзручніший варіант практично для будь-якого сайту. Однак створення такої CMS-системи вимагає чималих коштів у порівнянні з іншими CMS-системами і часто займає багато часу.

Усі CMS-системи також поділяються на три частини: прості системи, складні системи та системи електронного документообігу:

1. Прості CMS-системи дозволяють лише додавати, змінювати або видаляти інформацію з сайту.

2. Просунуті CMS-системи також можуть виконувати більш складні функції управління веб-проектами, наприклад, змінювати структуру сайту, змінювати права доступу і бути інструментом для створення нових веб-ресурсів.
3. CMS для електронного документообігу. Їхнє головне призначення – оптимізувати бізнес-процеси всередині компанії, а публікація контенту на веб-сайті – це одна з можливостей, часто не найважливіша. Дуже часто системи електронного документообігу інтегровані в загальну ІТ-систему компанії.

В основі класифікації CMS лежить їхня основна функція – тип контенту. Найпоширеніші типи – портали, блоги, інтернет-магазини, каталоги тощо. Існують також універсальні системи, які завдяки своїй модульній структурі можуть бути адаптовані до будь-якого типу контенту.

#### **1.3.4. Переваги CMS**

Щоб оцінити переваги систем управління контентом, необхідно повернутися в минуле і подивитися, як створювалися веб-ресурси до появи таких систем і як вони створюються сьогодні.

Початок розвитку веб-середовища можна віднести до середини 1990-х років. Цей період характеризувався відносно низьким рівнем розвитку веб-технологій, і багато динамічних веб-проектів (тобто здатних до інтерактивної взаємодії) могли розробляти лише програмісти.

Вся динаміка реалізовувалася за допомогою CGI та інших складних технологій, тому звичайний користувач, який створював власний проект, міг розраховувати лише на статичний дизайн в масштабах базового сайту і не більше. Такий сайт зазвичай являв собою набір статичних html-сторінок, підготовлених у візуальних html-редакторах, які почали з'являтися приблизно з середини 1995 року. Після того, як сторінки були набрані, їх пов'язували між собою (для забезпечення

навігації між сторінками) і розміщували на сервері. Вся робота з оновлення інформації та перевірки працездатності проекту належала автору. Так, наприклад, якщо на сторінці потрібно було змінити посилання, автор повинен був знайти цю сторінку серед інших, внести в неї зміни і перезавантажити сторінку на сервер. Якщо ж потрібно було не тільки змінити існуючу сторінку, але й додати нову, то автору також доводилося вирішувати питання, пов'язані з логічною та фізичною реалізацією останньої в рамках всього проекту. А якщо кількість сторінок зростала, то «керувати» ними ставало ще складніше. А змінювати і розширювати динамічні проекти було ще проблематичніше.

На той час систем для автоматизації всіх цих процесів практично не існувало, точніше, не було доступних і легких систем, не було вибору між відкритими і комерційними системами, а популярність нечисленних комерційних систем була такою, що майже ніхто не знав про їхнє існування. Однак така ситуація тривала недовго, і на зміну статичним html-сторінкам, CGI-програмуванню прийшли більш «дружні» технології – Asp (кінець 1996 року), ColdFusion (червень 1995 року), а згодом PHP (версія 2, випущена в 1997 році). Нові технології дозволили поєднати розмітку html-сторінок і простий програмний код, зробивши пасивні html-сторінки активними. Активність останніх спростила організацію інтерактивної взаємодії з користувачами, оскільки кожен раз, коли користувач звертався до однієї і тієї ж активної сторінки, він міг отримувати нові дані. Водночас активні сторінки створювалися для автоматизації певних дій, таких як завантаження файлів на сервер або створення нових сторінок. Такі корисні сторінки поступово об'єднувалися в пакети підтримки. Такі пакети використовувалися для вирішення типових завдань, але їхніх можливостей не завжди було достатньо, і багато клієнтів вважали за краще замовляти індивідуальні системи управління для своїх проектів. Спочатку ці кастомні системи не були універсальними – коли їх функціонал потрібно було змінити або розширити, замовнику доводилося знову звертатися до розробників. Незабаром розробники самі вирішили створювати універсальні системи. Так з'явилися перші універсальні комерційні системи управління.

Абревіатура CMS також була прийнята україномовними користувачами. Пізніше з'явилися і відкриті системи управління. Подальша еволюція цих систем вивела всю сферу на такий рівень розвитку, що нинішній вираз «система управління контентом» вже не зовсім точний, оскільки функціонал сучасних систем значно розширився і вони тепер можуть працювати з темами, модулями та керувати іншими елементами. Сучасні системи можна з упевненістю назвати «системами управління веб-проектами».

Сьогодні системи управління контентом – це не тільки зручний менеджер-оболонка для користувача, але й потужний інструмент для веб-розробників (останнє стосується не всіх систем). Завдяки таким системам все менше і менше потрібно створювати веб-проекти з нуля – підготовлений користувач або інтегратор може просто вибрати, встановити і налаштувати існуючу систему, щоб отримати прийнятний результат професійного рівня швидко і без зайвих фінансових витрат. Сьогодні більшість організацій мають власний веб-сайт у тій чи іншій формі. Набагато менше компаній мають внутрішню інтранет-систему. Часто виникає питання: чому б не створити ще один сайт для внутрішнього користування і зберігати там необхідні документи? Які переваги використання CMS? Такі питання часто виникають тому, що складність адміністрування та управління ІТ-інфраструктурою в цілому та управління веб-сайтом компанії зазвичай прихована від користувачів. Більшість сучасних сайтів компаній базуються на статичних сторінках. Як наслідок, інформація на таких сторінках прихована від звичайних користувачів. Водночас ця операція вимагає знань основ програмування та HTML. Імовірність помилок, особливо для звичайних користувачів, дуже висока.

Таким чином, використання CMS надає наступні переваги:

- Швидке оновлення інформації – інформація публікується співробітником, який нею володіє, без додаткових посередників у вигляді технічних спеціалістів.

- Зменшення витрат на підтримку – інформація оновлюється самостійно, немає необхідності оплачувати роботу штатних або зовнішніх веб-майстрів.
- Надання додаткових сервісів користувачеві – деякі сервіси, такі як пошук, форуми, голосування тощо, потребують інтерактивної взаємодії з користувачем. Вони вже реалізовані в CMS.
- Скорочення часу та витрат на розробку – найпопулярніші функції вже реалізовані в CMS і можуть бути використані одразу.
- Покращена якість розробки – під час розробки використовуються готові модулі, які вже були протестовані повністю або частково кілька разів.
- Зменшити вартість подальших модифікацій – CMS дозволяє відокремити дані та їхнє представлення. Це значно полегшує зміну зовнішнього вигляду сторінки, ніж у випадку зі статичною сторінкою.

Щоб забезпечити ці переваги, КМВ має вирішити наступні ключові проблеми:

- Публікація інформації нетехнічним фахівцем.
- Відокремлення даних від презентації.
- Організація спільної роботи при публікації інформації.
- Пошукові можливості.
- Інші послуги включають форуми, голосування, анкети тощо.

#### **1.4. Огляд існуючих систем управління контентом**

Системи управління контентом (CMS) стали дуже популярними в останні роки і активно використовуються як початківцями, так і професіоналами веб-розробки.

Сучасні CMS-системи використовуються для створення веб-проектів практично будь-якої складності (рис. 1.2). Професійні студії дизайну пропонують послуги зі створення корпоративних сайтів, інтернет-магазинів, новинних та



інформаційних порталів і навіть соціальних мереж на базі існуючих CMS-систем, що добре зарекомендували себе на ринку.

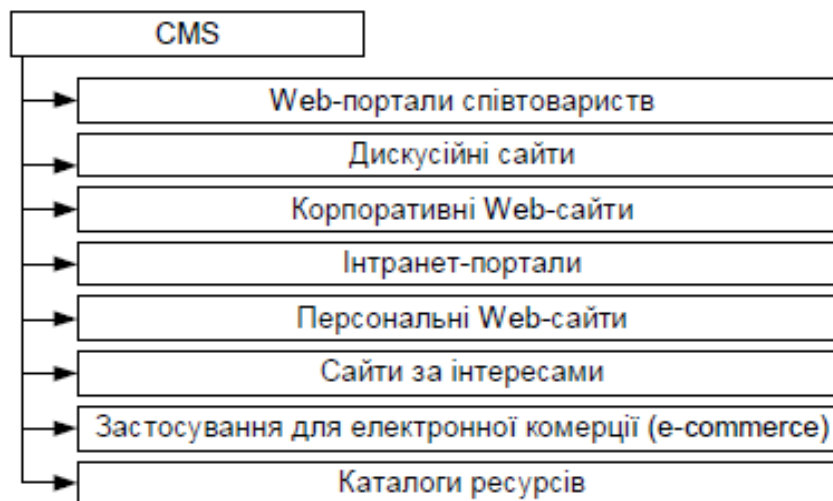


Рис 1.2. Типи веб-проектів

Наразі існують десятки CMS-систем різної складності, головною перевагою яких є зручність редагування та додавання інформації. Можливості таких систем, як правило, постійно вдосконалюються. Більшість CMS, в тому числі комерційних, використовують модульну архітектуру, що дозволяє легко адаптувати функціонал системи під конкретні завдання.

Сьогодні в Україні широко використовуються понад 50 безкоштовних і 100 комерційних систем управління контентом. Звичайно, серед них є системи управління, спеціально розроблені для сайтів певних розмірів і типів, але всі ці CMS легко дозволяють редагувати всю інформацію, розміщену на сайті. Основними недоліками безкоштовної системи управління контентом є повна відсутність підтримки і велика затримка з випуском оновлень, що робить таку систему управління контентом небезпечною. З цієї причини використання безкоштовної системи управління контентом вкрай небажано для комерційного сайту.

Використання комерційної системи управління контентом може підвищити безпеку веб-сервера компанії, але не всі постачальники CMS дбають про безпеку

належним чином. Це пов'язано з тим, що основні зусилля компаній зосереджені на розробці користувацького інтерфейсу. Фактично, майже всі компанії, які купують CMS, оцінюють її лише з точки зору простоти використання, забуваючи про безпеку.

Сучасний підхід до веб-розробки полягає в автоматичній генерації сторінок на основі певних вихідних даних (контенту), що зберігаються в структурованому вигляді. Зазвичай контент зберігається в реляційній базі даних, а сторінки генеруються спеціальними скриптами, написаними однією з сучасних мов програмування.

Система управління контентом забезпечує зручне управління даними, що зберігаються в базі даних, і забезпечує відображення веб-сторінок. Такі системи часто називають «динамічними», оскільки сторінки можуть створюватися «на льоту» за запитом користувача, але точніше було б назвати їх системами, заснованими на реляційній базі даних і серверних сценаріях. Сучасний підхід до створення веб-сторінок полягає в автоматичній генерації сторінок на основі початкових даних (контенту), що зберігаються в структурованому вигляді.

Зазвичай контент зберігається в реляційній базі даних, а сторінки генеруються спеціальними скриптами, написаними однією з сучасних мов програмування.

Існує ряд об'єктивних критеріїв, які слід враховувати при виборі системи управління контентом:

- вартість;
- юзабіліті;
- функціональність системи;
- наявність технічної підтримки.

Крім того, кожен серйозний продукт має демо-версію на своєму веб-сайті або пропонує користувачам завантажити та встановити локальну версію продукту на свій комп'ютер для тестування.

Ринок CMS досить чітко розділений на два основних типи представлених на ньому продуктів – коробкові системи та послуги з встановлення та впровадження CMS власними силами, які надає компанія-розробник сайту. Часто фахівці не відносять другий тип рішень до CMS. Але така думка не зовсім правильна, оскільки такі системи виконують всі стандартні функції CMS, хоча і не є самостійними продуктами. Більше того, більшість популярних систем потрапляють саме в цю проміжну категорію.

Обираючи між коробковою та вбудованою системою, важливо враховувати переваги та недоліки обох варіантів. Спочатку розглянемо коробковий продукт. Головною особливістю коробкового продукту є те, що така система може бути самостійно встановлена і запущена без залучення програміста. Для цього програмне забезпечення повинно мати такі характеристики: портативність самої системи, наявність документації для користувачів і розробників, можливість роботи на стандартному програмному забезпеченні без додаткового встановлення спеціалізованого програмного забезпечення на сервері тощо.

Основний показник якості коробкового продукту – переважання зовнішніх налаштувань системи над внутрішніми. В ідеалі розробник взагалі не повинен займатися інтеграцією, а лише надавати технічну підтримку та консультації.

Основними перевагами «коробки» є можливість самостійного встановлення системи, розвинений сервіс оновлення та підтримки, а також можливість самостійного додавання програмних модулів (за наявності API) до системи.

Основні недоліки: відносно високий рівень знань, необхідний для встановлення системи, часті конфлікти з іншими програмними продуктами (наприклад, при інтеграції в загальну IT-структуру компанії).

Провідні гравці на ринку коробочної продукції:

– NetCat – це система, що підходить для проектів будь-якої складності. Стандартні функції можуть бути легко доповнені кастомізованими функціями. Вона має окремий інтерфейс для користувача та розробника (рис. 1.3).

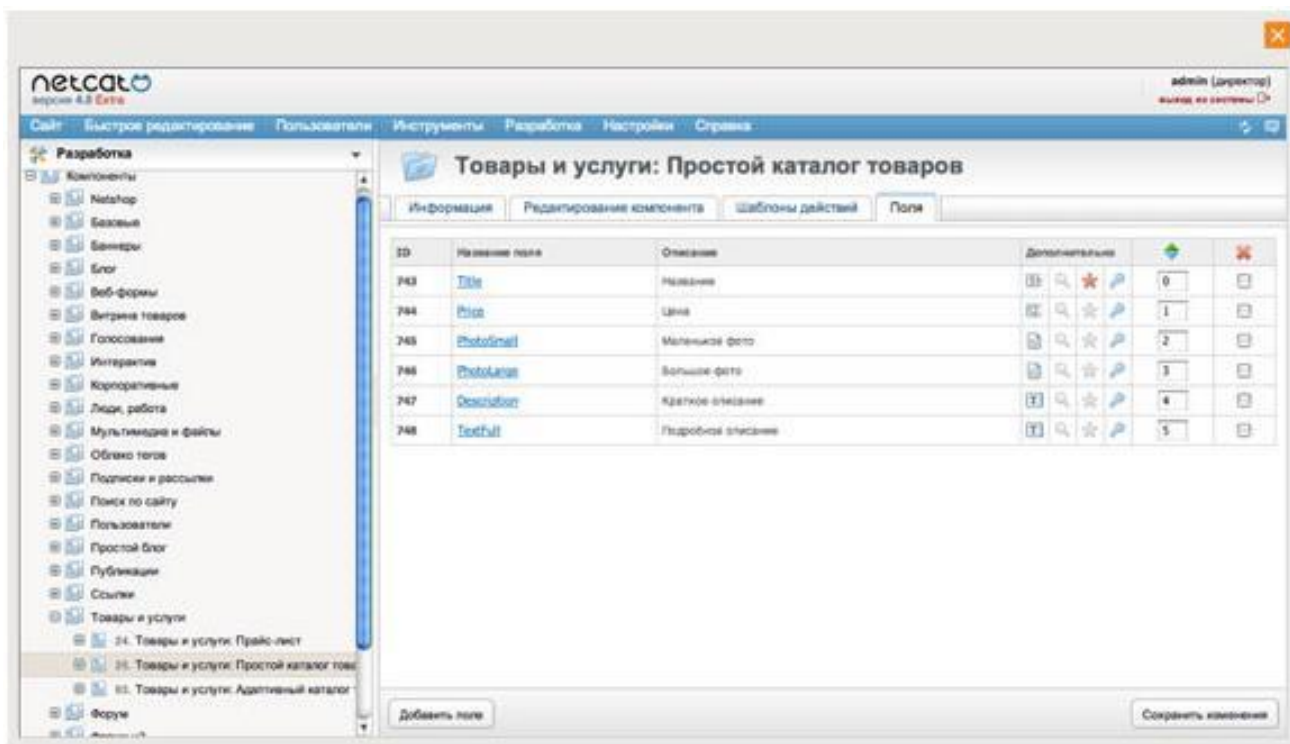


Рис. 1.3. NetCat – система управління контентом

– 1С-Бітрікс – це, мабуть, найпотужніша і найпопулярніша модульна система управління контентом. Вона складається з 23 модулів для будь-якого проекту, включаючи бізнес-проекти [3]. Вона підтримує кілька сайтів, безпеку, стабільність, роботу з дизайнами та шаблонами, оновлення, чудову документацію з відео та багато-багато іншого (рис. 1.4).

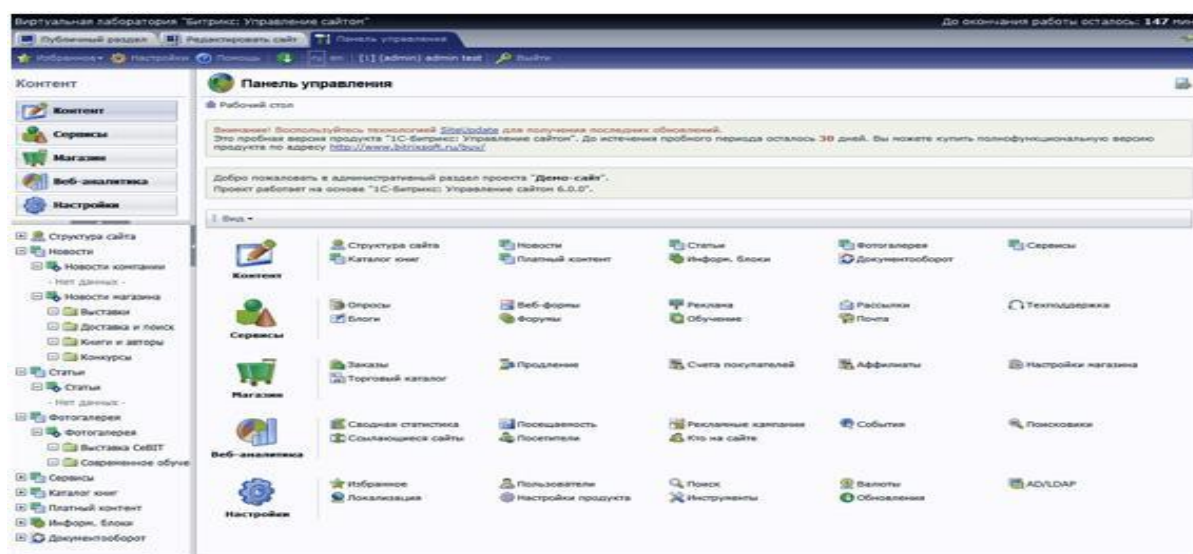


Рис. 1.4. Система управління контентом 1С-Бітрікс

– Saitistika – це зручна корпоративна система управління даними, що підходить для різних напрямків проєктів. Доступні модулі дозволяють ефективно працювати з контентом сайту (рис. 1.5).

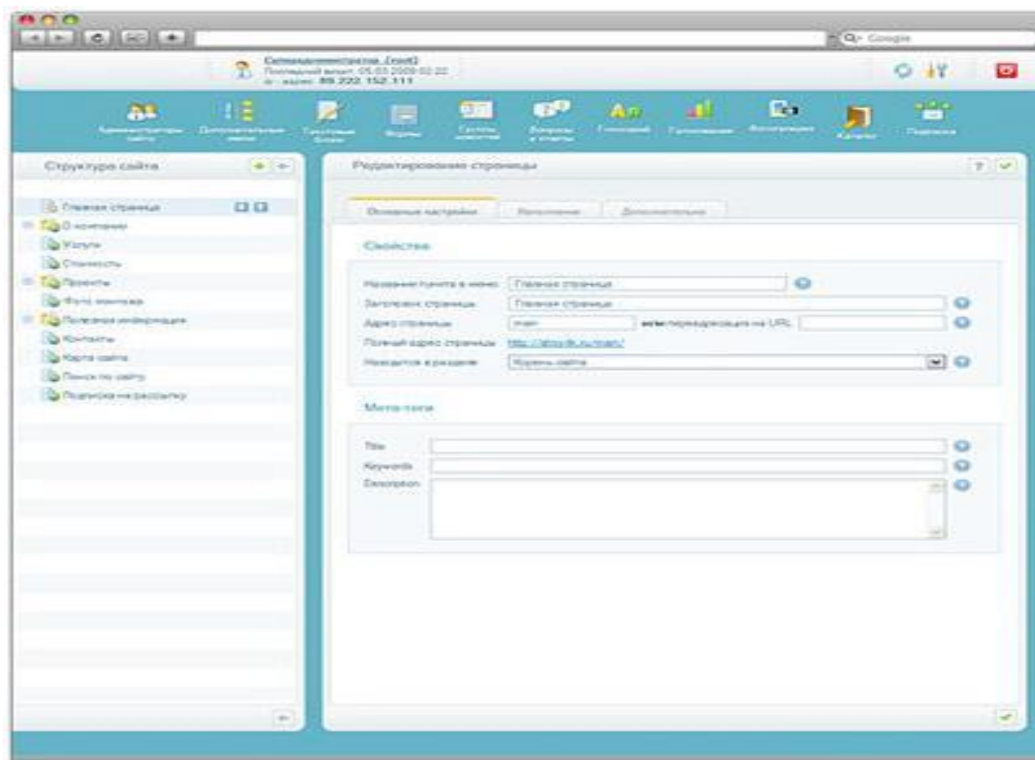


Рис. 1.5. Система управління контентом Saitistika

– Joomla – одна з найпотужніших і найпоширеніших систем управління контентом. Своєю популярністю вона завдячує простоті використання, надійності та легкості встановлення у поєднанні з унікальною функціональністю управління (рис. 1.6).



Рис. 1.6. Система управління контентом Joomla

– WordPress – це система для ведення блогів. Вона вирізняється простотою встановлення, налаштування, додатковими модулями та великим вибором тем (рис. 1.7).

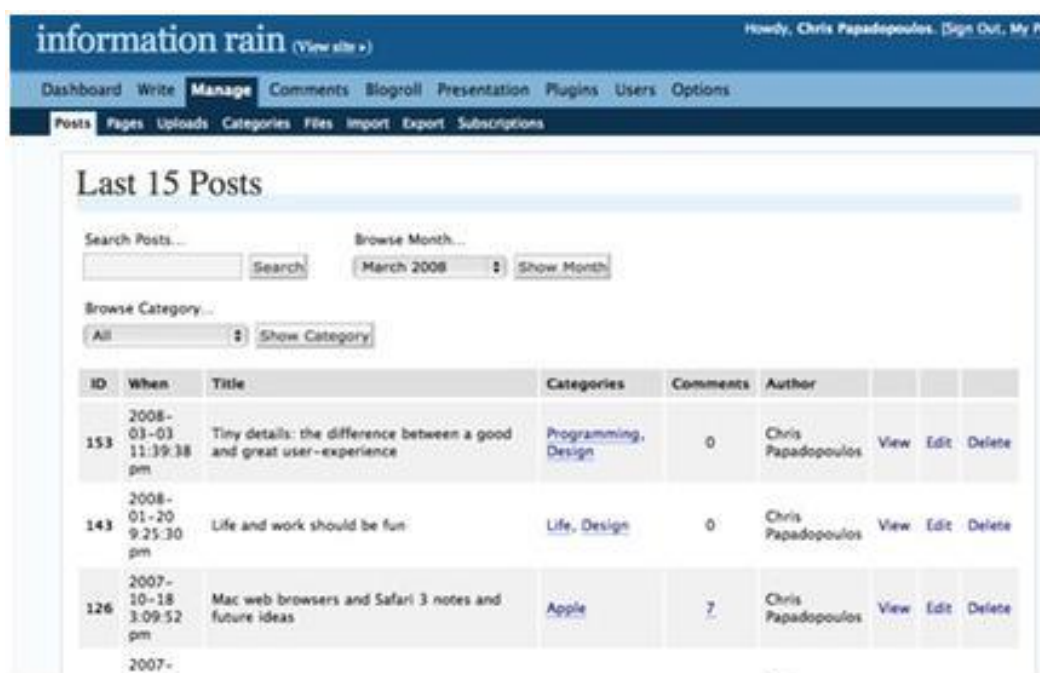


Рис. 1.7. Система управління контентом WordPress

–Drupal є одним з лідерів на ринку CMS, але пересічному користувачеві буде складно його освоїти, хоча він має великий набір модулів у стандартній комплектації. Крім того, систему можна розширювати за допомогою сторонніх

модулів, які розроблялися роками. На жаль, звичного механізму українізації системи не передбачено, і все доведеться робити самостійно і своїми руками (рис. 1.8).

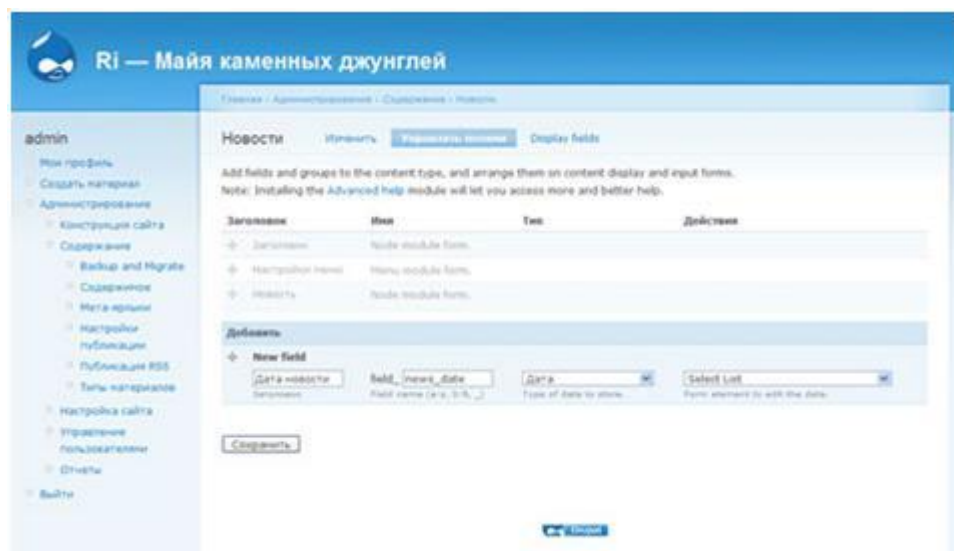


Рис. 1.8. Система управління контентом Drupal

Коли мова йде про внутрішні системи, ці типи систем, як правило, не є окремим продуктом і зазвичай поставляються в стандартній комплектації з розробленим веб-сайтом. Вони часто не мають стандартизованого обліку версій – невеликі домашні системи можуть навіть мати різні модифікації для кожного окремого проекту. Промислові рішення (такі як RBC Contents) постачаються як окремі продукти, що мають набір версій, сумісних між собою.

Недоліками таких систем є необхідність встановлення на сервері спеціального програмного забезпечення, обов'язкова участь розробника у створенні сайту та відсутність детальної документації. Такі рішення мають і очевидні переваги: можливість зміни системи під потреби кожного конкретного проекту, відсутність проблем з впровадженням і підтримкою (оскільки за цим стежить розробник) і можливість інтеграції з іншими інтегрованими системами, що надаються розробником.

Провідні гравці у сфері внутрішньокорпоративних систем:

– RBC Contents – це ефективна система управління контентом з хорошим співвідношенням ціни та якості. Вона підтримує різні мови, багатосайтовість, оновлення та хороший розділ адміністрування (рис. 1.9).

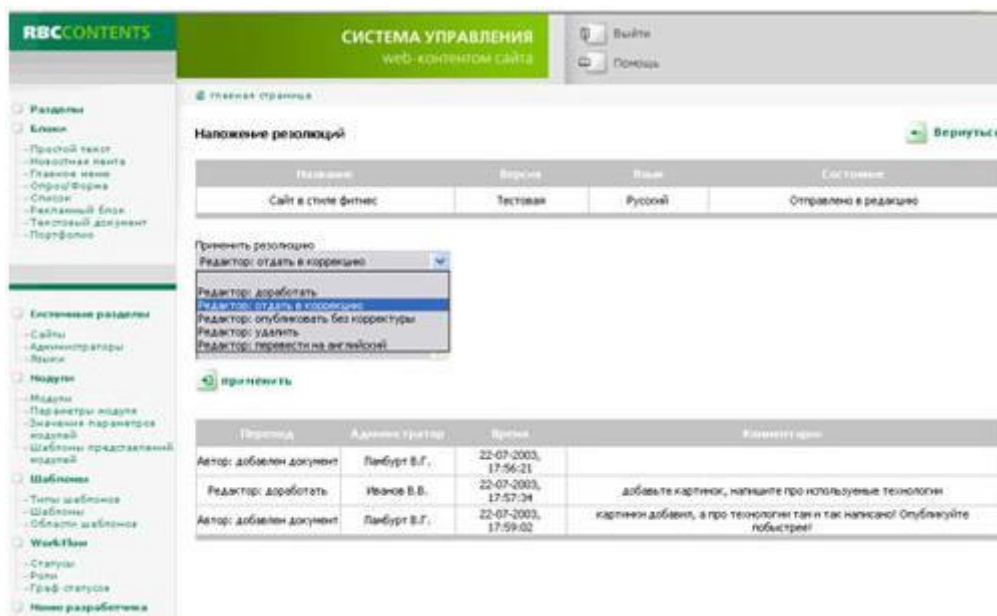


Рис. 1.9. Система управління контентом RBC Contents

– Optimizer.NET – поєднує в собі простоту використання та безпеку. З ним легко створити сайт будь-якого напрямку, а потім так само легко керувати ним (рис. 1.10).



Рис. 1.10. Система управління контентом Optimizer



– QP7.Framework – це система з широкими можливостями як для розробників, так і для користувачів. Вона підходить для запуску всіх видів проектів з відстеженням статистики, управлінням документами, рекламою та зручною системою публікацій (рис. 1.11).

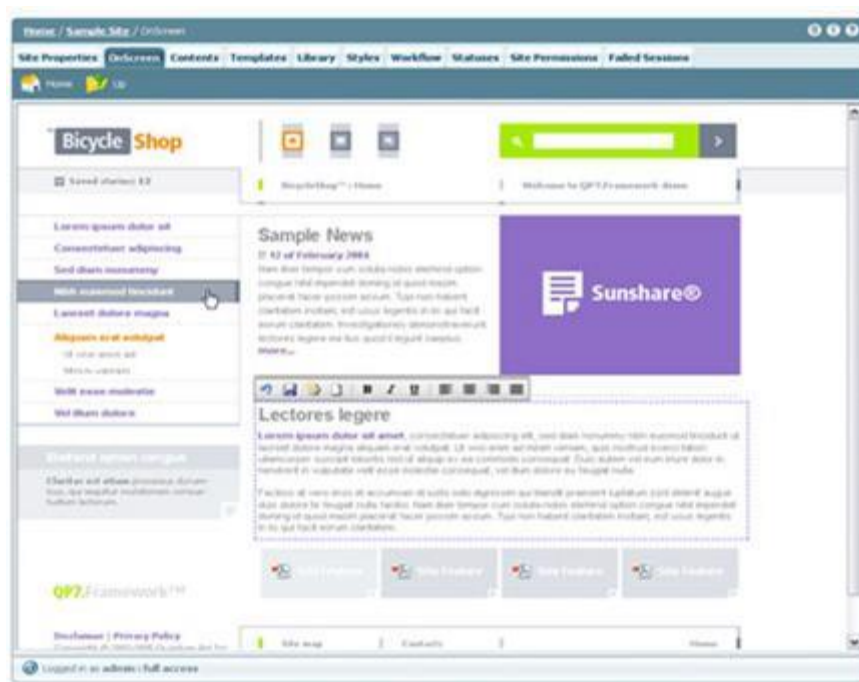


Рис. 1.11. Система управління контентом QP7.Framework

Наразі, за оцінками різних експертів, на ринку представлено понад 100 систем, анонсованих розробниками. Однак лише близько 30 з них можна вважати професійними. Критеріїв професійності системи не так багато: постійна команда розробників, комерційний успіх, велика кількість інсталяцій та висока функціональність.

Сьогодні CMS встановлені на більш ніж 70% комерційних сайтів. Немає сенсу говорити про середню вартість рішень, оскільки ціновий діапазон визначається діапазоном від 150 до 1500 доларів США для систем легкого та середнього рівня і від 2000 доларів США і вище для систем високого рівня.

Ринок CMS стрімко розвивається, і його лідери можуть змінитися всього за кілька місяців. Постійно з'являються нові системи та випускаються нові версії існуючих CMS. Існує низка якісних продуктів, які через низьку популярність і

функціональність або низькі технічні показники не займають лідируючі позиції в галузі, але є дуже перспективними.

## Висновок до розділу 1

Вибір системи управління контентом – це складний процес, який вимагає врахування багатьох факторів, перерахованих вище. Перед створенням і вибором важливо врахувати, що потрібно в розробці і для чого буде використовуватися сайт. Це зумовлює очікувану модульність системи, а також її вартість. У розробленій системі управління контентом візуалізація не замінює прості методи створення об'єктів: існують способи введення даних вручну. Шаблонна система побудови сайту дозволить налаштувати практично будь-який дизайн для ІС. Для будь-якої сторінки можна встановити власні стилі оформлення. Система має модульну конструкцію.

Можна підключити або відключити будь-який модуль. Код написаний з чітким розділенням HTML і PHP. Це підвищує читабельність коду. Весь HTML розміщується в шаблонах. Великою перевагою системи є те, що абсолютно весь контент зберігається в базі даних і редагується безпосередньо з адміністративної панелі, що дозволяє легко оновлювати інформацію на сайті без редагування будь-яких файлів. Ця особливість дозволяє легко переносити систему з одного хостингу на інший (наприклад, локальний) – достатньо скористатися пунктом меню «Зберегти базу даних» на одному комп'ютері і виконати «Відновити базу даних» на іншому. Система дозволяє створювати деревоподібну структуру сайту з будь-яким контентом, необмежену з точки зору вкладено

## РОЗДІЛ 2

### ПРОЕКТУВАННЯ МОДЕЛІ СИСТЕМИ НА ОСНОВІ ШАБЛОНІЗАТОРІВ. ОЦІНКА ЯКОСТІ СУЧАСНИХ СИСТЕМ УПРАВЛІННЯ WEB-САЙТОМ

#### 2.1 Основні вимоги до систем

Основним критерієм вибору має бути сумісність інформаційної моделі системи з вимогами до веб-сайту.

Інформаційна модель – це структура того, з чого складається веб-сайт з точки зору системи. Вона визначає, чи буде сайт складатися з «цегли», «колод» або «збірних панелей». Щоб пояснити це, наведемо приклади.

Приклад 1:

Інформаційна модель веб-сайту громади. Сайт складається з новин. Новини можуть розміщувати учасники спільноти. Меню сайту містить посилання на найважливіші новини. Інші новини поділені на розділи. Учасники спільноти можуть додавати коментарі до новин, які також є новинами.

Приклад 2:

Інформаційна модель промо-сайту. Веб-сайт складається з однієї головної сторінки, на якій розміщено флеш-ролик, що демонструє переваги продукту. У нижній частині сторінки міститься контактна інформація та посилання на основний сайт виробника.

Приклад 3:

Інформаційна модель веб-сайту команди розробників (вікі). Веб-сайт складається зі сторінок, які можуть посилатися на інші сторінки.

Сторінки можуть редагувати всі розробники. Меню містить посилання на найважливіші сторінки, і кожен розробник створює найзручніше для себе меню.

Кафедра КІТ (47)				НАУ 23.26.51 000 ПЗ			
Виконав	Янголь В.Є.			ПРОЕКТУВАННЯ МОДЕЛІ СИСТЕМИ НА ОСНОВІ ШАБЛОНІЗАТОРІВ. ОЦІНКА ЯКОСТІ СУЧАСНИХ СИСТЕМ УПРАВЛІННЯ WEB- САЙТОМ	Літера	Аркуш	Аркушіє
Керівник	Моденов Ю.Б.					44	13
Консульт.					УС-211М 122		
Н-кترول.	Райчев І.Е						

#### Приклад 4:

Інформаційна модель корпоративного сайту. Сайт складається зі сторінок, корпоративних новин та статей. Статті поділяються на теми. Меню сайту може містити пункти до третього рівня вкладеності та посилання на сторінки сайту. Деякі сторінки можуть містити посилання на статті з певної тематики. За зміст сторінок, статей і новин відповідає редактор сайту.

Ці відмінності в інформаційних моделях не виключають непрофільного використання систем управління контентом. Візьмемо, наприклад, систему управління, яка включає інформаційну модель сайту спільноти з прикладу 1. Перейменуємо новини на сторінки та статті, призначимо одного члена спільноти редактором і заборонимо іншим публікувати новини (сторінки). Приховаємо всі елементи навігації та управління, характерні для спільноти. Результат буде виглядати точно так само, як корпоративний сайт з прикладу 4. Однак внутрішня структура сайту буде іншою, що ускладнить додавання інших елементів, необхідних для корпоративного сайту, і в кінцевому підсумку збільшить вартість підтримки. Так само літак можна перетворити на автомобіль, якщо залишити шасі, кабінку і двигун, а решту сховати.

Помилка у виборі інформаційної моделі може призвести до багаторазового збільшення складності внесення змін на сайт і практичної неможливості подальшого розвитку. До вибору інформаційної моделі слід підходити з обережністю.

Інформаційна модель значною мірою визначає структуру сайту. Навігація по сайту повинна максимально відповідати структурі. Користувач повинен мати можливість легко знайти потрібну йому сторінку, причому не одним способом, а кількома. Основним елементом навігації зазвичай є меню. На корпоративних сайтах воно має кілька рівнів вкладеності. Як додаткові інструменти навігації зазвичай використовуються карта сайту та інструменти пошуку по сайту. Крім

того, щоб користувач міг зрозуміти, де він знаходиться, часто використовуються так звані «хлібні крихти», які вказують на положення сторінки в ієрархії сайту (наприклад: Головна > Продукти > Програмне забезпечення > Sun Java System).

При виборі системи управління контентом слід звернути увагу на наявність додаткових інструментів навігації.

Сучасні пошукові системи висувають дедалі вищі вимоги до формату веб-сторінок. Пошукові системи набагато чутливіші до відхилень від стандарту HTML, ніж сучасні браузері. Якщо сторінка містить незрозумілі елементи, браузер все одно спробує її відобразити, тоді як пошукова система може просто проігнорувати її.

Для того, щоб побачити, які сторінки створює та чи інша система управління контентом, необхідно подивитися на готові сайти, створені на основі цієї системи (наприклад, на власний сайт системи). Відповідність сайту стандарту HTML можна перевірити за допомогою HTML-валідатора. Слід віддавати перевагу системі, яка створює найбільш відповідний код і найменшу кількість коду. Адреси головних сторінок повинні бути максимально простими і не містити знаків питання з числовими атрибутами. Крім того, система повинна надавати можливість редагування заголовка кожної HTML-сторінки та META-тегів, а також встановлювати альтернативний текст для розміщених зображень. Розширюваність і масштабованість Якщо ви маєте намір використовувати обрану вами систему більше одного дня, ви повинні знати, що вимоги до неї будуть змінюватися (іноді навіть навпаки). Тому система повинна бути достатньо гнучкою, щоб витримувати необхідні зміни.

Розширюваність систем управління контентом зазвичай забезпечується модульною структурою, тобто можливістю підключення до системи модулів, які змінюють поведінку системи та розширюють її функціональність. Необхідно

оцінити два фактори: наскільки функціональність може бути розширена або змінена за допомогою модулів і наскільки легко писати модулі.

На другий фактор суттєво впливає наявність детальної документації з розробки модулів.

Слід розуміти, що модулі розширюють функціональність системи, але рідко можуть суттєво змінити інформаційну модель. Можна перетворити стайню на готель, додавши модулі, але не можна перетворити її на хмарочос, бо доведеться змінювати фундамент. Крім того, важливо враховувати максимальну відвідуваність і кількість документів, на яку розрахована система. Для цього корисно подивитися на роботу готових сайтів, побудованих на цій системі.

## **2.2 Критерії вибору систем управління**

CMS має бути обрана на основі наступних критеріїв:

### **1. Функціональність.**

Сучасна CMS повинна дозволяти робити це, не вдаючись до програмування:

- редагувати вміст сторінки, включаючи додавання/видалення графіки;
- додавати нові сторінки;
- змінювати структуру сторінки та різні метадані;
- створювати реєстраційні форми;
- керувати опитуваннями, голосуваннями та форумами;
- вести статистику відвідувань;
- розподіляти права на управління сайтом між користувачами.

Не так багато CMS-систем реалізують такий функціонал:

- система затвердження матеріалів до публікації (документообіг);

- розсилати листи;
- можливість встановити URL-адресу веб-сайту, яка читається пошуковими системами та є зрозумілою для відвідувачів;
- інтеграція в CRM-системи;
- управління проектами;
- персоналізація.

Якщо потрібен функціонал, якого немає в наявності, бажано, щоб система дозволяла розширювати можливості шляхом додавання власних модулів. Деякі системи мають для цього API, деякі поширюються у вихідному коді, але є й такі, що не мають нічого з перерахованого.

## 2 Безпека.

Після функціональності це, мабуть, другий за важливістю критерій. Необхідно враховувати надійність системи як від зовнішніх атак, так і від необережних дій користувачів системи.

## 3. Оновлення.

Не менш важливою перевагою готових CMS-систем над кастомними є те, що вони створюються розробником без участі клієнта і можуть оновлюватися. Тут можна не тільки отримати функціонал без необхідності розробляти його самостійно, але й знайти шляхи для вирішення різних нових завдань.

## 4. Гнучкість.

Гнучкість системи залежить від інтенсивності роботи та необхідних навичок персоналу для адаптації системи до своїх вимог – дизайну, структури, правової політики.

## 5. Вартість.



При оцінці витрат також важливо враховувати кількість сайтів, які можна створити на одній копії системи, вартість технічної підтримки та подальших оновлень.

#### 6. Технічні вимоги.

Система не повинна вимагати специфічних налаштувань хостингу, які можуть бути надані дуже обмеженою кількістю хостинг-провайдерів. Дуже важливим критерієм є вимогливість системи до ресурсів, її «жадібність» до дискового простору, частоти процесора та оперативної пам'яті. Якщо перший з цих параметрів не є особливо важливим і дізнатися його не складно, то наступні два, навпаки, є надзвичайно важливими і знайти достовірну інформацію про них вкрай складно.

#### 7. Документація.

Успіх використання тієї чи іншої системи значною мірою залежить від того, наскільки вона проста для розуміння користувачами, що залежить від наявності або відсутності документації, довідкової та навчальної літератури і, звичайно, від необхідного рівня технічної підготовки.

#### 8. Простота у використанні.

Деякі CMS-системи призначені більше для спеціалістів, ніж для звичайних користувачів. Крім того, кожна CMS вимагає часу на вивчення, іноді досить тривалого. Деякі CMS-системи важко освоїти, тому що вони погано розроблені. Тому цей критерій є одним з найважливіших, оскільки оновленням сайтів зазвичай займаються люди без досвіду і ті, хто веде власний бізнес і рекламу в інтернеті самостійно.

### 2.3. Порівняння CMS на основі вимог та критеріїв.

У 2003 році перша версія WordPress одразу ж стала надзвичайно популярною, ідеально вписавшись у хвилю блогерського руху – і, зрештою, вона багато в чому визначила сучасну блогосферу.

Надзвичайно проста установка, зручне адміністрування та оптимізація під пошукові системи – все це фактори, які зробили WordPress справді культовим движком [4]. Хоча WordPress спочатку позиціонувався і досі позиціонується як движок для блогів, умільці розробили спеціальні шаблони та модулі, які дозволяють використовувати його для сайтів-візиток, каталогів і навіть інтернет-магазинів.

На основі WordPress було створено багато компіляцій та спеціалізованих модифікацій, деякі з яких отримали власні назви. Сьогодні WordPress є дуже широко використовуваним і популярним движком у всьому світі. WordPress є ідеальною платформою для публікацій з акцентом на сучасний вигляд, стандарти та зручність використання (таблиця 2.1). WordPress є безкоштовним і вільно поширюваним.

Таблиця 2.1

Особливості системи Wordpress

<b>Характеристики системи управління контентом Wordpress</b>	<b>Значення</b>
Тип CMS	Блог
Вартість	Безкоштовно
Навантаження на сервер	Від середнього до високого
Складність доопрацювання	Середня
Складність адміністрування	Низька

Joomla – найпопулярніша портальна CMS у світі. Ця портальна система була розроблена в 2005 році на базі також досить популярної CMS Mambo. Популярність Joomla в першу чергу базується на величезній кількості шаблонів і модулів, які дозволяють реалізувати на цьому движку практично все, що завгодно – причому без залучення PHP-програмістів. Слід зазначити, що сам движок (його ядро) розвивається не дуже швидко, але завдяки багатьом стороннім розробкам це майже не впливає на його актуальність[5].

Систему Joomla можна легко встановити на будь-який хостинг, що підтримує технології PHP, MySQL, Apache.

Joomla підходить як для невеликих проєктів (персональні сайти, інформаційні сайти), так і для інтернет-магазинів, корпоративних і великих інформаційних порталів. Цю систему завжди можна розширити відповідно до потреб за допомогою гнучкої системи компонентів і модулів.

Основний недолік CMS Joomla впливає з її власних переваг – початкова орієнтація на сторонні модулі зробила систему надзвичайно важкою для написання коду і відверто неоптимізованою (табл. 2.2). Це особливо помітно в різних версіях Joomla, які вже містять певні набори модулів, що часто дублюють один одного за функціональністю [6]. Система управління контентом Joomla є безкоштовною.

Таблиця 2.2

#### Можливості Joomla

<b>Характеристики CMS Joomla</b>	<b>Значення</b>
Тип CMS	Портальна
Вартість	Безкоштовно
Навантаження на сервер	Дуже високий
Складність доопрацювання	Висока

Складність адміністрування	Низька
----------------------------	--------

NetCat – популярна комерційна CMS порталного типу з дещо більш дружнім інтерфейсом і більш доступною ціною в порівнянні з Бітрікс. Основний функціонал майже ідентичний.

NetCat дозволяє створювати корпоративні та інформаційні портали, онлайн-бібліотеки та інші інтерактивні сайти з елементами Web 2.0. Слід розуміти, що, як і Бітрікс, NetCat не призначений для створення сайтів-візиток – хоча іноді його використовують і для цих цілей, якщо замовник сайту в першу чергу керується репутацією бренду.

Головною особливістю NetCat є наявність двох підрозділів панелі адміністрування: для розробників та менеджерів. Професійно реалізований інтерфейс є зручним для користувача, але об’єктивна складність системи все ж не дозволяє повноцінно використовувати її без попереднього навчання (таблиця 2.3).

Таблиця 2.3

#### Функції NetCat

<b>Характеристики CMS NetCat</b>	<b>Значення</b>
Тип CMS	Портальна
Вартість	Від 1200 грн до 7000 грн
Навантаження на сервер	Дуже високе
Складність доопрацювання	Висока
Складність адміністрування	Середня

Drupal – безкоштовна портална CMS з модульною системою, друга за популярністю в інтернеті після Joomla [7].

Проект Drupal був запущений в 2000 році і спочатку орієнтувався на публічну ліцензію GNU, що дозволяє комерційне використання движка. У 2003 році почалося глобальне проникнення Drupal на територію пострадянських країн, був запущений офіційний сайт Drupal в цьому сегменті і місцеві розробники почали все більше залучатися до розвитку проекту.

Перевагою цієї системи є її модульність, що дозволяє створювати широкий спектр сайтів, від блогів до каталогів та інтернет-магазинів, на основі готових модулів Drupal. Основним конкурентним недоліком, який поки що заважає їй обігнати свого головного конкурента, CMS Joomla, є менш інтуїтивно зрозумілий і зручний інтерфейс. Іншими недоліками системи є традиційне для портальних CMS високе навантаження на сервер і надмірність функціоналу для сайтів-візиток (таблиця 2.4).

Таблиця 2.4

#### Можливості Drupal

<b>Характеристики CMS Drupal</b>	<b>Значення</b>
Тип CMS	Портальна
Вартість	Безкоштовно
Навантаження на сервер	Високе
Складність доопрацювання	Середня
Складність адміністрування	Середня

Варто зазначити, що визначальними ознаками порівняння систем управління контентом є: наявність вбудованого редактора зображень, можливість скасування змін, наявність кошика для відновлення об'єктів, копіювання об'єктів, гнучкість налаштувань, багатокористувацькість, багатомовність.

На основі наведених вище визначальних ознак можна розробити порівняльну таблицю деяких поширених CMS (таблиця 2.5).

Таблиця 2.5

Порівняння CMS за специфічними особливостями, що визначають функціональність

	Joomla	NetCat	Saitistik a	Drupal	Бітрікс
<b>Вбудований редактор зображень</b>	+	-	+	-	-
<b>Скасування внесених змін</b>	+	-	+	+	-
<b>Кошик (відновлення об'єктів)</b>	+	-	-	+	-
<b>Копіювання системних об'єктів</b>	+	-	+	+	+
<b>Гнучкі налаштування</b>	+	+	-	+	+
<b>«Дружні» URL-адреси (зрозумілі адреси сторінок)</b>	+	+	+	+	+
<b>Мультисайт</b>	+	+	+	+	+
<b>Багатомовність</b>	+	+	+	+	+
<b>Програмна платформа</b>	PHP5	PHP4	Perl	PHP5	PHP4

Оскільки зараз у світі існує багато CMS-систем, їх варто розділити на безкоштовні, комерційні та пропріетарні. А потім порівняти їх (Додаток Г) за наступними загальними критеріями:

- безпека та захист від вторгнень;
- складність впровадження та підтримки;
- продуктивність;
- стандартний і розширений функціонал;
- вартість.

На основі вищезазначених критеріїв розроблено приблизний рейтинг використання CMS у світі (рис. 2.1).

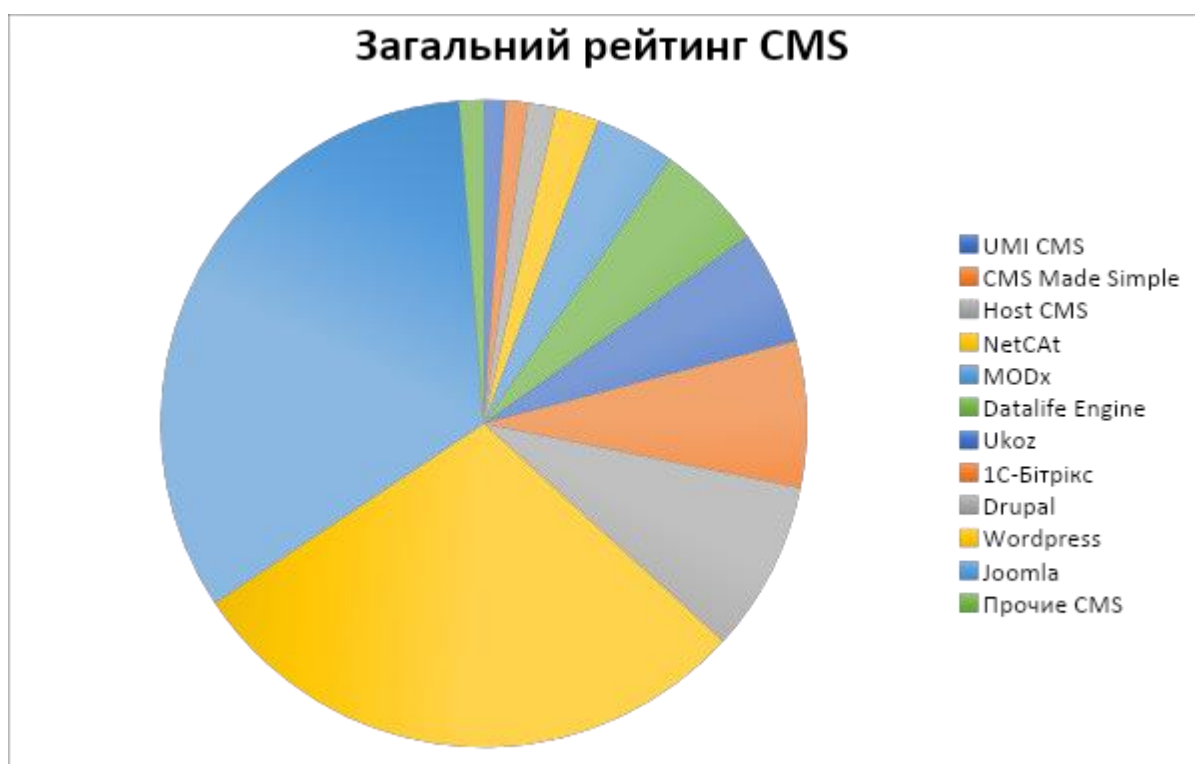


Рис. 2.1 Загальний рейтинг CMS

## Висновок до розділу 2

Вибір системи управління контентом – це складний процес, який вимагає врахування багатьох факторів, перерахованих вище. Перед створенням і вибором важливо врахувати, що потрібно в розробці і для чого буде використовуватися сайт. Це визначає очікувану модульність системи, а також її вартість.

У розробленій системі управління контентом візуалізація не замінює прості способи створення об'єктів: існують способи введення даних вручну. Шаблонна система побудови сайту дозволить адаптувати для ІС практично будь-який дизайн. Для будь-якої сторінки можна встановити власні стилі оформлення. Система має модульну конструкцію.

Можна підключити або відключити будь-який модуль. Код написаний з чітким розділенням HTML і PHP. Це підвищує читабельність коду. Весь HTML розміщується в шаблонах.

Основною перевагою системи є те, що абсолютно весь контент зберігається в базі даних і редагується безпосередньо з адміністративної панелі, що дозволяє легко оновлювати інформацію на сайті без редагування будь-яких файлів. Ця особливість дозволяє легко перенести систему з одного хостингу на інший (наприклад, локальний) – слід просто скористатися пунктом меню «Зберегти базу даних» на одному комп'ютері і виконати «Відновити базу даних» на іншому.

Система дозволяє створювати деревоподібну структуру сайту з необмеженою кількістю рівнів вкладеності та будь-яким контентом.



### РОЗДІЛ 3

## ПРОЕКТУВАННЯ МОДЕЛІ СИСТЕМИ НА ОСНОВІ ШАБЛОНІЗАТОРІВ

### 3.1 Загальне поняття та теоретичні аспекти використання шаблонізаторів.

Програмне забезпечення для створення шаблонів – це програмне забезпечення, яке дозволяє використовувати html-шаблони для створення кінцевих html-сторінок.

Основна мета використання інструментів розробки шаблонів – відокремити представлення даних від виконуваного коду. Це часто необхідно для того, щоб програміст і верстальник могли працювати паралельно. Використання інструментів розробки шаблонів часто покращує читабельність і зовнішній вигляд коду, коли проект виконується однією людиною.

CMS отримує інформацію для відображення (наприклад, текст статті) з бази даних. Для відображення інформації у форматі HTML використовується механізм шаблонів.

Шаблон – це файл дизайну сторінки, створений за допомогою спеціальної мови. Як правило, це HTML-код, розмічений певним чином, який вказує, де вставити заголовок сторінки, де вставити основний текст, а де меню або інші елементи, взяті з бази даних. Найпростіший варіант – створити шаблон на PHP, але існують і більш складні рішення.

Движок шаблонів має власну досить просту мову, з якою може впоратися програміст, який не знайомий з PHP. Шаблони конвертуються в PHP-файл, а потім просто виконуються інтерпретатором PHP. Для підвищення продуктивності можна скомпілювати шаблон на PHP, оскільки він буде рідко змінюватися.

Кафедра КІТ (47)				НАУ 23.26.51 000 ПЗ			
<i>Виконав</i>	<i>Янголь В.Є.</i>			<b>ПРОЕКТУВАННЯ МОДЕЛІ СИСТЕМИ НА ОСНОВІ ШАБЛОНІЗАТОРІВ</b>	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Моденов Ю.Б.</i>					57	15
<i>Консульт.</i>					<b>УС-211М 122</b>		
<i>Н-котрол.</i>	<i>Райчев І.Е</i>						

### 3.2. Класифікація шаблонізаторів для створення CMS

Умовно поділимо шаблонізатори на 4 групи. В якості ознак поділу будемо використовувати синтаксис мови шаблонів та логіку представлення. Для прикладу розглянемо найпопулярніші шаблонізатори (рис. 3.1).

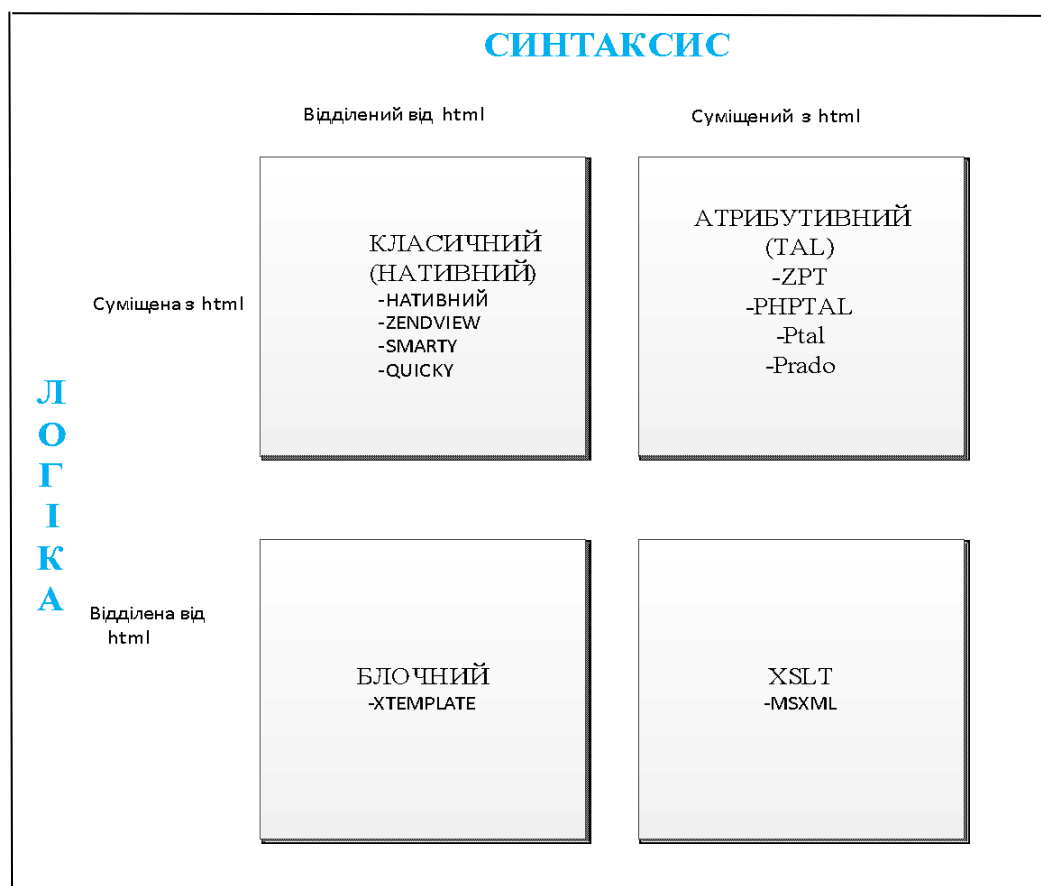


Рис. 3.1. Класифікація шаблонізаторів

Логіка представлення та синтаксис були обрані в якості класифікації механізмом шаблонізатору.

Логіка:

- HTML-логіка – класична схема, в якій команди мовою шаблону вбудовуються безпосередньо у файл шаблону. Так працюють нативні шаблони, Smarty-шаблони та шаблони атрибутів. У шаблонах

атрибутів логіка розміщується безпосередньо у файлі HTML-коду (або, навіть, безпосередньо в тегах).

- логіка відокремлена від HTML – логіка представлення знаходиться в окремому файлі. Так працюють шаблони блоків і XSLT.

Синтаксис:

- Синтаксис, окремий від HTML – класичний стиль шаблонізатора. Движок шаблонів має власну мову, відмінну від HTML. Це може бути мова, якою працює система (PHP, Java, Python, .Net) або мова самого шаблонізатора (Smarty, Quicky, Xtemplate).
- синтаксис пов'язаний з HTML – команди задаються як HTML-теги або атрибути. Це XSLT або шаблонізація атрибутів.
- Майже всі відомі програми для створення шаблонів можна класифікувати за цією схемою. Для прикладу було обрано найпопулярніші програми для створення шаблонів.

Таблиця 3.1

#### Класичні інструменти для створення шаблонів

<b>Нативний</b>	Відправна точка в порівнянні. Коли мова йде про те, що шаблонний движок не використовується, – це означає, що використовується нативний движок
<b>ZendView</b>	Нативний шаблонний движок у чистому вигляді
<b>Smarty</b>	Найпопулярніше класичне програмне забезпечення для створення шаблонів
<b>Quicky</b>	Аналог Smarty, який набирає популярності

Таблиця 3.2

## Шаблони атрибутів

<b>ZPT</b>	Є, мабуть, найкращим прикладом для наслідування. Движок шаблонів, що використовується у сервері Zope
<b>PHPTAL</b>	Є еквівалентом ZPT для PHP. Входить до складу Drupal
<b>Ptal</b>	Рушій шаблонів атрибутів для PHP.
<b>Prado</b>	Шаблонізатор компонентів. Основна логіка полягає у виклику методів контролера безпосередньо в HTML. Хоча інші шаблонізатори також дозволяють це робити

Таблиця 3.3

## Машини для виготовлення блоків

<b>XTemplate</b>	Дуже популярний блоковий шаблон
------------------	---------------------------------

Таблиця 3.4

## XSLT

<b>MSXML</b>	Найпопулярніший движок шаблонів XSLT
--------------	--------------------------------------

### 3.3. Текстовий візуальний редактор CKeditor

CKeditor – це безкоштовний WYSIWYG-редактор, який можна використовувати на веб-сайтах. До версії 3.0 він називався FCKeditor («FCK» на честь творця редактора Фредеріко Калдейра Кнаббена), але змінив свою назву, щоб уникнути асоціації з популярним англійським лайливим словом (коли проект був запущений, бразилець Фредеріко не врахував це). Особливістю редактора є його невеликий розмір без необхідності встановлення на стороні клієнта. Редактор інтегрований з:

- ASP.Net
- ASP
- PHP

- Java
- JavaScript
- Active-FoxPro
- Lasso
- Perl
- Python
- Opera 9.50+.
- Safari 3.0+.
- Firefox 1.5+.
- Camino 1.0+ (Mac)
- Internet Explorer 5.5+.
- Google Chrome.

### **3.4. Вибір програмної та апаратної платформи**

#### **3.4.1. Вибір СУБД**

Багато веб-сайтів зараз використовують базу даних MySQL як сховище інформації. Її розробником є компанія MySQL AB.

MySQL – це система управління базами даних. База даних – це структурована колекція даних. Це може бути що завгодно: від простого списку майбутніх покупок до переліку експонатів у художній галереї або величезної кількості інформації в корпоративній мережі.

Для запису, відбору та обробки даних, що зберігаються в комп'ютерній базі даних, необхідна система управління базами даних, якою є програмне забезпечення MySQL.

Оскільки комп'ютери вміють обробляти великі обсяги даних, управління базами даних відіграє ключову роль в ІТ. Таке управління може бути реалізоване різними способами – як окремими інструментами, так і у вигляді коду, включеного в інші додатки.

MySQL також керує реляційними базами даних. У реляційній базі даних дані зберігаються не разом, а в окремих таблицях, що підвищує швидкість і гнучкість. Таблиці пов'язані між собою за допомогою зв'язків, що дозволяє об'єднувати дані з декількох таблиць при виконанні запиту. SQL, як частину MySQL, можна описати як мову структурованих запитів і найпопулярнішу стандартну мову, що використовується для доступу до баз даних.

Програмне забезпечення MySQL має відкритий вихідний код. Відкритий код означає, що будь-хто може використовувати та модифікувати його. Таке програмне забезпечення можна отримати через Інтернет і використовувати безкоштовно. Водночас будь-який користувач може вивчити вихідний код і модифікувати його відповідно до своїх потреб. Спочатку сервер MySQL був розроблений для управління великими базами даних з метою забезпечення

більшої швидкості в порівнянні з існуючими аналогами. Протягом років сервер успішно використовувався в промислових середовищах з високими вимогами.

Програмне забезпечення MySQL – це клієнт-серверна система, яка включає в себе багатопотоковий SQL-сервер, що забезпечує підтримку різних комп'ютерів з базами даних, а також кілька різних клієнтських додатків і бібліотек, інструменти адміністрування та широкий спектр інтерфейсів прикладного програмування (API).

MySQL Server також постачається у вигляді багатопотокової бібліотеки, яку можна підключати до користувацьких додатків для створення компактного, швидшого та більш керованого продукту. Також доступна велика кількість стороннього програмного забезпечення MySQL.

Програмне забезпечення MySQL – це дуже швидкий, багатопотоковий, багатокористувацький, надійний сервер баз даних SQL (SQL – це мова структурованих запитів). Сервер MySQL призначений як для критично важливих виробничих систем з великими робочими навантаженнями, так і для вбудовування в програмне забезпечення для масового розповсюдження.

MySQL – це рішення для малих і середніх додатків. Він входить до складу LAMP. Зазвичай MySQL використовується як сервер, доступний локальним або віддаленим клієнтам, але дистрибутив містить внутрішню серверну бібліотеку, яка дозволяє вбудовувати MySQL в окремі додатки.

Гнучкість MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибирати між таблицями MyISAM, які підтримують повнотекстовий пошук, і таблицями InnoDB, які підтримують транзакції на рівні окремих записів. Завдяки відкритій архітектурі та ліцензії GPL та MySQL, постійно додаються нові типи таблиць.

У 2008 році було розроблено новий інтерфейс SQL, але API було успадковане від mSQL.

MySQL портовано на багато платформ: AIX, BSDi, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS,

SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2003 і Windows Vista. Існує також порт MySQL для OpenVMS.

MySQL має API для C, C++, Java, Perl, PHP, Python, Ruby, Smalltalk і Tcl, бібліотеки для мов платформи .NET і забезпечує підтримку ODBC за допомогою драйвера MyODBC ODBC.

MySQL 5.0 включає такі нововведення: збережені процедури і функції; обробка помилок; курсори; тригери; подання; інформаційна схема (так званий системний словник, що містить метадані).

MySQL – це невеликий, компактний, багатопотоковий сервер баз даних. MySQL характеризується високою швидкістю, стабільністю та простотою використання.

MySQL була розроблена компанією ТсХ для внутрішніх потреб, які полягали у швидкій обробці дуже великих баз даних. Компанія стверджує, що використовує MySQL з 1996 року на сервері з більш ніж 40 базами даних, що містять 10 000 таблиць, понад 500 з яких мають більше 7 мільйонів рядків. MySQL ідеально підходить для малих і середніх додатків. Вихідний код сервера скомпільований для різних платформ. Найбільш повно можливості сервера проявляються на Unix-серверах, де є підтримка багатопоточності, що дає значний приріст продуктивності. У версії для Windows MySQL можна запускати як службу Windows NT або як простий процес у Windows 95/98.

Сервер MySQL безкоштовний для некомерційного використання.

MySQL підтримує мову запитів ANSI 92 SQL і має багато розширень до цього стандарту, яких немає в жодній іншій базі даних.

Короткий список функцій MySQL:

- підтримується необмежена кількість користувачів, які одночасно працюють з базою даних;
- кількість рядків у таблицях може досягати 50 мільйонів;



- швидке виконання команд. Можливо, MySQL є найшвидшим сервером з усіх існуючих;
- проста та ефективна система безпеки.

MySQL дійсно є дуже швидким сервером, але для того, щоб досягти цього, розробникам довелося пожертвувати деякими вимогами до реляційних баз даних. У MySQL немає певного функціоналу:

- обробка транзакцій не реалізована. Замість цього пропонується використовувати Lock / Unlock Table;
- немає підтримки тригерів та збережених процедур;
- немає підтримки зовнішніх ключів.

За словами розробників, ці пункти дозволили досягти високої продуктивності. Їх реалізація значно знижує швидкість роботи сервера. Ці особливості не є критичними при розробці веб-додатків, що в поєднанні з високою продуктивністю і низькою ціною дозволило серверу стати дуже популярним.

Основними особливостями MySQL є

- MySQL написана на мовах C та C++ і була протестована на багатьох різних компіляторах.
- він працює на різних платформах і підтримує різні операційні системи.
- повністю багатопотоковий з використанням потоків ядра. Це означає, що, якщо є можливість, можна легко організувати роботу з декількома процесорами.
- дуже швидкі дискові таблиці на основі B-дерев зі стисненням індексів.
- дуже швидка система базується на потоках виділення пам'яті.
- дуже швидке з'єднання за допомогою оптимізованого методу багат шарового з'єднання за один прохід.
- кеш-таблиці в пам'яті, які використовуються як тимчасові таблиці.
- функції SQL реалізуються за допомогою добре оптимізованої бібліотеки класів, щоб вони виконувалися якомога швидше. Зазвичай після ініціалізації запиту пам'ять не виділяється.

### 3.4.2. Вибір мови програмування

Оскільки PHP-MySQL є крос-платформним, це означає, що під час роботи під Windows можна створювати додатки, призначені для роботи під Unix [8]. Крім того, PHP може працювати як зовнішній CGI-процес, як простий інтерпретатор сценаріїв або як модуль, що підключається до веб-сервера Apache або IIS. І, нарешті, позаяк продукт є результатом спільної розробки, існує величезна кількість документації та списків розсилки, до яких можна звернутися, якщо виникнуть будь-які питання.

В якості мови програмування для клієнта підключення до бази даних було обрано PHP. PHP – це мова сценаріїв, вбудована в HTML, яка інтерпретується і виконується на сервері. Основна відмінність від CGI-скриптів, написаних іншими мовами, такими як Perl або C, – полягає в тому, що в CGI-програмах ви самі пишете HTML-вивід, тоді як з PHP ви вбудовуєте свою PHP-програму в готову HTML-сторінку за допомогою відкриваючих і закриваючих тегів. Різниця між PHP і JavaScript полягає в тому, що PHP-скрипт виконується на сервері, а клієнту надсилається лише результат роботи, тоді як у JavaScript код повністю надсилається на клієнтський комп'ютер і лише там виконується.

PHP може робити все, що можуть робити CGI-програми. Обробка даних форм, генерація динамічних сторінок, отримання та надсилання файлів cookie, завантаження файлів, створення файлів і папок, малювання динамічних зображень, надсилання електронних листів тощо.

Крім того, PHP включає підтримку декількох баз даних, що значно розширює можливості для написання по-справжньому динамічних веб-додатків. На додаток до всього цього, PHP розуміє протоколи IMAP, SNMP, NNTP, POP3 і HTTP, а також має можливість працювати з сокетом і спілкуватися за допомогою інших протоколів.

Повноцінний веб-сайт повинен підтримувати певний рівень інтерактивності з користувачем: всілякі голосування, пошук інформації, продаж товарів,

конференції, форуми тощо. Традиційно все це реалізується за допомогою CGI-скриптів, написаних на мові Perl. Але CGI-скрипти дуже погано масштабуються.

Це означає, що кожен новий виклик CGI вимагає від ядра запуску нового процесу, що забирає процесорний час і значно споживає оперативну пам'ять. Велика кількість хакерських атак на сервери спирається на численні виклики CGI, що призводить до навантаження на ядро процесора і оперативну пам'ять.

PHP пропонує інший варіант – він працює як частина веб-сервера і в цьому схожий на ASP від Microsoft. Синтаксис PHP легко читається і загалом зрозумілий. Мова не має суворої типізації даних і не потребує дій для виділення (або звільнення) пам'яті.

Програми, написані на PHP, досить прості для розуміння. Код, написаний на PHP, легко читати і розуміти візуально, на відміну від програм на Perl. Крім того, він характеризується досить високою швидкістю роботи. Програми, написані на PHP, чудово працюють на Solaris, Linux та багатьох інших операційних системах, окрім Windows.

### **3.4.3. Вибір веб-сервера**

Apache є найпоширенішим веб-сервером в Інтернеті з коефіцієнтом використання 80%. Існують веб-сервери від Microsoft і Netscape, але вони платні і не мають таких можливостей, програмної та інформаційної підтримки, як Apache. Крім того, Apache є повністю безкоштовним.

Apache складається з ядра, яке надає найпростіші функції, що завжди присутні у стандартній поставці, та модулів, які розширюють його можливості. Деякі модулі включено до дистрибутиву за замовчуванням. Інші необхідні модулі можуть бути додані до Apache пізніше. Наприклад, таким чином реалізується підтримка SSL або мов програмування.

Apache має досить гнучкі можливості конфігурації. Такі завдання вирішуються шляхом редагування текстових конфігураційних файлів. Це

твердження справедливе і для багатьох додаткових модулів. Такі файли складаються з набору директив і відповідних їм опцій, а також коментарів.

Основними перевагами цього сервера є надійність і відносна стійкість до злому. Зокрема, він дозволяє підключати зовнішні модулі для надання даних, використання бази даних для аутентифікації користувачів, модифікації повідомлень про помилки тощо. Apache легко переноситься на інші платформи.

Наразі доступні версії для Windows, Unix та Mac. Ще однією перевагою є надзвичайна потужність Apache. Підтримка CGI, PHP, SSI та контроль доступу – це лише невелика частина пропонованих послуг. Крім того, сервер можна придбати та встановити абсолютно безкоштовно. Можливо, саме тому він так широко розповсюджений.

#### **3.4.4. Вибір апаратної платформи**

Вибір апаратної платформи тісно пов'язаний з програмною системою. По-перше, апаратне забезпечення повинно підтримувати та забезпечувати роботу операційної системи та інструментів розробки.

Для коректної роботи інформаційної системи на цільовому комп'ютері має бути встановлене наступне програмне забезпечення:

- Операційна система Microsoft Windows
- База даних Microsoft SQL
- PHP 5
- Веб-браузер Internet Explorer версії 5.5 або новішої, Mozilla Firefox

Програмне забезпечення накладає наступні вимоги до апаратної конфігурації:

*Мінімальні вимоги до обладнання:*

Процесор: 3,2 ГГц процесор сімейства Intel Pentium IV

Мережевий інтерфейс: Ethernet

Оперативна пам'ять: 512 МБ

Місце на жорсткому диску: мінімум 1 ГБ

*Рекомендована конфігурація обладнання:*

Процесор: Процесор Intel Pentium core 2 duo 1800 МГц або еквівалент.

Мережевий інтерфейс: Fast Ethernet

Оперативна пам'ять: 1 ГБ

Місце на жорсткому диску: 5 ГБ

Така система, як СУБД, дуже вимоглива до швидкості читання і запису на жорсткому диску, а також до обсягу даних, що зберігаються на ньому. Вибір такого обсягу дискового простору автоматично виключає застарілі та ненадійні моделі жорстких дисків.

Оскільки однією з умов функціонування системи є робота в мережі, для неї також потрібне мережеве обладнання: мережева карта з підтримкою протоколу Ethernet і швидкістю передачі даних 100 Мбіт/с.

Оскільки двоядерні процесори з'являються і розвиваються, огляд включає технічні характеристики для порівняння старих платформ з новими.

Сучасні веб-сайти стають все більш складними та логічно навантаженими. У минулому продуктивність таких додатків залежала в основному від швидкості відповідного SQL-сервера і від того, чи мав він достатньо потужну реалізацію драйвера доступу до SQL-сервера для обраної мови програмування. Це пов'язано з тим, що перше покоління веб-сайтів просто читало і записувало інформацію в бази даних. Користувачів було відносно небагато. Таким чином, час обробки SQL-запиту становив 70-90% від загального часу обробки HTTP-запиту.

Зі зростанням вимог до масштабованості (збільшення кількості користувачів) та логіки додатків значно зростають вимоги до мови програмування та середовища виконання. До цього слід додати той факт, що веб-сайти останнім часом перемістилися зі світу Інтернету у світ корпоративних додатків. Це знову підвищило вимоги до продуктивності середовища виконання.

Зараз існує багато різних мов програмування для створення веб-сайтів. Найпопулярніші з них – PHP, Perl, C# (DOT. NET), Java2. Класична технологія

ASP відходить у минуле, оскільки ASP DOT. NET практично замінила цю технологію на серверах Windows у нових додатках.

Для розробки в цьому дослідженні обрана мова PHP у поєднанні з сервером баз даних MySQL. Для цього необхідно встановити сервер Denwer. Середовищем розробки було обрано Macromedia Dreamweaver MX 2004.

### Висновок до розділу 3

Налаштування системи управління оновленнями сайту складається з декількох етапів:

- встановлення локального сервера;
- створення баз даних і користувачів в MySQL;
- вибір конфігурації та шаблону системи управління контентом;
- розробка та адаптація шаблону дизайну сайту;
- підключення до системи необхідних модулів і компонентів;
- створення системних матеріалів;
- створення пунктів системного меню.

Кожен з цих етапів має свої нюанси та особливості. Практична частина дослідження охоплює принципи роботи та основні етапи створення освітнього проекту з використанням сучасних веб-технологій.

Використовуючи основні методи, описані вище, досягається модульність системи, що дозволяє розширювати її функціональність

## РОЗДІЛ 4

### ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ WEB-САЙТОМ

#### 4.1 Розробка web-сайту

Першим завданням є розробка структурної схеми дизайну веб-сайту для використання в Інтернеті. Веб-сайт, що розробляється, буде мати наступні функції:

- гнучка, зручна система управління контентом;
- підтримка використання графічних вставок для посилення емоційно-ціннісної мотиваційної складової змісту та форми;
- користувачі також повинні мати можливість роздрукувати будь-яку сторінку сайту.

Розроблений веб-сайт «Science and Technologies» (рис. 4.1) не є веб-проектом великої складності. Він був розроблений з використанням шаблонного движка, тому основною метою дипломної роботи була розробка CMS для його управління, щоб чітко розділити інтерфейси управління та веб-сайту.

Сайт «Science and Technologies» є простим за структурою та функціональністю і складається з п'яти розділів: «Головна», «Про сайт», «Статті» (рис. 4.2), «Новини», «Контакти».

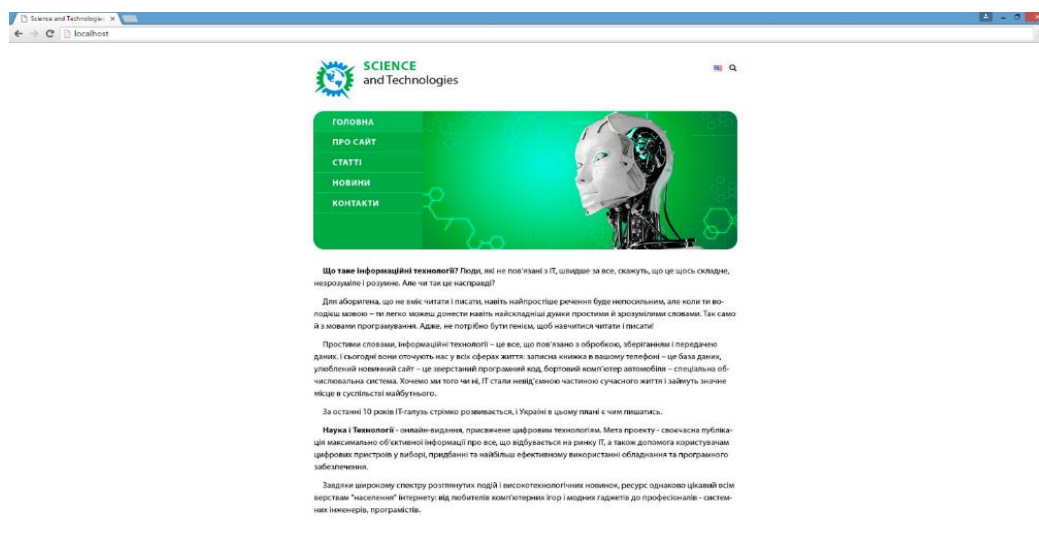


Рис. 4.1. Головна сторінка сайту

Кафедра КІТ (47)				НАУ 23.26.51 000 ПЗ			
Виконав	Янголь В.Є.			ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ WEB-САЙТОМ	Літера	Аркуш	Аркушів
Керівник	Моденов Ю.Б.					72	18
Консульт.					УС-211М 122		
Н-контрол.	Райчев І.Е.						



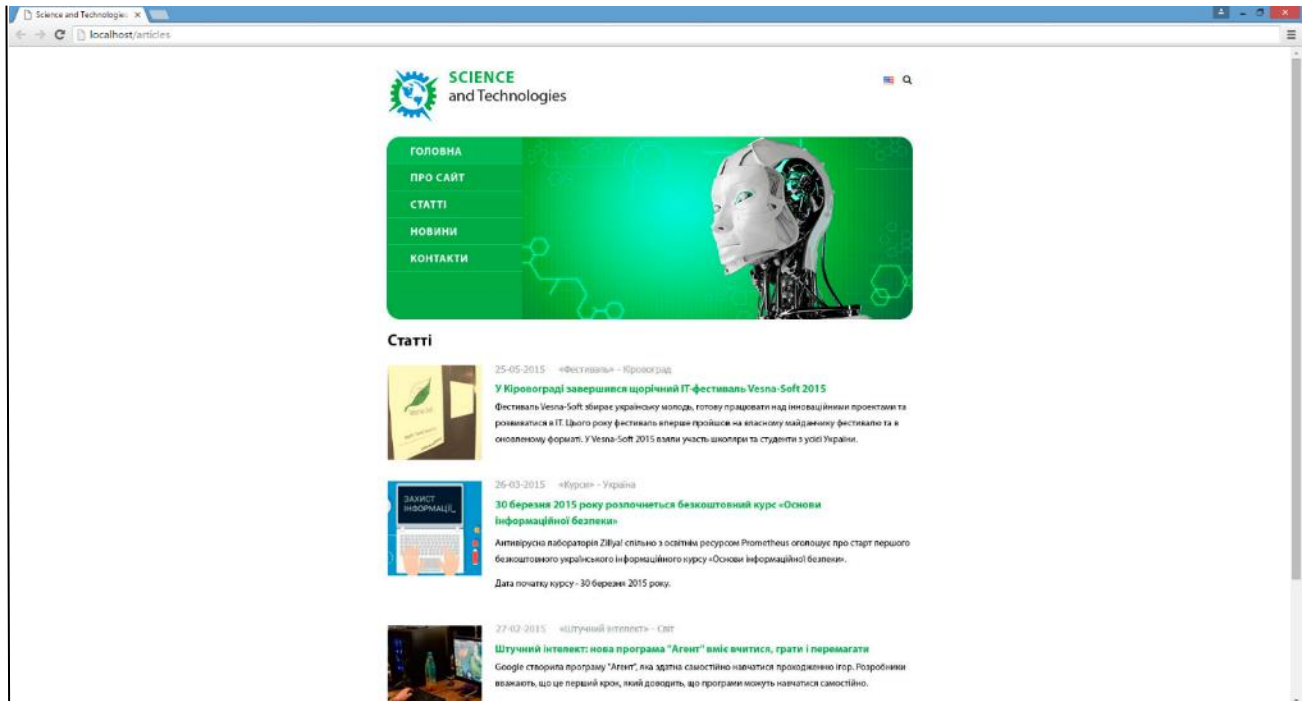


Рис. 4.2. Розділ «Статті»

## 4.2. Принципи роботи системи.

CMS складається з ядра, яке викликає модулі.

Модулі побудовані на основі моделі MVC (Додаток В). Для відокремлення коду від представлення буде використано скомпільований обробник шаблону (рис. 4.3, рис. 4.4). Всі виклики будуть спрямовані до основного скрипту – ядра (index.php). Адміністративна частина не відокремлена і включена в загальну структуру сайту. Ядро є основним поєднуваним елементом системи [9].

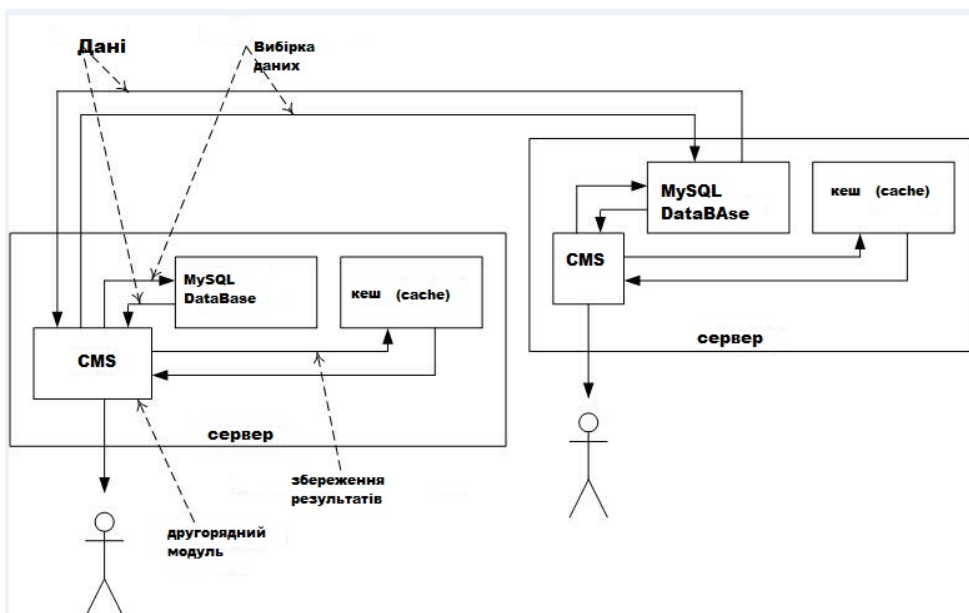


Рис. 4.3. Детальна схема принципу роботи системи управління контентом веб-сайту

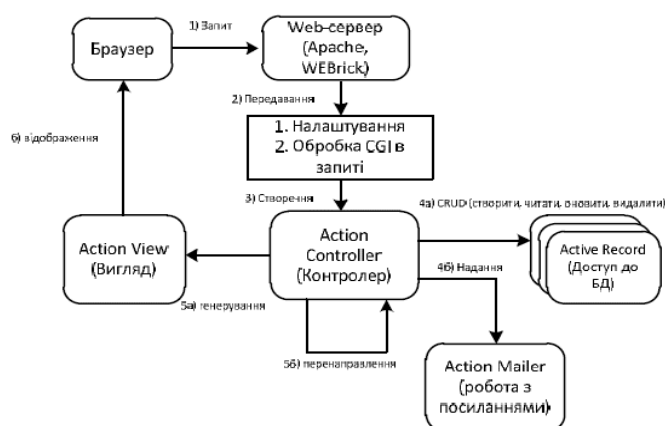


Рис. 4.4. Відображення шаблонів у браузері

### 4.3 Доступ до адміністративної панелі сайту та захист

Для доступу до панелі керування системою, починаючи з її встановлення на сервері, виконання базових налаштувань, слід перейти за наступною адресою: URL: */admin*

З'явиться вікно авторизації, в якому необхідно ввести логін і пароль адміністратора і натиснути кнопку «Вхід». Користувач побачить форму авторизації (рис. 4.5).

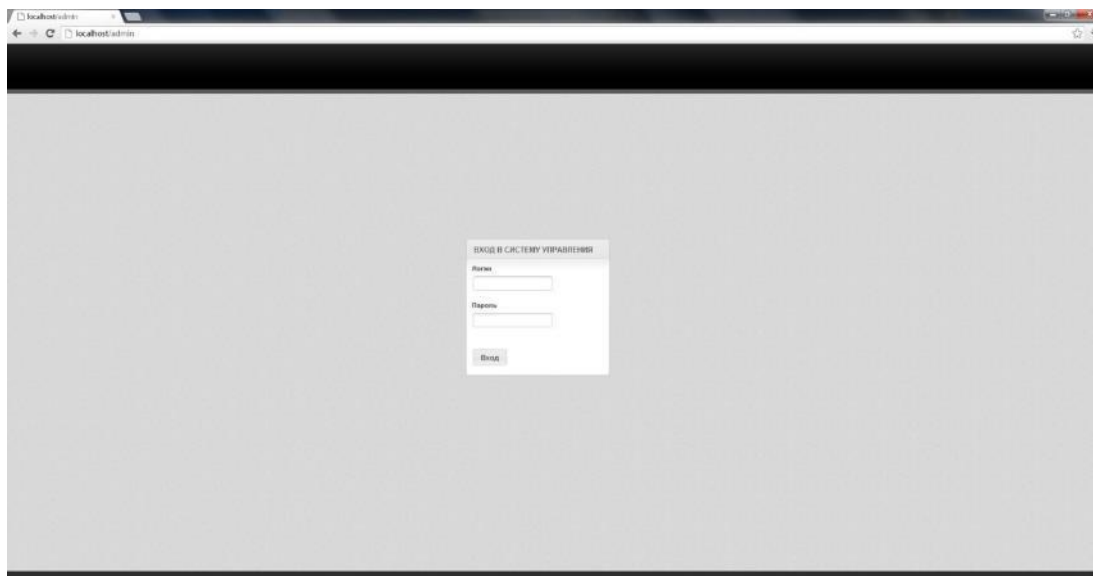


Рис. 4.5. Вхід до системи

Логін адміністратора створюється автоматично під час створення підсистеми. Після авторизації адміністратор отримує доступ до адміністративної панелі сайту.

Інтерфейс адміністративної частини являє собою перелік основних реалізованих модулів відповідно до структури та функціоналу сайту (рис. 4.6).

Потрапивши в розділ адміністрування, користувач може керувати системою через головне меню системи: «Управління меню», «Управління матеріалами», «Управління статтями», «Меню розробника», «Налаштування».

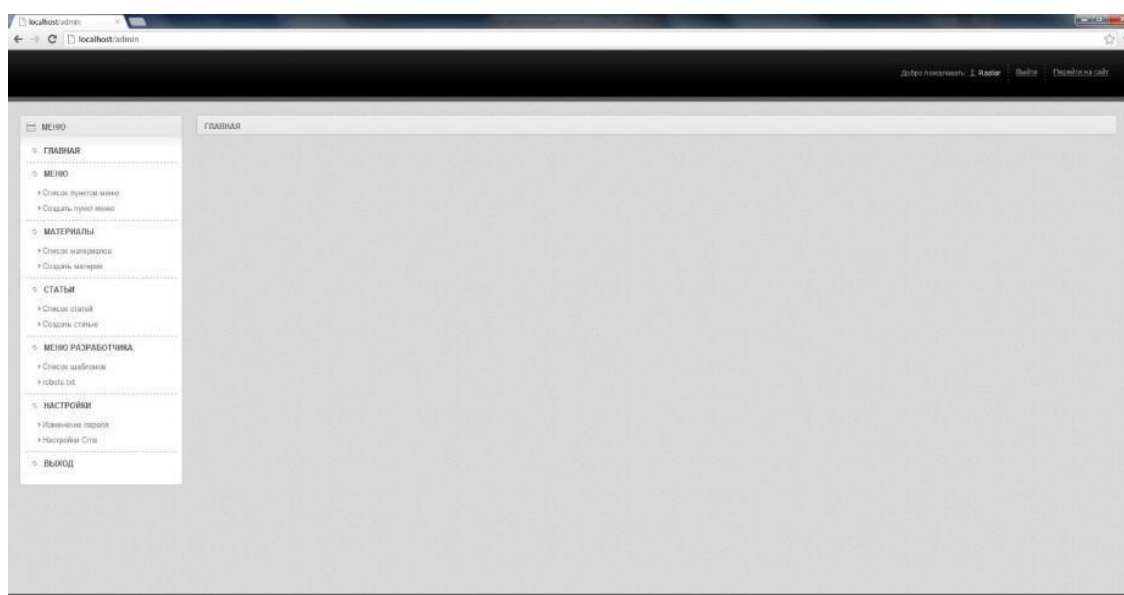


Рис. 4.6. Головна сторінка адміністративної панелі сайту

#### 4.4. Інструкція по користування текстовим редактором SKeditor

Перш ніж почати керувати контентом, слід ознайомитися з можливостями текстового редактора, який використовується для додавання та редагування контенту на сайті.

Розміщення текстової інформації за допомогою візуального текстового редактора подібне до MSWord. Щоб виділити текст напівжирним шрифтом або курсором, використовуються кнопки **B** та **I** відповідно. Підкреслювати текст не рекомендується, оскільки підкреслений текст асоціюється з посиланнями на веб-сторінці.

Під час копіювання та вставки тексту з веб-сторінки текст може містити форматування (відступи, шрифт, виділення), задане на сторінці-джерелі. Щоб видалити це форматування, слід скористатися кнопкою «Вставити лише текст» (рис. 4.7).



Рис. 4.7. Вставка тексту без форматування

Вирівнювання блоку тексту по лівому, правому або центральному краю. Для цього типу форматування на панелі візуального текстового редактора є кілька кнопок (рис. 4.8).



Рисунок 4.8 Вирівнювання текстового блоку

Не рекомендується вирівнювати текст по ширині блоку, оскільки такий тип форматування вважається поганим тоном через різну довжину рядків на моніторі і мобільному гаджеті, що, в свою чергу, сильно погіршує читабельність.

Під час форматування вирівнювання тексту важливо пам'ятати, що форматування застосовується до всього абзацу, в якому наразі знаходиться курсор.

Абзаци в тексті створюються автоматично. Коли вводиться перший блок тексту і натискається клавіша Enter, наступний блок тексту розміщується в окремому абзаці. Абзац – це контейнер, до якого застосовується форматування.

Щоб створити лівий відступ або одночасно лівий і правий відступи, слід скористатися кнопками «Збільшити відступ» і «Лапки» відповідно (рис. 4.9).



Рис. 4.9 Створення відступу для блоку тексту

Для встановлення заголовків використовується спадне меню «Форматування» (рис. 4.10). Під час використання цього типу форматування до виділеного фрагмента тексту буде застосовано обраний стиль заголовка n-го рівня.

Слід враховувати, що пошукові системи чутливі до тегів заголовків, які в даному випадку проставляються візуальним редактором. По можливості, в тегах заголовків повинен міститися текст зі словами, які планується використовувати для просування сайту в пошукових системах.



Рис. 4.10. Встановлення заголовків-тегів

Меню «Шрифт», що випадає (рис. 4.11), використовується для встановлення примусового накреслення шрифту для виділеної текстової області. За замовчуванням використовується гарнітура шрифту, задана в стилях сайту. Рекомендується змінювати гарнітуру шрифту, задану в стилях сайту, тільки в тих випадках, коли це дійсно необхідно, оскільки в цьому випадку текстовий блок може відрізнятись за зовнішнім виглядом від загального дизайну сайту.



Рис. 4.11. Зміна шрифту

Для зміни розміру виділеного тексту використовується спадне меню «Розмір» (рис. 4.12).



Рис. 4.12. Зміна розміру виділеного тексту

Щоб встановити колір тексту або колір фону виділеного тексту, слід скористатися кнопками «Колір тексту» і «Колір фону» на панелі візуального редактора (рис. 4.13).



Рис. 4.13. Налаштування кольору тексту та фону

Щоб створити список, використовуються кнопки: Створити / Видалити нумерований список (маркований список) (рис. 4.14). Кожен елемент нумерованого або маркованого списку повинен бути в окремому блоці-абзаці. Створивши кілька абзаців, які потрібно переформатувати в список, слід виділити їх і натиснути відповідну кнопку на панелі візуального редактору.



Рис. 4.14. Створення нумерованих або маркованих списків

Щоб вставити зображення на сторінку, слід скористатися кнопкою Вставити/Редагувати зображення (рис. 4.15).



Рис. 4.15. Вставлення зображення

Після натискання цієї кнопки з'являється відповідне вікно (рис. 4.16).

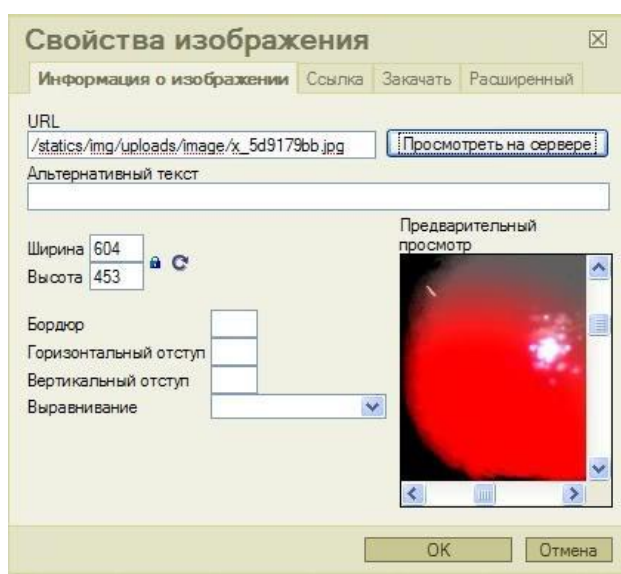


Рис. 4.16 Діалогове вікно Вставити зображення

Поле «URL» містить адресу зображення на сервері. Це поле заповнюється автоматично при завантаженні зображення на сервер (вкладка «Завантажити») або при виборі зображення з уже завантажених зображень (кнопка «Переглянути на сервері»).

Щоб почати роботу з зображенням, його слід обрати серед тих, що вже завантажені на сервер, або завантажити нове.

Зображення на сервері можна переглянути, натиснувши на однойменну кнопку. У цьому випадку з'явиться вікно для відображення структури файлів у папці завантаження зображень за замовчуванням.

## 4.5. Інформаційне наповнення та оновлення матеріалів

### 4.5.1. Створення та редакція пунктів меню

Користувач без навичок програмування зможе створювати, редагувати та видаляти пункти меню на сайті. Для цього в лівій частині адміністративної панелі в розділі «Меню» слід обрати підрозділ «Створити пункт меню» або «Перелік пунктів меню», залежно від необхідної дії на сайті.

Після переходу до підрозділу «Список пунктів меню» (рис. 4.17) на екрані з'явиться список усіх розділів сайту. Поруч з кожним розділом є три основні кнопки швидкого доступу: «Редагувати», «Видалити» і «Додати» новий запис.

«Редагувати» – кнопка, яка відкриває вибраний пункт меню для внесення змін (рис. 4.18).

«Видалити» – автоматично видаляє пункт меню в CMS.

«Додати» новий пункт – кнопка, яка відкриває інтерфейс для створення нового пункту меню (рис. 4.19).

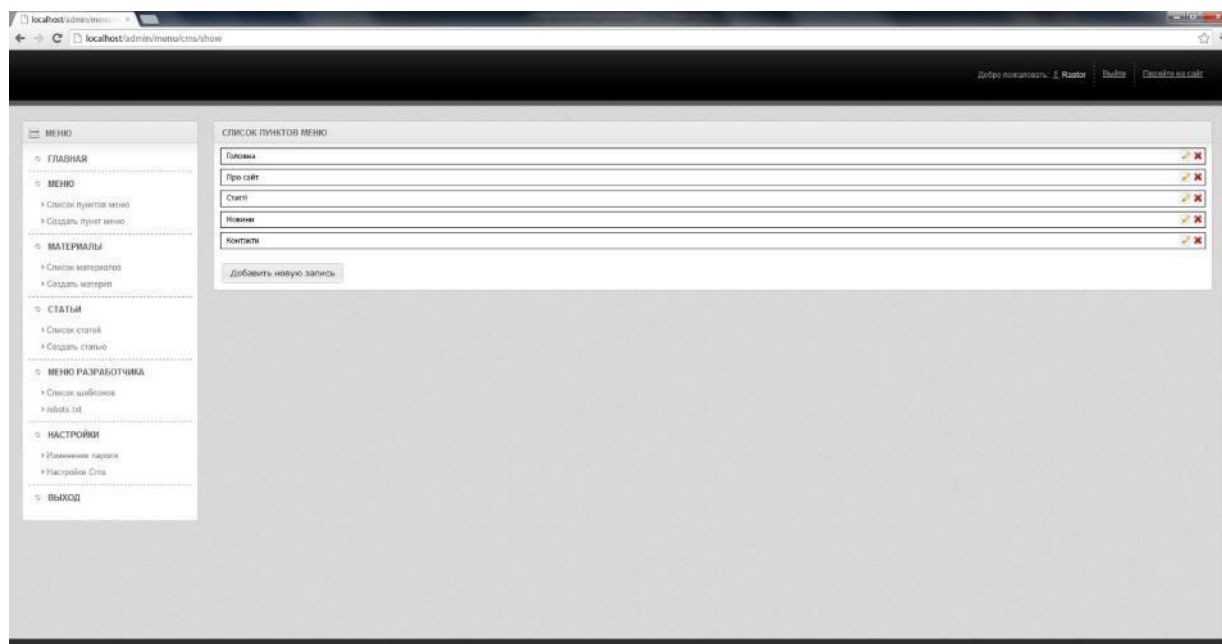


Рис. 4.17. Список пунктів меню



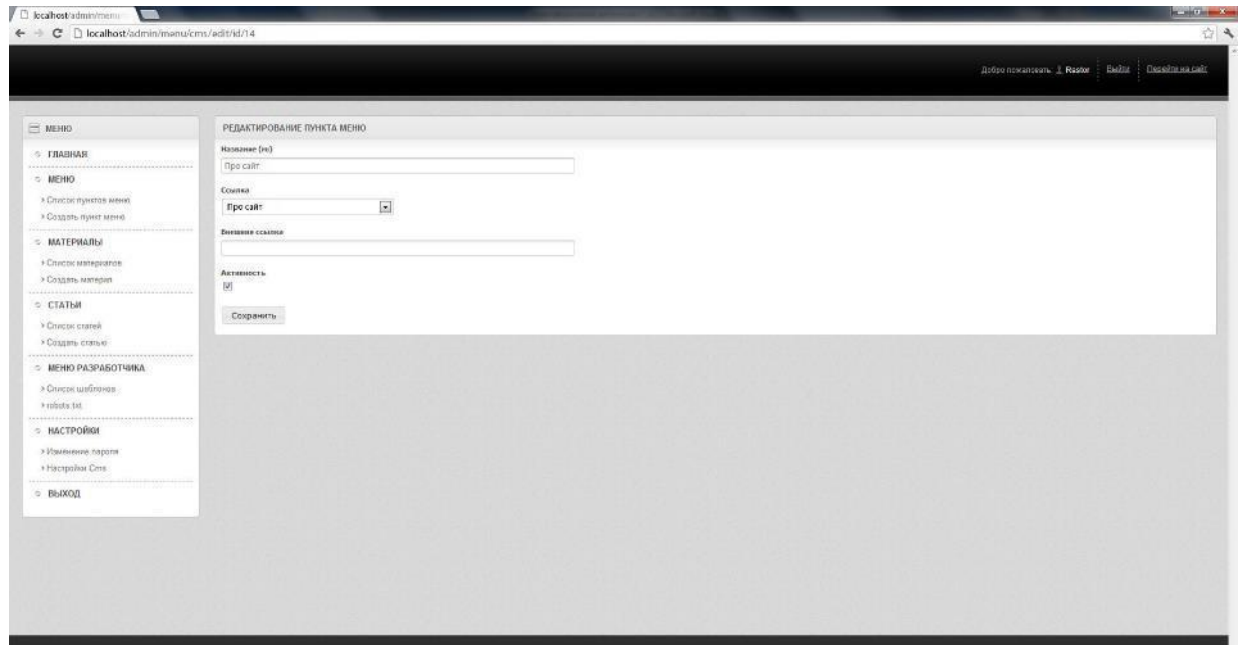


Рис. 4.18. Редагування пункту меню

Якщо потрібно створити новий пункт, необхідно перейти до підрозділу «Створити пункт меню», після чого відкриється форма для заповнення даних.

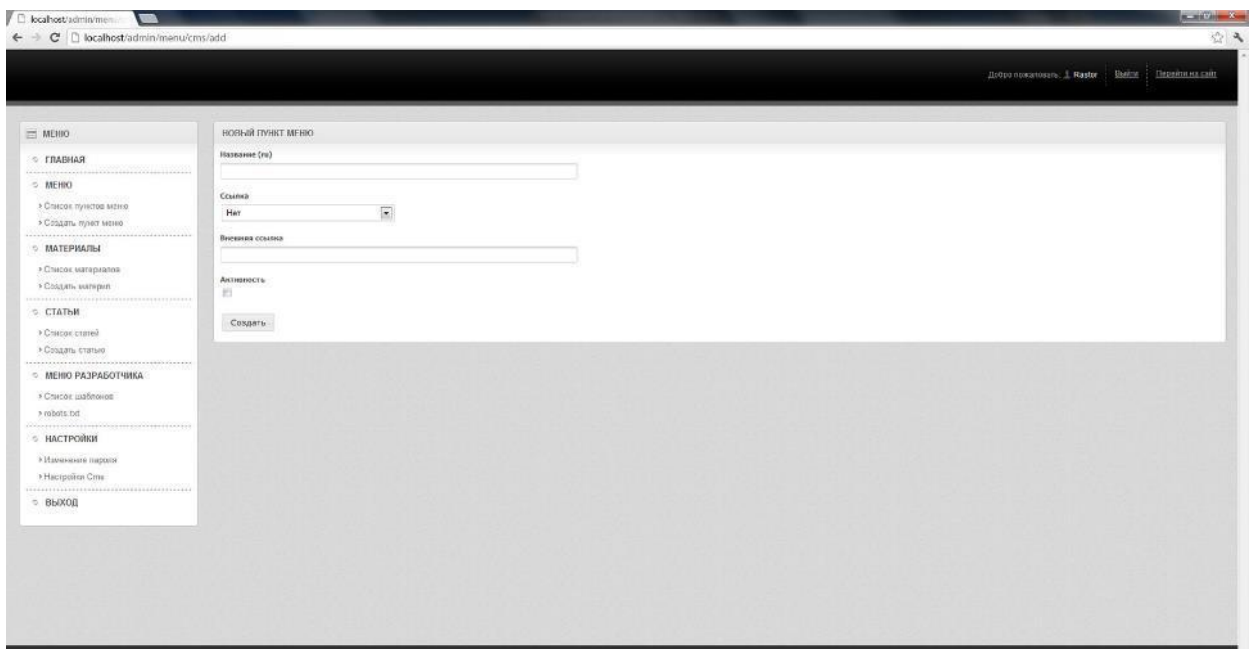


Рис. 4.19. Створення нового пункту меню

#### 4.5.2. Створення та редакція основних текстових матеріалів.

Модуль «Матеріали» реалізований в даній системі управління контентом для того, щоб дозволити додавати, редагувати і видаляти основні матеріали на сайті. Для того, щоб користувач міг оновлювати основні матеріали, в лівій частині панелі адміністрування CMS в розділі «Матеріали» користувач повинен обрати «Список матеріалів» або «Створити матеріал», якщо це необхідно.

Після того, як користувач переходить до підрозділу «Список матеріалів» (рис. 4.20), на екрані з'явиться список усіх матеріалів, доступних на сайті. Поруч з кожним розділом є п'ять основних кнопок швидкого доступу: «Переглянути», «Редагувати», «Видалити», «Видалити вибране», «Оновити».

«Переглянути» – кнопка, щоб відкрити вибраний матеріал для перегляду.

«Редагувати» – кнопка, яка відкриває вибраний матеріал для внесення змін (рис. 4.21).

«Видалити» – автоматично видаляє матеріал в CMS.

«Видалити вибране» – кнопка для видалення всіх вибраних матеріалів.

«Оновити» – автоматично оновлює матеріал при внесенні змін.

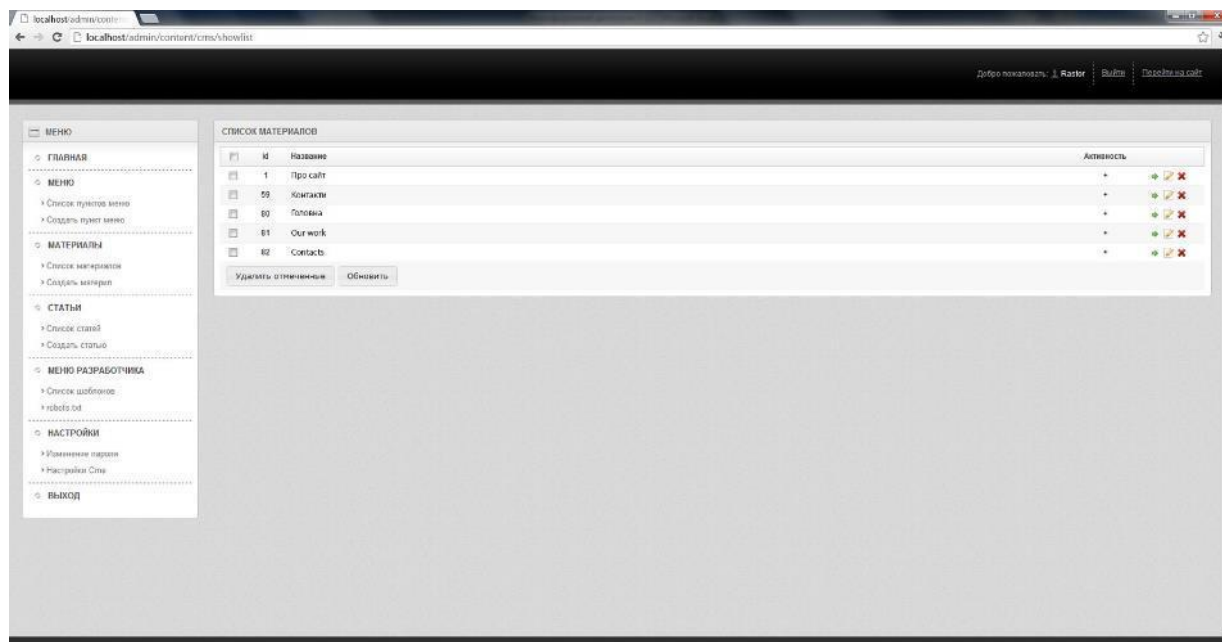


Рис. 4.20. Перелік матеріалів

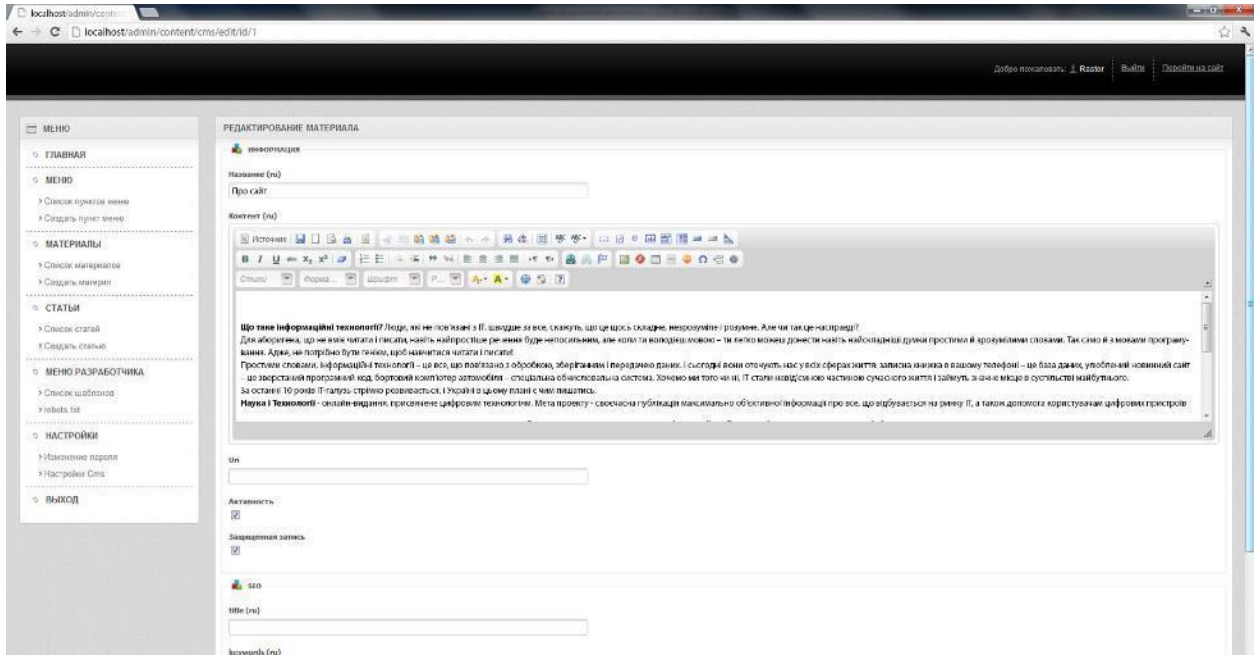


Рис. 4.21. Редагування матеріалу

Якщо слід створити новий пункт меню, потрібно перейти до підрозділу «Створити пункт меню», після чого відкриється форма для заповнення даних (рис. 4.22).

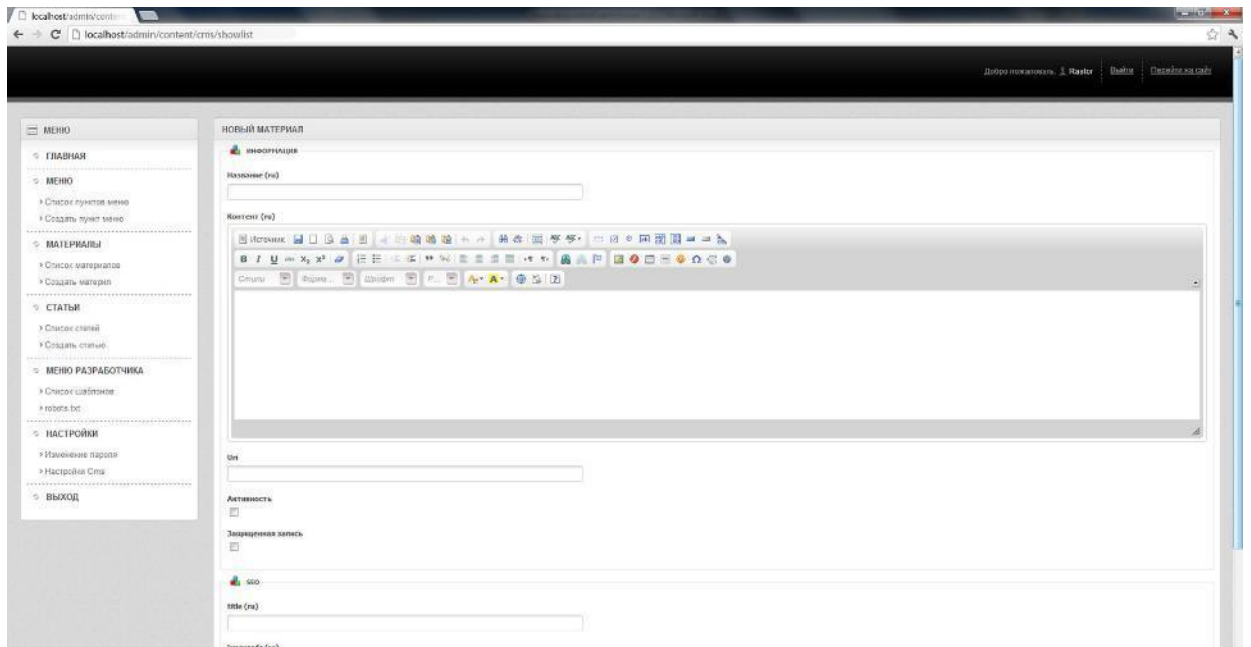


Рис. 4.22. Створення нового матеріалу

### 4.5.3. Створення та редакція статей

Модуль «Статті» призначений для додавання, редагування та видалення статей. Для того, щоб користувач міг оновлювати основний контент, в лівій частині панелі адміністрування CMS в розділі «Статті» необхідно обрати «Список статей» або «Створити статтю» відповідно.

Після переходу до підрозділу «Список статей» (рис. 4.23), на екрані з'явиться список усіх статей, розміщених на сайті. Поруч з кожним розділом є п'ять основних кнопок швидкого доступу: «Переглянути», «Редагувати», «Видалити», «Видалити вибране», «Оновити».

«Переглянути» – кнопка, щоб відкрити обрану статтю для перегляду.

«Редагувати» – кнопка, яка відкриває статтю для внесення змін (рис. 4.24).

«Видалити» – автоматично видаляє статтю в CMS.

«Видалити вибране» – кнопка для видалення всіх вибраних статей.

«Оновити» – автоматично оновлює статті при внесенні змін.

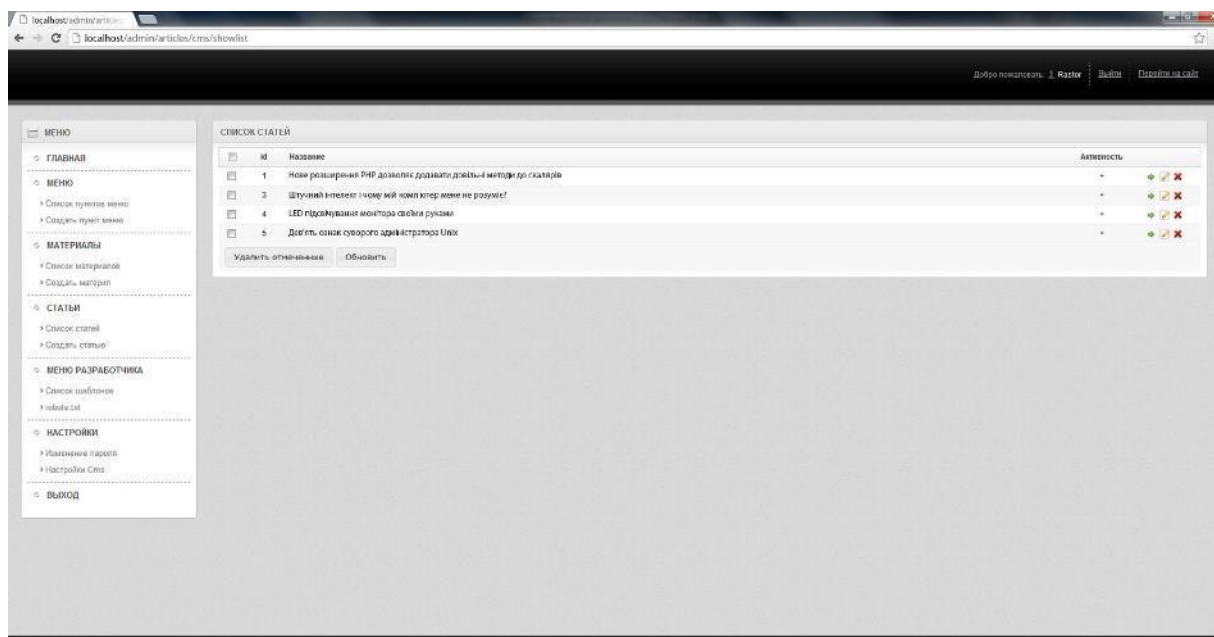


Рис. 4.23. Список статей

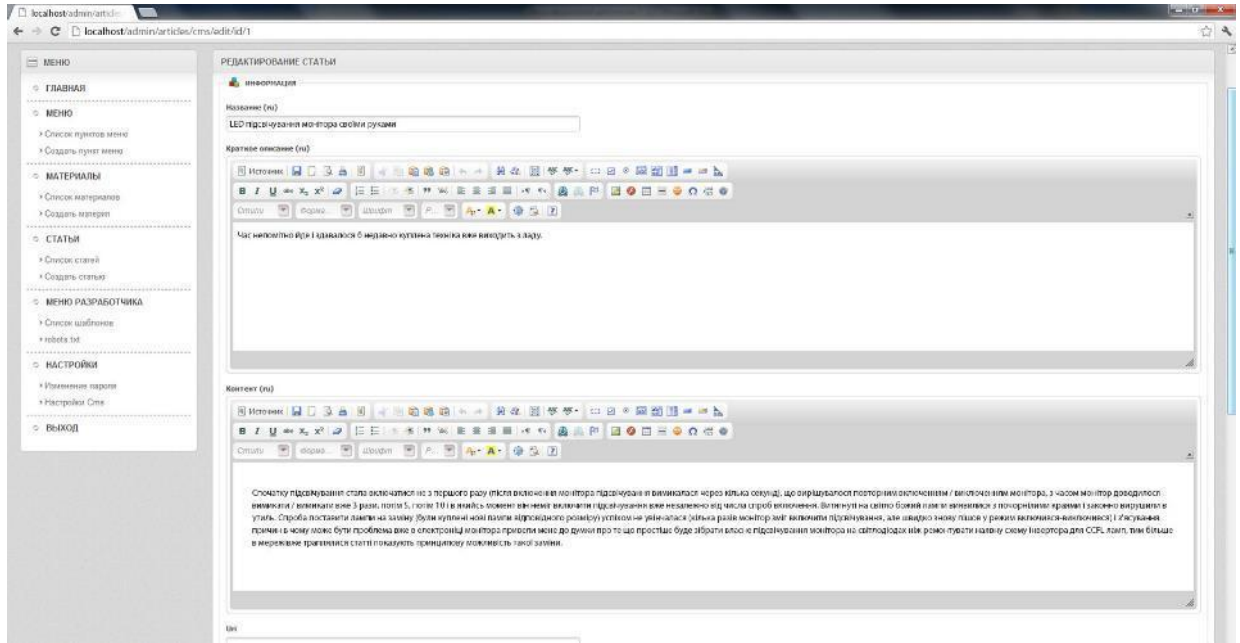


Рис. 4.24. Редагування статті

Якщо слід створити нову статтю на сайті, необхідно перейти в підрозділ «Створити статтю», після чого відкриється форма для заповнення даних (рис. 4.25).

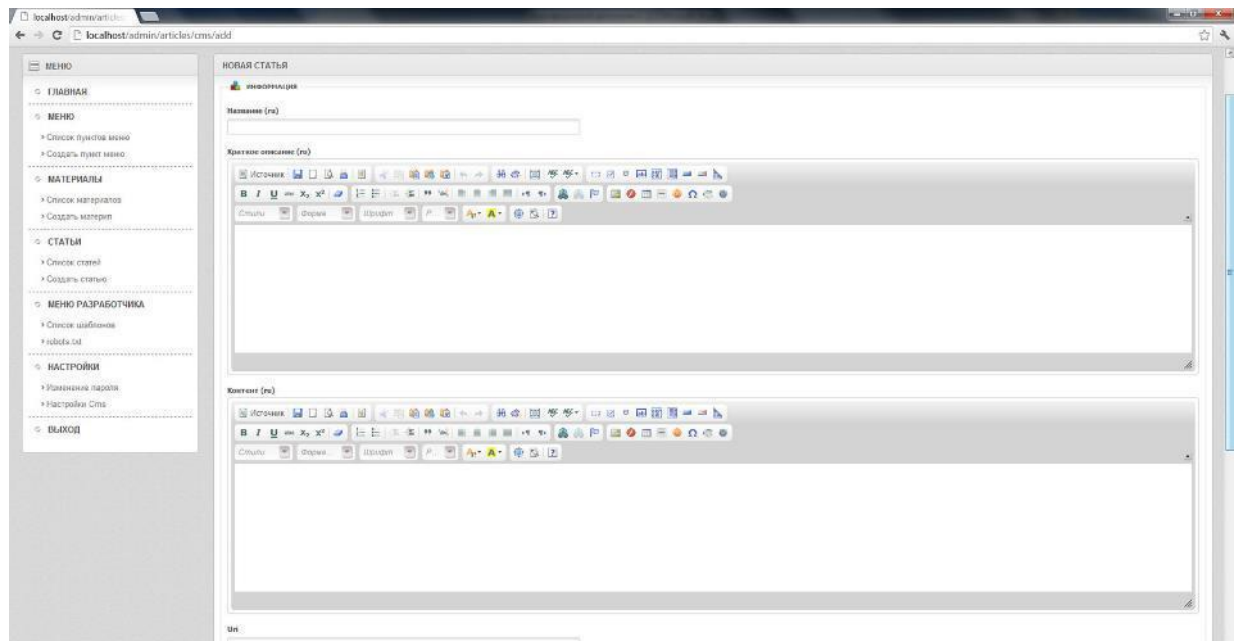


Рис. 4.25. Створення нової статті

#### 4.5.4. Основні налаштування та вихід з системи

Ця система управління, як і всі CMS-системи, має можливість змінювати пароль і налаштування самої системи. Для цього використовується модуль «Налаштування», який розділено на два підрозділи. Перший відповідає за зміну паролю (рис. 4.26), а другий – за загальні налаштування зовнішнього вигляду системи (рис. 4.27).

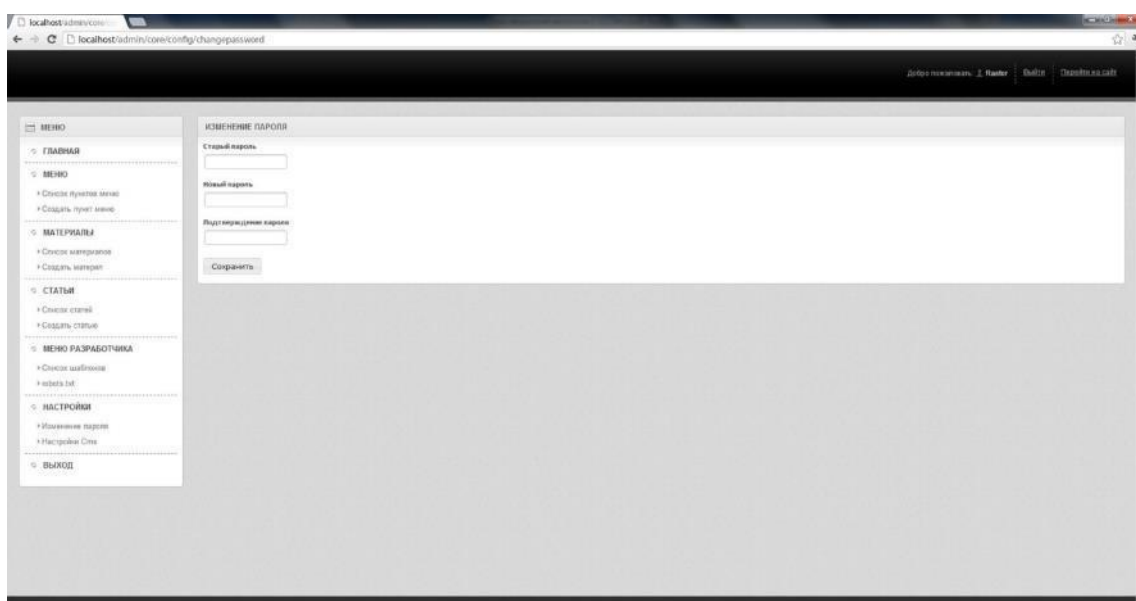


Рис. 4.26. Зміна паролю в системі

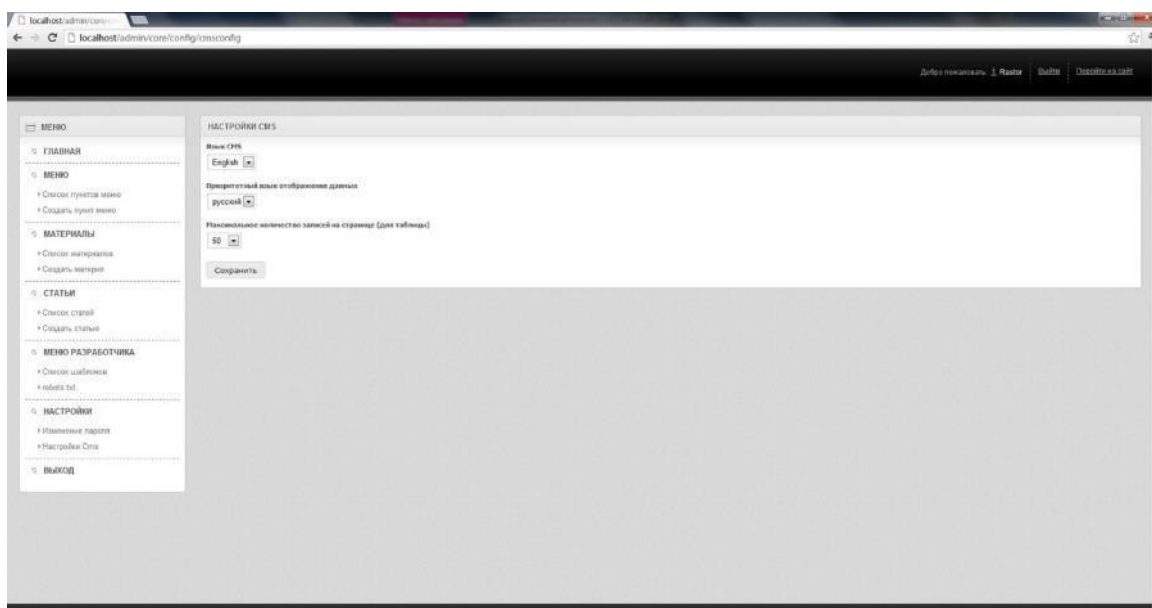


Рис. 4.27. Загальні налаштування зовнішнього вигляду системи

Щоб вийти з системи, слід скористатися кнопкою «Вихід» у лівій частині головного інтерфейсу. Після її натискання користувач автоматично повернеться до входу в систему (рис. 4.28).

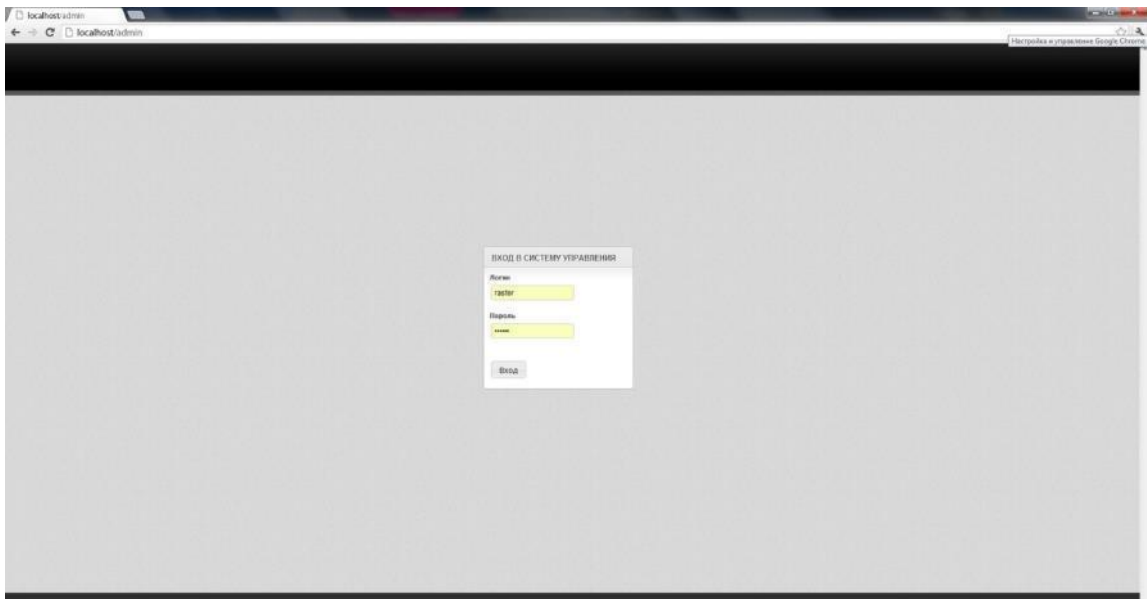


Рис. 4.28. Результат виходу з системи

#### **4.5.5. Оптимізація веб-сайту через систему управління контентом.**

Однією з особливостей систем управління контентом є можливість оптимізації веб-сайту за допомогою системи.

Пошукова оптимізація – це комплекс заходів, спрямованих на підвищення позицій сайту в результатах видачі пошукових систем (Google, Yahoo та інших) за певними ключовими словами або фразами. Це означає, що якщо є сайт довідкового центру, але він посідає 50-те місце в результатах пошуку за запитом «довідковий центр», тобто не є першою сторінкою в загальному списку результатів, навряд чи хтось із потенційних читачів відвідає цей сайт. Тому існують засоби, які називаються SEO-оптимізацією.

Сьогодні просування сайту є однією з визначальних функцій системи управління контентом. Саме тому розроблена CMS має спеціальні поля, щоб у

майбутньому, за необхідності, оптимізатор міг вивести сайт на перші позиції в пошукових системах.

Оскільки всі сторінки веб-проекту можуть бути використані для оптимізації сайту, поля для заповнення під час просування включені в кожен текстовий матеріал адміністративного розділу (рис. 4.29).

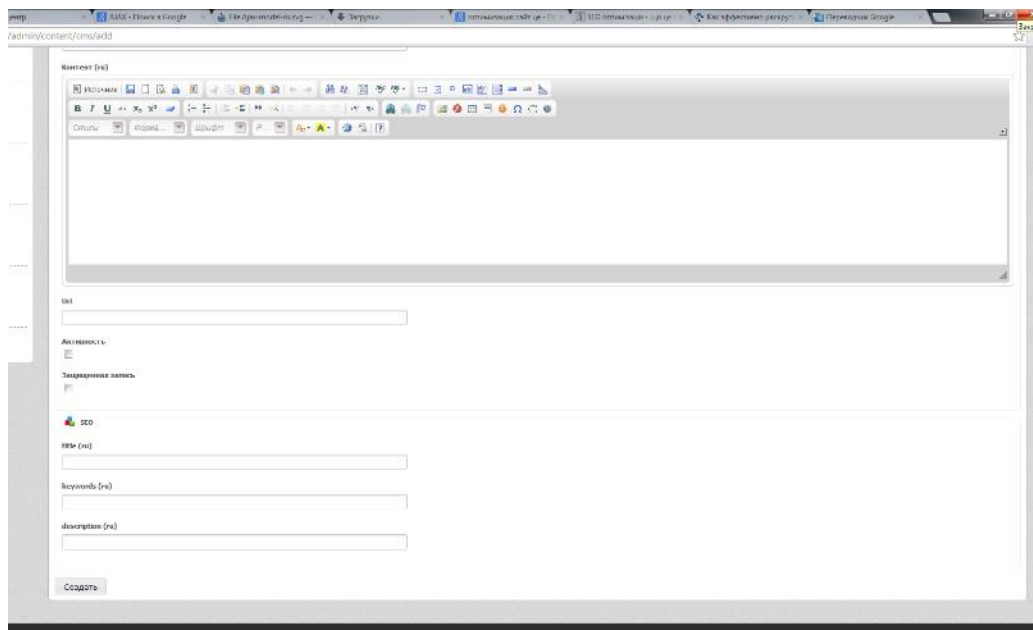


Рис. 4.29. Поля для оптимізації веб-сайту



#### **Висновок до розділу 4**

На основі критеріїв порівняння систем управління контентом було зроблено висновок про доцільність розробки власної системи управління на базі одного з відомих шаблонізаторів. Цей вибір обґрунтовано тим, що:

- така система значно скорочує часові та матеріальні витрати;
- система проста у використанні, не потребує спеціальної підготовки для користувачів без навичок гіпертекстової розмітки та програмування;
- у цій системі немає зайвих функцій, що полегшує роботу і робить інтерфейс зрозумілим для користувача.

## ВИСНОВКИ

За останнє десятиліття веб-сайти стрімко еволюціонували від статичних до динамічних. Основна відмінність динамічного сайту полягає в тому, що він дозволяє власнику сайту змінювати структуру і контент без допомоги програміста. Повна розробка веб-сайту включає використання системи управління контентом, щоб зробити сайт динамічним. Власник динамічного сайту може легко змінювати текст, зображення, відео і навіть зовнішній вигляд сайту. Особливістю динамічного веб-сайту, розробленого за допомогою CMS, є можливість завантажувати файли та керувати діями користувачів без допомоги програміста.

В роботі були вирішені наступні завдання:

- зібрати найбільш важливу та актуальну інформацію про різні відомі системи управління контентом;
- дослідити та проаналізувати основні принципи роботи систем, їх структуру та функціональність компонентів;
- визначено основні критерії, якими слід керуватися при виборі CMS для створення власного веб-сайту;
- здійснено порівняння відомих платних та безкоштовних CMS-систем, визначено їх переваги та недоліки, а також зроблено детальний огляд кількох найпоширеніших платних та безкоштовних систем управління контентом;
- зроблено аналітичний огляд ринку CMS. На основі порівняння всіх існуючих CMS автор дійшов висновку, що на основі одного з популярних конструкторів шаблонів можна створити CMS, яка буде відрізнятися від інших;
- здійснено огляд програмних технологій та інструментів, обраних для реалізації проекту;
- визначено вимоги до програмного та апаратного забезпечення системи; надано короткий огляд новітніх технологій.

Таким чином, використання CMS дає розробнику багато переваг при створенні веб-сайту:

- абстрагування від дизайну;
- людина працює лише над наповненням сайту;
- вирішуються завдання автоматизації управління сайтом;
- дані зберігаються не у файлах, а в системах реляційних СУБД, що набагато простіше і зручніше;
- є можливість створення різних частин сайту з різними правами доступу.

Розроблена система успішно реалізувала всі перераховані вище цілі і завдання. Це дозволяє говорити про високу ефективність використання даної розробки.

Матеріал, що міститься в інструкціях (Додаток В), призначений для широкого кола читачів завдяки своїй простоті та доступності. Інструкції супроводжуються рисунками та прикладами, що також полегшує розуміння та засвоєння матеріалу.

## **СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. "CMS Design Using PHP and jQuery" by Khasru Asif, Apress, 2010
2. "WordPress Plugin Development: A Comprehensive Guide" by Vladimir Prelovac, Packt Publishing, 2009
3. "Flask Web Development: Developing Web Applications with Python" by Miguel Grinberg, O'Reilly Media, 2018
4. "Node.js, MongoDB, and AngularJS Web Development" by Brad Dayley, Addison-Wesley, 2014
5. JavaScript and jQuery: Interactive Front-End Web Development by John Duckett, Jack Moore, 2014
6. High Performance Browser Networking: What every web developer should know about networking and web performance, Ilya Grigorik, 2013
7. Web Application Architecture: Principles, Protocols and Practices, Rich Rosen, Leon Shklar, 2009

# ДОДАТКИ

## Додаток А

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <script type="text/javascript" src="/design/js/maxheight.js"></script>
    <script type="text/javascript" src="/design/js/jquery-1.4.2.min.js" ></script>
    <script type="text/javascript" src="/design/js/cufon-yui.js"></script>
    <script type="text/javascript" src="/design/js/cufon-replace.js"></script>
    <script type="text/javascript" src="/design/js/Myriad_Pro_400.font.js"></script>
    <script type="text/javascript" src="/design/js/script.js"></script>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta http-equiv="Content-Language" content="ru_RU" />
    <meta name="keywords" content="" />
    <meta name="description" content="" /><title>Про центр</title><link href="/design/css/reset.css"
    media="screen" rel="stylesheet" type="text/css" />
    <link href="/design/css/layout.css" media="screen" rel="stylesheet" type="text/css" />
    <link href="/design/css/style.css" media="screen" rel="stylesheet" type="text/css" /><script
    type="text/javascript" src="/design/js/maxheight.js"></script>
    <script type="text/javascript" src="/design/js/jquery-1.4.2.min.js"></script>
    <script type="text/javascript" src="/design/js/cufon-yui.js"></script>
    <script type="text/javascript" src="/design/js/cufon-replace.js"></script>
    <script type="text/javascript" src="/design/js/Myriad_Pro_400.font.js"></script>
    <script type="text/javascript" src="/design/js/script.js"></script><!--[if lt IE 7]>
    <link rel="stylesheet" href="/design/css/ie/ie6.css" type="text/css" media="screen">
    <script type="text/javascript" src="/design/js/ie_png.js"></script>
    <script type="text/javascript">
      ie_png.fix('.png, #slogan, header nav ul li a, .icon img, .link1, .link1 span');
    </script>
  <![endif]-->
  <!--[if lt IE 9]>
    <script type="text/javascript" src="/design/js/html5.js"></script>
  <![endif]-->
  </head>
  <body id="page1" onLoad="new ElementMaxHeight();">
    <div class="wrap">
      <!-- header -->
      <header>
        <div class="container">

          <h1><a href="/">Science and Technologies</a></h1>

          <nav>
            <ul class="navigation">
              <li>
                <a href="/index.html">Головна</a>
              </li>
              <li>
                <a href="/about.html">Про сайт</a>
              </li>
              <li>
                <a href="/articles">Статті</a>
              </li>
              <li>
                <a href="/news.html">Новини</a>
              </li>
              <li>
                <a href="/contacts.html">Контакти</a>
              </li>
            </ul>
          </nav>
          <div id="slogan"></div>
        </div>
      </header>
    </body>
  </html>
```

```

</header>
  <!-- content -->
  <section id="content">
    <div class="container">
      <div class="inside">
        <div class="wrapper indent">
          <p style="text-align: center; ">
            <p><strong>Що</strong><strong> таке інформаційні технології</strong> Люди, які не
пов'язані з IT, швидше за все, скажуть, що це щось складне, незрозуміле і розумне. Але чи так
це насправді?</p>
            <p>Для аборигена, що не вміє читати і писати, навіть найпростіше речення буде непосильним, але
коли ти володієш мовою &ndash; ти легко можеш донести навіть найскладніші думки простими й
зрозумілими словами. Так само й з мовами програмування. Адже, не потрібно бути генієм, щоб навчитися
читати і писати!</p>
            <p>Простими словами, інформаційні технології &ndash; це все, що пов'язано з обробкою,
зберіганням і передачею даних. І сьогодні вони оточують нас у всіх сферах життя: записна книжка в
вашому телефоні &ndash; це база даних, улюблений новинний сайт &ndash; це зверстаний програмний код,
бортовий комп'ютер автомобіля &ndash; спеціальна обчислювальна система. Хочемо ми того чи ні,
IT стали невід'ємною частиною сучасного життя і займають значне місце в суспільстві
майбутнього.</p>
            <p>За останні 10 років IT-галузь стрімко розвивається, і Україні в цьому плані є чим
пишатись.</p>
            <p><strong>Наука і Технології</strong> - онлайн-видання, присвячене цифровим технологіям. Мета
проекту - своєчасна публікація максимально об'єктивної інформації про все, що відбувається на ринку
IT, а також допомога користувачам цифрових пристроїв у виборі, придбанні та найбільш ефективному
використанні обладнання та програмного забезпечення.</p>
            <p>Завдяки широкому спектру розглянутих подій і високотехнологічних новинок, ресурс однаково
цікавий всім верствам "населення" інтернету: від любителів комп'ютерних ігор і модних гаджетів до
професіоналів - системних інженерів, програмістів.</p>
            <ul class="banners
col-2">
              <li><a href="#"></a></li>
              <li><a href="#"></a></li>
            </ul>
          </div>
        <div>
          <h2>Recent Articles</h2>
          <ul class="articles">
            <li class="maxheight">
              <div class="box maxheight">
                <span class="icon"></span>
                <div class="border-right maxheight">
                  <div class="border-bot maxheight">
                    <div class="border-left maxheight">
                      <div class="left-top-corner maxheight">
                        <div class="right-top-corner maxheight">
                          <div class="right-bot-corner maxheight">
                            <div class="left-bot-corner maxheight">
                              <div class="inner">
                                <h3>About Template</h3>
                                <p>Sed ut perspiciatis ungomnis
iste natus error sit volup tiatem <a href="#">accusantium loremque</a> lautium, totam rem
aperiam.</p>
                                <div class="wrapper">
                                  <a href="#"
class="link1"><span><span>Read More</span></span></a>
                                </div>
                              </div>
                            </div>
                          </div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </li>
            <li class="maxheight">

```

```
<div class="box maxheight">
  <span class="icon"></span>
  <div class="border-right maxheight">
    <div class="border-bot maxheight">
      <div class="border-left maxheight">
        <div class="left-top-corner maxheight">
          <div class="right-top-corner maxheight">
            <div class="right-bot-corner maxheight">
              <div class="left-bot-corner maxheight">
                <div class="inner">
                  <h3>Donations</h3>
                  <p>At vero eos et accusamus et
iusto odio dignissimos ducimus qui blanditiis praesentium vo- luptatum deleniti.</p>
                <div class="wrapper">
                  <a href="#"
class="link1"><span><span>Read More</span></span></a>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
</li>
<li class="maxheight last">
  <div class="box maxheight">
    <span class="icon"></span>
    <div class="border-right maxheight">
      <div class="border-bot maxheight">
        <div class="border-left maxheight">
          <div class="left-top-corner maxheight">
            <div class="right-top-corner maxheight">
              <div class="right-bot-corner maxheight">
                <div class="left-bot-corner maxheight">
                  <div class="inner">
                    <h3>Our Mission</h3>
                    <p>Temporibus autem quibusdam et
aut officiis <a href="#">debitis aut rerum</a> necessitatibus saepe eveniet ut et voluptates.</p>
                  <div class="wrapper">
                    <a href="#"
class="link1"><span><span>Read More</span></span></a>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</li>
</ul>
</div>
</div>
</section>
</div>
<footer>
<div class="container">
  <div class="left">
    <div class="right">
      <div class="inside">
```

```
                <a rel="nofollow" href="http://www.templatemonster.com/"
class="new_window">Website template</a> designed by TemplateMonster.com<br/><a
href="http://www.templates.com/product/illustrations/" class="new_window">Vector Illustrations</a>
provided by Templates.com
            </div>
        </div>
    </div>
</div>
</footer>
<script type="text/javascript">Cufon.now();</script>    </body>
</html>
```



## Додаток Б

### Вихідний код системи управління контентом

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta http-equiv="Content-Language" content="ru_RU" /><title></title><link href="/favicon.ico"
rel="favicon" />
    <link href="/design/cms/css/style.css" media="screen" rel="stylesheet" type="text/css" />
    <link href="/design/cms/jqueryui/jquery-ui-1.8.16.custom.css" media="screen" rel="stylesheet"
type="text/css" />
    <link href="/design/cms/jqueryui/jquery.ui.selectmenu.css" media="screen" rel="stylesheet"
type="text/css" />
    <link href="/design/cms/css/uniform.css" media="screen" rel="stylesheet" type="text/css" />
    <link href="/design/cms/css/jquery.jcrop.css" media="screen" rel="stylesheet" type="text/css"
/>
    <link href="/design/fancybox/jquery.fancybox-1.3.4.css" media="screen" rel="stylesheet"
type="text/css" /><script type="text/javascript" src="/design/cms/js/jquery-
1.6.2.min.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery-ui-1.8.16.custom.min.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery.ui.dialog.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery.ui.nestedSortable.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery.ui.selectmenu.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery.uniform.min.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery.iframe-post-form.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery.jcrop.min.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery.json.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery.rcwolf.table.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery.rcwolf.messenger.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery.rcwolf.pictures.js"></script>
<script type="text/javascript" src="/design/cms/js/jquery.rcwolf.tree.js"></script>
<script type="text/javascript" src="/design/fancybox/jquery.fancybox-1.3.4.pack.js"></script>
<script type="text/javascript" src="/ckeditor/ckeditor.js"></script>
<script type="text/javascript" src="/ckeditor/adapters/jquery.js"></script>
<script type="text/javascript">
  jQuery.fn.exists = function() {
    return $(this).length;
  }

  $(document).ready(function(){
    $("form input[type=text], input[type=password], form textarea, form select, form
input[type=file]").uniform();
    $(' .msg').click(function(){
      $(this).fadeOut(350, 0);
      $(this).slideUp(350);
    });

    $(' .msg.autoFadeDown').click(function() {
      $(this).fadeOut(350, 0);
      $(this).slideUp(350);
    }).delay(3000).fadeOut(350, 0).slideUp(350);

    $('button, input[type=button], input[type=submit], input[type=reset]').button();
    $(' .wyswsg').ckeditor({
      'filebrowserBrowseUrl': '/ckeditor/kcfinder/browse.php?type=files',
      'filebrowserImageBrowseUrl': '/ckeditor/kcfinder/browse.php?type=images',
      'filebrowserFlashBrowseUrl': '/ckeditor/kcfinder/browse.php?type=flash',
      'filebrowserUploadUrl': '/ckeditor/kcfinder/upload.php?type=files',
```

```
'filebrowserImageUploadUrl': '/ckeditor/kcfinder/upload.php?type=images',
'filebrowserFlashUploadUrl': '/ckeditor/kcfinder/upload.php?type=flash'
```

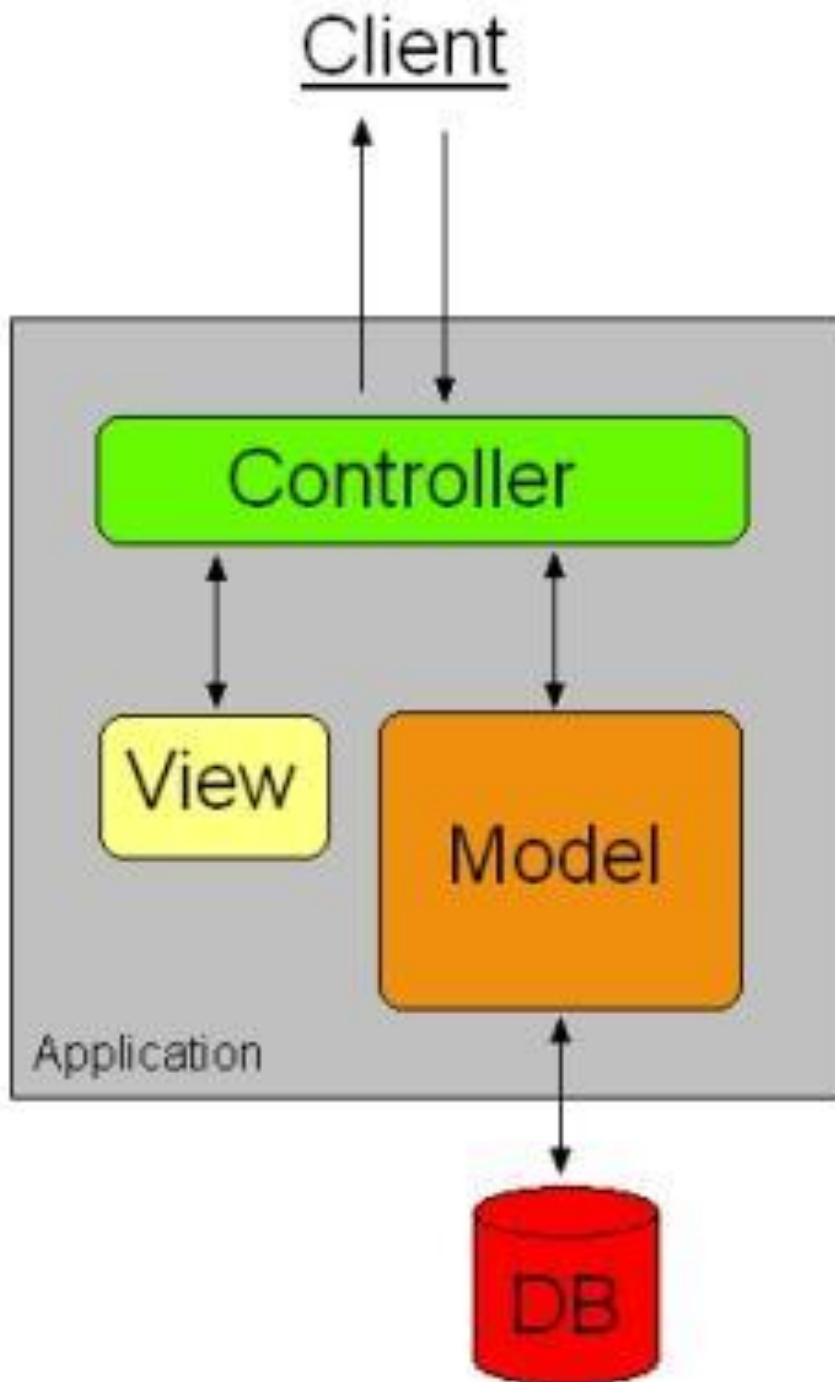
```
$('.fancy').fancybox();
});
</script>
</head>
<body>
  <div id="wrapper">
    <div id="header">
      <div class="main">
        <div class="logo">
          <div class="version"></div>
        </div>
      </div>
      <div class="bottom"></div>
      <div id="user_info">
        <div>
          <a href="/" target="_blank">Перейти на сайт</a>
        </div>
        <div class="exit">
          <a href="/admin/core/admin/logout">Выйти</a>
        </div>
        <div class="wellcome">
           <span>Rastor</span>
        </div>
      </div>
    </div>

    <div id="middle">
      <div id="container">
        <div id="mcontent">
          <div class="box">
            <h1>Главная</h1>
          </div>
        </div>
      </div>
    </div>
    <li class="active">
      <a href="/admin">Главная</a>
    </li>
    <li>
      <a href="#">Меню</a>
      <ul>
        <li>
          <a href="/admin/menu/cms/show">Список пунктов меню</a>
        </li>
        <li>
          <a href="/admin/menu/cms/add">Создать пункт меню</a>
        </li>
      </ul>
    </li>
    <li>
      <a href="#">Материалы</a>
      <ul>
        <li>
          <a href="/admin/content/cms/showlist">Список материалов</a>
        </li>
        <li>
          <a href="/admin/content/cms/add">Создать материл</a>
        </li>
      </ul>
    </li>
  </li>
```

```
<a href="#">Статьи</a>
<ul>
  <li>
    <a href="/admin/articles/cms/showlist">Список статей</a>
  </li>
  <li>
    <a href="/admin/articles/cms/add">Создать статью</a>
  </li>
</ul>
</li>
<li>
  <a href="#">Меню разработчика</a>
  <ul>
    <li>
      <a href="/admin/developer/view/showlist">Список шаблонов</a>
    </li>
    <li>
      <a href="/admin/developer/editor/edit/file/robots.txt">robots.txt</a>
    </li>
  </ul>
</li>
<li>
  <a href="#">Настройки</a>
  <ul>
    <li>
      <a href="/admin/core/config/changepassword">Изменение пароля</a>
    </li>
    <li>
      <a href="/admin/core/config/cmsconfig">Настройки Cms</a>
    </li>
  </ul>
</li>
<li>
  <a href="/admin/core/admin/logout">Выход</a>
</li>
```

Додаток В

MVC Diagram



## Додаток Г

### Таблиця

### Порівняння CMS по загальним критеріям

	<b>Безкоштовні Open source движки  (Drupal, Joomla, Wordpress)</b>	<b>Комерційні движки  (Bitrix, netcat, uni.cms)</b>	<b>Системи власної розробки</b>
<b>Безпека та захист від взлому</b>	<ul style="list-style-type: none"> <li>• Відкритий код дозволяє хакерам знаходити дірки в безпеці системи. Навіть постійні оновлення системи - не гарантія від випадкового або цілеспрямованого злому.</li> <li>• в разі серйозних доопрацювань системи під потреби замовника регулярні оновлення - під питанням.</li> </ul>	<ul style="list-style-type: none"> <li>• Високий рівень безпеки забезпечений закритим кодом і, як правило, особливою увагою розробників до підвищеної безпеки.</li> <li>• Швидке реагування в разі виявлення вразливостей.</li> </ul>	<ul style="list-style-type: none"> <li>• Високий рівень безпеки забезпечений закритим кодом і особливою увагою розробників до підвищеної безпеки.</li> <li>• Швидке реагування в разі виявлення вразливостей.</li> </ul>
<b>Складність впровадження та підтримки</b>	Велика кількість фахівців-самоучок забезпечує відносно низьку вартість впровадження. Але відсутність централізованої підготовки фахівців обумовлює низьку якість впровадження.	Як правило, компанія-розробник системи проводить навчання та сертифікацію програмістів, що працюють з движком. У результаті якість їхньої роботи в середньому вище, але і вартість значно більше.	<ul style="list-style-type: none"> <li>• Впровадження виконується і контролюється фахівцями, які брали участь у розробці системи.</li> <li>• Безкоштовна гарантія технічної працездатності сайту на весь період обслуговування.</li> <li>• Оперативне виправлення помилок, що виникли з вини розробника і виявлених у процесі роботи сайту.</li> </ul>

<p><b>Продуктивність</b></p>	<ul style="list-style-type: none"> <li>Є безліч порівняльних тестів продуктивності різних систем. Але ці тестування мають мало спільного з продуктивністю роботи системи в реальних умовах. Фактично продуктивність залежить більше від якості впровадження та обладнання, ніж від вибору конкретної системи.</li> <li>Загальна відміна всіх «коробкових» систем - зайва функціональність, що виражається в підвищеній ресурсоемності системи. Відповідно продуктивність нижче, ніж у спеціально підготовленої версії движка.</li> </ul>		<ul style="list-style-type: none"> <li>При збірці сайту включаються тільки той функціонал, який затребуваний по ТЗ.</li> <li>Всі зайві функції, наявність яких може погіршити продуктивність системи - відключаються.</li> <li>Глибокі знання ядра системи дозволяють, у разі необхідності, проводити тонкі налаштування скриптів і запитів до БД, що дозволяє мінімізувати ресурсомісткість системи і забезпечити максимальну продуктивність.</li> </ul>
<p><b>Стандартна та розширена функціональність</b></p>	<p>Обмежена стандартна функціональність, компенсується великою базою додаткових модулів (плагінів), розроблених сторонніми виробниками. Але відсутність централізованого контролю якості розробки плагінів, ставить під сумнів їх стабільність і можливість використання для бізнес-сайту.</p>	<p>Багата стандартна (готова) функціональність. Але будь-який додатковий модуль - це додаткові витрати - вартість модуля і його впровадження.</p>	<p>Багата стандартна (готова) функціональність. Необхідні модулі допрацьовуються при необхідності. Низька вартість, рівна вартості впровадження.</p>
<p><b>Вартість</b></p>	<p>Система - безкоштовно Впровадження - від \$ 300</p>	<p>Бізнес-версія системи, від \$ 1000, Впровадження - від \$ 1000</p>	<p>Система + впровадження бізнес-версії \$ 500</p>