

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри

_____ Скарга-Бандурова І.С.
« ____ » _____ 2017 р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА
ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Інформаційна комп'ютерна система управління вмістом веб-сайту

Освітньо-кваліфікаційний рівень “бакалавр”

Керівник проекту:

_____ (підпис)

Щербакова М.Є.

_____ (ініціали, прізвище)

Консультант з охорони праці:

_____ (підпис)

Я.О. Критська

_____ (ініціали, прізвище)

Студент:

_____ (підпис)

Куркчі А.П.

_____ (ініціали, прізвище)

Група:

КІ-13аД

Севєродонецьк 2017

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерної інженерії

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки 6.050102 Комп'ютерна інженерія

(шифр і назва)

Спеціальність _____

(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____

І.С. Скарга-Бандурова

« _____ » _____ 20__ р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Куркчі Антону Павловичу

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна комп'ютерна система управління вмістом веб-сайту

керівник проекту (роботи) Щербакова Марина Євгенівна доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " " 2017 р. № _____

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи Мова програмування PHP

Мова структурування та подання вмісту HTML 5

Система управління базами даних MySQL

Віртуальний сервер "Денвер" або його фізичний аналог

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____

1. Розглянути види сайтів, web-додатки, ринок CMS

2. Вимоги до системи управління вмістом веб-сайту

3. Розробити систему управління вмістом веб-сайту

4. Розробити базу даних до системи управління

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

Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Критська Яна Олександрівна, асистент		

7. Дата видачі завдання _____

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Збір та вивчення інформації для написання дипломного проекту(роботи)	До	
2	Написання вступу	До 10.06.2017	
3	Оформлення першого розділу	До 18.06.2017	
4	Вимоги до розроблюємої системи та написання другого розділу	До 30.05.2017	
5.	Розробка системи управління вмістом веб-сайту та опис її у третьому розділі	До 06.06.2017	
6	Написання висновків	До 10.06.2017	
7	Оформлення розділу з охорони праці	До 15.06.2017	
8	Виправлення зауважень	До 15.06.2017	
9	Захист дипломного проекту(роботи)	20.06.2017	
		(Згідно графіку)	

Студент _____

(підпис)

Куркчі А.П.

(прізвище та ініціали)

Науковий керівник _____

(підпис)

Щербакова М.Є.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту (роботи) бакалавра. 105с., 9 рис., 7 табл., 28 бібліографічних джерел посилань, 2 додатка.

Об'єкт розробки: інформаційна система управління вмістом веб-сайту(CMS)

Мета роботи: розглянути види сайтів і web-додатки; розглянути ринок CMS; розробити інформаційну систему управління вмістом веб-сайту.

В проєкті виконано:

1. Розглянені види сайтів, web-додатки та їх класифікація
2. Розглянений ринок популярних CMS
3. Написані вимоги до системи управління вмістом веб-сайту
4. Розроблена інформаційна система управління вмістом веб-сайту
5. Розроблена база даних до системи управління

Ключові слова: CMS (CONTENT MANAGEMENT SYSTEM), СТАТИЧНИЙ САЙТ, ДИНАМІЧНИЙ САЙТ, WEB-ДОДАТКИ, СТРУКТУРА САЙТУ, PHP, HTML, MYSQL, ДЕНВЕР.

ЗМІСТ

ВСТУП	7
1 ВИДИ САЙТІВ І WEB-ДОДАТКИ	8
1.1 Визначення CMS.....	10
1.2 Статичний сайт	11
1.3 Динамічний сайт.....	13
1.3.1 Поділ інформації та дизайну сайту.....	14
1.3.2 Спрощення модифікації і оновлення сторінок на сайті	14
1.3.3 Можливості змінювання контент сайту, реагуючи на дії відвідувач.....	15
1.4 Еволюція web-додатків	16
1.5 Класифікація web-додатків.....	18
1.6 Класифікація систем управління вмістом	21
1.7 Огляд ринку CMS	27
1.8 Висновок.....	31
2 ВИМОГИ ДО СИСТЕМИ УПРАВЛІННЯМ ВМІСТОМ WEB-САЙТУ	32
2.1 Структура сайту	33
2.2 Структура та елементи сторінок	34
2.3 Висновок.....	36
3 ОПИС РОЗРОБЛЕНОЇ СИСТЕМИ УПРАВЛІННЯМ СИСТЕМИ WEB-САЙТУ	37
3.1 Робота з системою адміністрування розробляємої (CMS)	37
3.2 Управління розділами(сторінками)	38
3.2.1 Редагування розділів(сторінок).....	41
3.3 Додавання новин.....	41
3.3.1 Редагування новин	43
3.3.2 Розділ довідка	43
3.4 Вибір платформи для реалізації проекту	44
3.5 Обґрунтування вибору мовних і програмних засобів.....	44
3.6 Розробка бази даних CMS.....	47
3.7 Принцип роботи системи.....	50
3.8 Функції конфігураційного файлу і їх використання	50
3.9 Висновок.....	55
4 ОХОРОНА ПРАЦІ	56
4.1 Правові та організаційні основи охорони праці	56
4.2 Вимоги до приміщень	58
4.3 Вимоги до організації місця праці	58
4.4 Виробнича санітарія	60
4.4.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу	60
4.5 Пожежна безпека	62

4.6	Електробезпека	65
4.7	Мікроклімат	66
4.8	Освітлення	67
4.9	Вентилювання	71
4.10	Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій	72
4.11	Висновки до розділу 4	78
ВИСНОВКИ		79
ПЕРЕЛІК ДЖЕРЕЛ І ПОСИЛАНЬ		80
Додаток А. Лістинг програми		83
Додаток Б. Презентація дипломного проекту		100

ВСТУП

Оскільки інтернет спочатку був засобом обміну даними між науковими центрами, ніхто особливо не дбав про прозорість і простоту його технологій. Лише в останні кілька років цей гігантський конгломерат найскладніших рішень почав "обростати" інструментарієм високого рівня, що дозволяє людям, що не володіє спеціальними знаннями в даній області, ефективно здійснювати будь-яку діяльність в Інтернеті.

Одним з основоположних ядер нового інтернету є системи управління контентом (CMS). Сайт, побудований на основі такої системи в ідеальному випадку перетворюється в бізнес-інструмент, яким можуть легко управляти безпосередньо контент-менеджери та піар-фахівці. З повсякденної роботи над ресурсом практично повністю виключається додаткове "передавальне ланка" (технічні фахівці), що призводить до реального зниження щомісячної вартості підтримки сайту і високою оперативності внесення змін до його структури.

З повсюдним впровадженням CMS уявлення про сайт як про аналог рекламного буклету або книги йде. Статичні ресурси, що вимагають для розвитку істотних обсягів ручної праці, стрімко поступаються місцем динамічним.

Користувачеві мережі Інтернет не видно, що відбувається всередині сайту, який він відвідує. Але Інтернет - розробникам і їх замовникам відомо, що більшість сучасних web-проектів будуються на основі CMS - систем управління контентом.

Сучасна добре спроектована система управління вмістом веб-сайту значно перевершує як за обсягом, так і за складністю розробки видиму частину ресурсу: вона дозволяє редагувати інформацію на сайті, виконувати різні завдання по адмініструванню, збирати й аналізувати різноманітну статичну інформацію про відвідувачів сайту та багато іншого.

1 ВИДИ САЙТІВ І WEB-ДОДАТКИ

Одне з найважливіших якостей хороших сайтів - не дизайн, хоча і це важливо, не наявність складних програмних блоків і навіть не швидкість завантаження по модему. Найважливіше властивість сайту з точки зору бізнесу - актуальність: наскільки правдива інформація, на скільки вона актуальна на поточний момент, наскільки точно вона відповідає цілям і задачам бізнесу.

Тому створення сайту - це лише початок його життєвого циклу. Хороший сайт - це не кам'яний нерухомий монумент, це живий організм, який швидко реагує на зміни зовнішнього середовища. Будь-яка інформація на сайті повинна змінюватися на стільки швидко, на скільки цього вимагають реалії бізнесу.

Кращим виходом, з точки зору оперативності, для компаній власників сайтів буде наявність штатного Web - майстра. Тільки він зможе зробити все швидко і якісно. Однак цей варіант далеко не найкращий з точки зору економії, так як вміст Web - майстри тільки для підтримки сайту не рентабельно, якщо це звичайно не дуже великий портал, або Інтернет - магазин.

Другий варіант - при першій необхідності звертатися до сторонніх Web - майстрам. Теж не кращий варіант. По-перше вимагає складання докладного технічного завдання на вносяться зміни, а значить вимагає додаткового часу. По-друге за послуги сторонніх розробників потрібно платити. Плата, як правило, береться за кожне звернення.

Третій варіант - договір на підтримку сайту. Цей варіант має свої плюси і мінуси, але зміни на сайті можуть бути зайвими протягом довгого періоду, в той час як гроші потрібно буде платити постійно.

Який же вихід? Організація системи управління вмістом сайту, що дозволяє працювати з нею людям, які не знайомі ні з мовою розмітки

гіпертексту, ні з мовою програмування. В цьому випадку можна самостійно оперативно вносити зміни на сайт, або покласти ці обов'язки на секретаря.

Розглянемо вигоди від використання систем управління вмістом веб-сайту, які вона дає іншій стороні: професіоналам сайтобудування - Web-майстер.

При обслуговуванні десятка сайтів вносити зміни в кожен з них відповідно до дизайну цього сайту є непосильним завданням. Не кажучи вже про те, що потрібно пам'ятати нюанси оформлення кожного з них.

Система управління (адміністрування) вмістом сайту або простіше CMS (Content Management System) дозволяє вирішити цю проблему. Досить налаштувати її один раз відповідно до дизайну, і всі зміни, в подальшому вносяться на сайт, будуть автоматично слідувати оформлення сайту. При цьому редагування HTML - коду потрібно тільки при наявності дійсно складних дизайнерських завдань.

Таким чином, розробка системи управління вмістом веб-сайту (CMS) принесе користь всім:

- Розробник підвищує якість і швидкість розробки своїх додатків, а отже і їх фінансову значимість. Крім того, надаючи замовнику, можливість самому вносити зміни на сайт, він тим самим, знімає з себе ці завдання, так як за часту підтримка сайту справа не дуже прибуткове, але досить клопітка;
- Замовник отримує зручний інструмент оперативного редагування свого сайту і незалежність від розробника [1].

1.1 Визначення CMS

Система управління контентом (Content Management System - CMS) - це автоматизований редакторський комплекс, що дозволяє управляти змістом і структурою Інтернет-ресурсу в режимі online. На практиці це означає, що будь-який, навіть незнайомий з інформаційними технологіями користувач, може абсолютно самостійно управляти змістом свого сайту, не вдаючись при цьому до послуг технічного персоналу.

Системи управління контентом можуть бути цікаві з тієї точки зору, що:

- Не вимагають професійної технічної підготовки при використанні;
- Можуть бути успішно інтегровані в внутрікорпоративну інформаційну систему і служити для організації документообігу;
- Економічно доцільні при організації внутрішньокорпоративних систем і інформаційних порталів.

Сайт (від англ., Site, ділянка) - це саме ділянку сервера, тобто розділ, повністю присвячений якійсь одній темі. Зрозуміло, практично всі сайти включають в себе безліч підрозділів, кожен з яких може дробитися на ще більш дрібні складові. Але в будь-якому випадку всі частини сайту об'єднує якась загальна ідея, смислова спрямованість, загальний стиль виконання. З точки зору всі сайти можна умовно розділити на два основних види: статичні і динамічні.

У 90-ті роки найбільш поширеною системою управління контенту була наступна: як програмного забезпечення використовувався web-сервер, що надається в складі послуги хостингу, редактор HTML-документів (від Notepad до Macromedia DreamWeaver) і FTP-клієнт (наприклад, CuteFTP, gFTP, FAR або TotalCommander). Сторінки створюються за допомогою редактора і розміщуються на сервері за допомогою FTP-клієнта. Структура сайту забезпечується посиланнями з одних сторінок на інші і, можливо, присутністю

меню на деяких сторінках. Зараз таку систему називають статичною. Назва пов'язана з тим, що вміст (контент) сторінок знаходиться в статичних (постійно розташованих на диску) файлах, на противагу динамічно створюваним сторінкам (генеруються на льоту за допомогою програми, що знаходиться на сервері) [2].

1.2 Статичний сайт

Статичною є гіпертекстова сторінка, яка цілком зберігається на сервері і показується відвідувачеві в своєму незмінному вигляді (слід врахувати, що статична сторінка може містити деякі змінювані елементи, наприклад банери, однак вона все одно залишається статичною). Статичним прийнято називати сайт, більшість або всі сторінки якого є статичними. Таких сайтів досить багато, з них, можна сказати, починався інтернет.

Переваги статичних сайтів:

- Легко змінити зовнішній вигляд будь-якої конкретної сторінки, які не вплинувши на вигляд інших сторінок;
- Нескладно додати на сайт нову сторінку, скопіювавши і виправивши файл з існуючої сторінкою;
- Людині, що займається підтримкою сайту, досить знати тільки мову HTML;
- Сайт буде працювати на будь-якому сервері хостингу, навіть з самими обмеженими можливостями;
- Сайт можна переглядати локально, не встановлюючи додаткове програмне забезпечення;
- Мале число використовуваних програмних компонентів утруднює злом такої системи.

Однак, при всіх своїх перевагах такі системи мають значні недоліки.

Недоліки статичних сайтів:

- Складно внести зміни в структуру і зовнішній вигляд сайту, так як для цього необхідно змінити вміст всіх сторінок (оскільки меню сайту, логотип і засоби навігації дублюються на кожній сторінці) ;
- Система не гарантує єдиний стиль сторінок сайту - будь-яка помилка при обробці файлів, що містять сторінки призводить до того, що різні сторінки сайту виглядають по-різному;
- Неможливо використовувати «динамічні» компоненти, такі як голосування, форум, і т.д;
- Відсутня поділ прав доступу до сайту, так як людина, що має FTP-доступ, може змінити будь-яку сторінку.

За роки використання таких систем були розроблені рішення, частково компенсують зазначені недоліки. Перший недолік може бути частково усунутий шляхом винесення меню і загальних елементів навігації в окремі файли, що може бути зроблено за допомогою технології серверних включень SSI - Server Side Includes або шляхом використання фреймів. Альтернативно, можна використовувати спеціальні засоби, що дозволяють замінювати шматки HTML-тексту в багатьох файлах одночасно. Третя вада може бути виключений з допомогою додавання необхідних програм, написаних на мові PHP. Четвертий недолік може бути частково компенсований шляхом розміщення сторінок в різні папки і призначенням різних прав доступу до цих папок.

Системи на базі HTML з використанням додаткових рішень до сих пір широко використовуються. Однак оскільки в ці системи додаються в тому чи іншому вигляді елементи програмування, виникає питання: «Чи варто засновувати систему на окремих HTML-сторінках або краще взяти за основу певний загальний програмний код і єдину структуру даних?». Тут ми перейдемо до сучасних «динамічним» системам управління сайтом.

1.3 Динамічний сайт

Динамічною є гіпертекстова станиця, яка формується сервером з декількох частин або отримується шляхом внесення або зміни даних в сторінку. Якщо статична сторінка не зазнає практично жодних змін і потрапляє на екран до відвідувача в своєму первісному вигляді, то динамічної сторінки, такий, як її бачить людина, на сервері не існує. Вона збирається кількома різними способами з даних, що зберігаються на сервері, і тільки після цього показується відвідувачеві.

Існує кілька способів генерації динамічних сторінок:

- Першим варіантом може бути об'єднання кількох (двох і більше) окремих частин в одну сторінку - це найпростіший спосіб генерації;
- Другим варіантом є заповнення шаблонної сторінки будь-якою інформацією, що зберігається окремо або надається у результаті роботи алгоритму (наприклад, в результаті обчислень) ;
- Третім, і, мабуть, найпоширенішим варіантом є поєднання перших двох у всіляких варіаціях, тобто сторінка збирається з декількох шматочків, в які при цьому вносяться різні зміни.

Відповідно динамічним можна називати сайт, більшість або всі сторінки якого, створюються динамічно.

Переваги динамічних сайтів:

- Поділ інформації та дизайну;
- Спрощення модифікації і оновлення сторінок;
- Можливість змінювати контент, реагуючи на дії відвідувача.

1.3.1 Поділ інформації та дизайну сайту

Використання динамічних сторінок дозволяє зберігати якийсь шаблон дизайну, в який, в залежності від сторінки, на яку зайшов відвідувач, поміщається необхідне наповнення. Такий варіант дуже зручний, адже для всього сайту створюється один або кілька шаблонів, і всі зміни дизайну, які потрібно зробити на сайті, виробляються тільки з ними.

Даний підхід також дозволяє розмежувати повноваження людей, що займаються наповненням і модифікацією сайту. Тобто одні співробітники можуть займатися дизайном, інші наповненням сторінок. В ідеалі вони навіть не будуть перетинатися, тільки відвідувач сайту буде бачити результат їхньої спільної роботи на екрані свого монітора.

З точки зору організації роботи та поділу праці, варіант створення сайту на підставі шаблонів практично ідеальний. Людина, відповідальний за дизайн сайту не зможе втрутитися в процес наповнення сторінок контентом, і за всі недоліки в дизайні буде нести відповідальність тільки він. І навпаки, той, хто займається інформаційним наповненням сайту, не зможе порушити дизайн сайту. Кожен займається своєю справою, не заважаючи іншим. Це прискорює роботу і знижує витрати.

1.3.2 Спрощення модифікації і оновлення сторінок на сайті

Ця перевага безпосередньо впливає з першого. Поділ інформації та дизайну на сайті дозволяє прискорити процес оновлення і наповнення сайту, тому що не вимагає від людей, що виконують ці операції, знань в областях

html-розмітки, графіки і т.д. Тобто можна навіть підготувати дизайн сайту (замовити у фірми, яка надає такі послуги), а потім проводити наповнення сайту власними силами, не витрачаючи кошти на постійне залучення сторонніх або включення в свій штат фахівців з дизайну, що треба було б при статичній організації сайту. Та й тимчасові витрати на оновлення значно скорочуються.

1.3.3 Можливості змінювання контент сайту, реагуючи на дії відвідувач

Третє досягнення стало проривом на шляху розвитку глобальної мережі. Всі переваги, описані в перших двох пунктах, полегшували роботу творців сайтів і знижували їх витрати, але для відвідувачів це було не дуже важливо. А ось третя перевага відбилася безпосередньо на них. Тільки завдяки йому ви можете бачити величезну кількість електронних магазинів, віртуальних клубів, форумів, інтернет - ігор і т.п. Тільки можливість змінювати наповнення сайту під конкретного користувача дозволило втілитися в життя цих проектів.

Недоліки динамічних сайтів:

Першим недоліком є необхідність використання додаткових програмних засобів для побудови динамічного сайту. Залежно від складності сайту, трудомісткість і вартість розробки таких програм може дуже сильно варіюватися.

Другим недоліком є підвищення вимог до апаратних потужностей серверних систем. Цей недолік безпосередньо впливає з попереднього, тому що тепер серверу потрібна ще виконати якусь програму для модифікації сторінки сайту, а тільки потім видати її відвідувачеві. Особливо помітною ця

проблема стає на сайтах з великою відвідуваністю. Часто в таких випадках доводиться проводити додаткові оптимізації коду для нормальної роботи сайту.

Отже, вартість послуг, за змістом такого сайту, набагато вище, ніж статичного. Хоча зараз, навіть багато безкоштовні хостинги підтримують можливість створення динамічних сайтів, не кажучи вже про платні, де все необхідне входить в стандартний набір послуг.

Незважаючи на ці недоліки, динамічних сайтів в мережі стає все більше і більше, мабуть переваги перекривають всі недоліки [3].

1.4 Еволюція web-додатків

Сервіс WWW почав функціонування в інтернеті в 1991 році, перший web-сервер дозволяв зберігати набір статичних web-сторінок мови розмітки гіпертексту (HTML). З практичної точки зору, можна було створити web-сайт, що представляє собою своєрідну візитівку, що не дивлячись на свою простоту користувалося великою популярністю, завдяки можливості отримати інформацію будь-якого роду в інтуїтивно зрозумілій і яскравій формі.

Як правило, перші web-сайти підтримувалися однією людиною. Власник сайту, створював статичну web-сторінку, за допомогою простого текстового редактора, або за допомогою спеціалізованої програми, після чого розміщував її на web-сервері. Перші web-сайти склалися з 1-3 сторінок і зміст міняли досить рідко, тому хоч і проста, але досить рутинна робота, яка не була проблемою.

Розробка web-додатки, як такого, стала можливою з впровадженням CGI, що дозволяло оперувати деякої логікою. Хоча насправді далі деяких дій над формами і відстеження сеансу користувача, тут справа не пішла і про більш

менш повноцінної платформи для розробки прикладних програм, можна говорити з появою деяких спеціалізованих Server-side мов сценаріїв призначених для інтеграції з HTML дозволяють динамічно генерувати сторінки.

Тут слід звернути увагу, що web спочатку не планувався як арени для розробки додатків, перші web-додатки, такі як: сайт-візитка, сайт-вітрина, Інтернет-магазин; дуже легкі за своєю структурою і реалізації, основна частина роботи по створенню подібних додатків, це більшою мірою дизайн, і перші web-розробники були більшою мірою дизайнерами. Мови призначені для web-розробки спочатку створювалися саме для людей, які не знайомі з програмуванням і основним завданням було забезпечити простоту використання.

Однак інтернет набирав популярність, стали доцільні нові «сервіси», в яких раніше не було сенсу, і від web- додатків стали вимагати більшого, подібні «полегшені» мови, за своєю суттю є шаблонизатор, не могли задовольняти виникаючим вимогам, і більшість з них було практично повністю перероблено. Серверні мови швидко наздогнали за своєю функціональністю своїх «повноцінних побратимів», однак тепер, при розробці серйозного додатки, дизайнеру доводилося працювати разом з програмістом, причому роль останнього поступово все більше збільшувалася.

І тут виникла проблема, пов'язана з повною відсутністю будь-яких методів до розробки web-додатків, якщо раніше на увазі їх простоти цим нехтували, то тепер, варварські способи до розробки унеможливлювали використання всіх функціональних можливостей мови розробки. Колективна робота з існуючими «методами» була практично неможливою, а взаємодія дизайнера і програміста зверталось в головний біль, як для першого, так і для другого.

Проблему вирішили за допомогою застосування патерну MVC, який з середини 80-х років минулого століття використовувався для розробки додатків з графічним інтерфейсом користувача.

Ідея паттерна MVC, лежить в поділі додатки, на кілька рівнів, стосовно web-додатків, паттерн знайшов реалізацію у вигляді Систем Управління Вмістом (Content Management System), що дозволило спростити проектування і розробку системи, дало можливість роботи над додатком кількох людей, полегшило процес налагодження та супроводу, скоротило терміни реалізації програми. Внаслідок чого існуючі програми (рівня сайту-візитки, сайту-каталогу, інтернет - магазину) були вдосконалені, і з'явилася можливість створювати нові, значно більш складні додатки.

1.5 Класифікація web-додатків

На рисунку 1.1, можна розглянути класифікацію web-додатків, що застосовуються комерційними організаціями в процесі своєї діяльності.

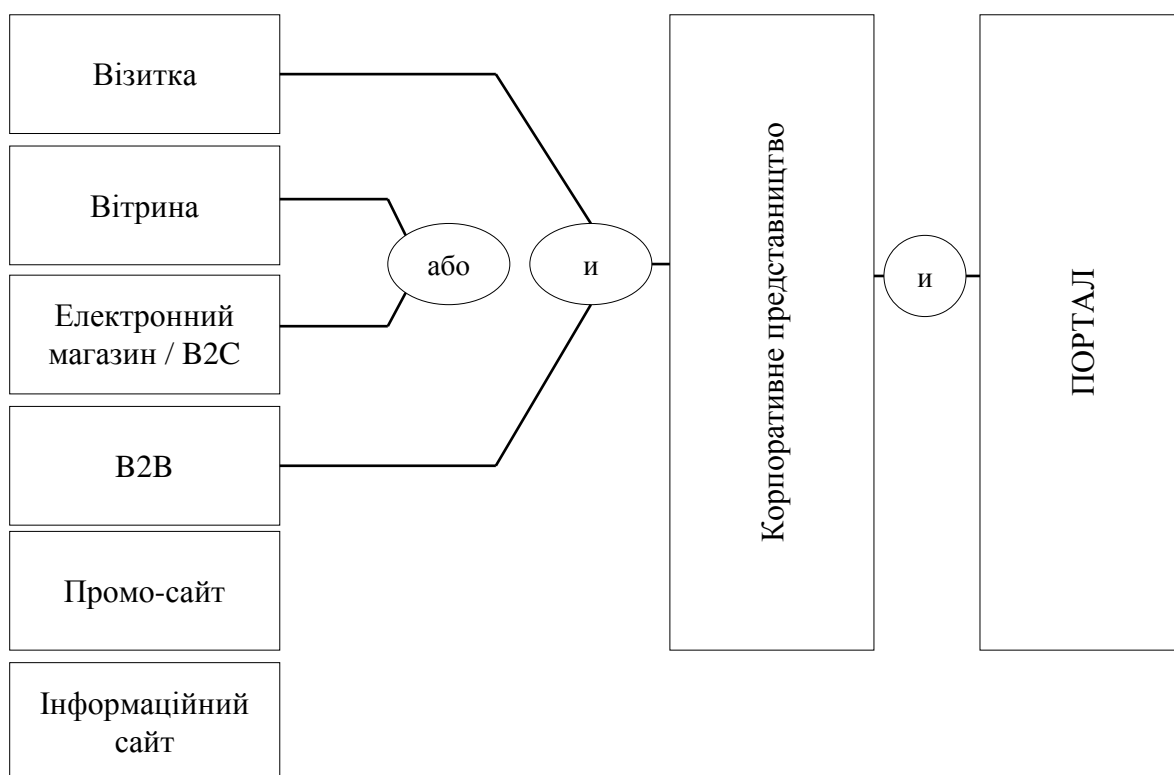


Рисунок 1.1. Класифікація web-додатків

Візитка - як правило, це загальна інформація про фірму, прайс-лист, реквізити, план проїзду і т.п. Тобто набір сторінок; можливо є форма для зворотного зв'язку.

Вітрина - сайти цього типу крім стандартних складових сайту-візитки містять ще й каталог товарів або послуг. На сторінках сайту розміщено докладний опис цих продуктів, яке може містити різні фотографії, схеми, презентації, детальну документацію і т.д. Часто в каталозі публікується вартість продукції і послуг. Додаток не містить системи ведення, замовлень, і покупки здійснюються або за допомогою електронної пошти, або за допомогою інших засобів зв'язку, в будь-якому випадку замовлення обслуговується менеджером.

Електронний магазин / B2C - електронні магазини багато в чому схожі на каталоги (сайти-вітрини). Ключове їх відмінність полягає в тому, що вони мають або власну систему формування, обробки, проведення і інших операцій

по замовленнях, або інтегруються з існуючою автоматизованою системою підприємства. Однак ця категорія умовна, сюди не тільки відносяться електронні магазини, в класичному уявленні, а будь-який додаток реалізує так звану схему B2C (Business To Consumer), тобто будь-який додаток, яке використовується для збуту товару кінцевому споживачеві. Крім стандартних магазинів, це різні інтернет-аукціони, електронні біржі праці і подібне.

B2B - і відповідно, додатки, що потрапляють під схему B2B (Business To Business), це додатки для роботи з дилерами, так звані електронні торгові майданчики. Сюди можна віднести платіжно-розрахункові системи, біржі, партнерські програми, і подібне. Це досить великовагові додатки, які, як правило, спочатку проектуються в якості становить модуля існуючої системи управління.

Промо-сайт - Розвиток client-side мов, призначених для інтеграції в HTML, такі як Java Script, ActiveX, і застосування існуючих мов, таких як Action Script, і Java аплетів, дало існування нової категорії web-додатків. Так званих промо-сайтів. Варіацій подібних рішень дуже багато, як правило несуть чисто іміджеву функцію, являють собою своєрідні листівки, дуже барвисті і динамічні.

Інформаційний сайт - Одним з поширених методів просування, організації в мережі, це формування так званого «інтернет комуніті», тобто спільноти постійних відвідувачів, сайту. Як правило, такі сайти містять велику кількість інформації, в тій області, в якій працює підприємство, ця інформація часто оновлюється і подібні сайти мають досить велику кількість коштів для спілкування між відвідувачами сайту.

Корпоративне представництво - як видно з наведеної вище схеми, корпоративні сайти виконують одночасно як функції сайту-візитки, так і електронного магазину або каталогу. Додатково корпоративні представництва містять B2B-системи, що дозволяють взаємодіяти з партнерами. Це відповідно найбільш просунуте додаток з описаних, як правило, для цієї програми

складається з кількох систем управління, які відповідають за функціонування, сервісів успадкованих від своїх попередників, і тут вже є власна система управління організуючу документообіг, управління, та інші функції з організації управління підприємством. Застосовується для великих підприємств, з великою дилерською і роздрібною мережею, яка має складної організацією і великою кількістю віддалених один від одного вузлів.

Портал - це найбільш складні web-додатки, які об'єднують в собі все перераховане, як правило, такий додаток розташовується, більш ніж на одному інтернет - ресурсі, одночасно функціонує на різних системах, функціонує в різних мережах, інтегрується з декількома системами управління.

1.6 Класифікація систем управління вмістом

До сих пір не розроблено достатньо чіткої класифікації систем управління контентом. Це відбувається тому, що ринок контент - систем досить молодий, і розробники такого роду програмних продуктів в значній мірі роз'єднані. Складність класифікації контент-систем полягає ще і в тому, що при цьому необхідно враховувати десятки параметрів, що відрізняють одну CMS від іншої. Тому будь-яку класифікацію можна назвати в достатній мірі умовною.

Можливо виділити 4 критерії класифікації системи управління вмістом:

- залежно від сфери застосування;
- за засобом поширення;
- за рівнем складності;
- за засобом роботи.

За способом розповсюдження системи управління бувають вільно-розповсюджуваними і платними. Разом з останніми в переважній більшості

випадків клієнти отримують супровід і підтримку.

Платні системи можна розділити на три цінові категорії.

1. Найбільш дешеві, які вироблені одним web-розробником або групою. Такі системи намагаються зробити якомога більше універсальними, щоб продати якомога більшої кількості потенційних покупців (як правило, інтернет-представництв дрібних компаній) ;

2. Системи середнього цінового діапазону, які розробляються під конкретного замовника;

3. Системи, створені гігантами розробки - Microsoft і ін.

Для класифікації за рівнями складності можна виділити наступні характеристики:

- наявність тих чи інших функцій і модулів, зрозумілість і доступність користувачеві;
- можливість функціонування системи на різних платформах, сумісність з базами даних, можливість підключення додаткових модулів;
- технологічність - використання технологій, що дозволяють підвищити надійність і швидкодію системи;
- потенціал розвитку системи.

На рисунку 1.2 приведено класифікація системи управління вмістом.



Рисунок 1.2. Класифікація систем управління вмістом

Отже, за рівнями складності CMS системи можна розділити на наступні групи.:

1. Проста CMS система. Система збирається з програмних модулів, для кожної системи вони свої. Модулі одного разу налаштовуються розробником, ніж жорстко закріплюється структура проекту. Для подальшої зміни структури сайту і його параметрів потрібна участь технічного персоналу. Від користувача системи потрібне знання основ HTML. Сумісність: система сумісна з певними платформами і типами СУБД. Спроба впровадження додаткових модулів, залежно від використовуваних технологій, може привести до повної переробки проекту. Технологічність: динамічне формування сторінок, обмежена пропускна здатність в діапазоні 3 000 - 5 000 відвідувачів на добу. Система інсталується розробником.

Розвиток системи досягається шляхом перекладу сайту на нову, більш пристосовану до поточних завдань, версію системи управління контентом;

2. Шаблонна CMS система. Єдиний модуль або набір модулів з жорстко закріпленою структурою сайту. Система містить набір сервісних функцій, що дозволяють виконати стандартні дії з сайтом: сформувати новинну стрічку, створити видалити новий розділ, вибрати шаблон для роботи з інформацією, завантажити файл і розставити по тексту картинку, встановити атрибути тексту і т.д. Система сумісна з певними платформами і типами СУБД. Спроба впровадження додаткових модулів так само, як і в попередньому випадку, обмежена. Формування сторінок відбувається динамічно або з використанням кешування даних. Залежно від методів формування сторінок в діапазоні на сайтах даного типу 5000 - 50000 відвідувачів на добу. Інсталюється розробником. Удосконалення досягається шляхом перекладу сайту на нову, більш складну, версію системи управління контентом;

3. Професійна CMS система. Інтуїтивно зрозумілий інтерфейс, візуалізація і розширені можливості редагування. Можливість зміни структури проекту - створення найрізноманітніших сайтів. Сумісність з різними програмно-апаратними платформами. Можливість підключення додаткових модулів як від розробника, так і прикладного програмного забезпечення. Кешування формованих динамічних сторінок. Пропускна спроможність не обмежена (залежить від апаратного забезпечення). Можливість безболісного підключення додаткових модулів без порушення структури та ідеології управління web-ресурсом;

4. Універсальна CMS система. Має просунуті засоби керування контентом, налаштування функціональності системи, можливість перевизначення атрибутів об'єктів сайту. Система надає засоби для розробки нових сервісів і можливостей. Технічні особливості. Розширені можливості API, наявність готових прикладних рішень, в т.ч. за участю програмних продуктів третіх фірм. Наявність сертифікованої системи забезпечення безпеки - розмежування прав доступу до системи на внутрішньокорпоративному рівні.

Кешування формованих динамічних сторінок.

Пропускна здатність - не обмежена .

За засобом роботи CMS системи можна розділити на три типи:

1. Генерація сторінок за запитом. Системи такого типу працюють на основі зв'язки «Модуль редагування → База даних → Модуль уявлення». Модуль уявлення генерує сторінку з вмістом при запиті на нього, на основі інформації з бази даних. Інформація в базі даних змінюється за допомогою модуля редагування. Сторінки заново створюються сервером при кожному запиті, що в свою чергу створює додаткове навантаження на системні ресурси. Навантаження може бути багато разів знижена при використанні коштів кешування, які є в сучасних web-серверах;

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

3. Змішаний тип. Як зрозуміло з назви, поєднує в собі переваги перших двох.

Може бути реалізований шляхом кешування - модуль уявлення генерує сторінку один раз, надалі вона в кілька разів швидше завантажується з кешу. Кеш може оновлюватися як автоматично, після закінчення деякого терміну часу або при внесенні змін до певні розділи сайту, так і вручну по команді адміністратора. Інший підхід - збереження певних інформаційних блоків на етапі редагування сайту і збірка сторінки з цих блоків при запиті відповідної сторінки користувачем .

За сферами застосування CMS системи можна розділити на наступні групи.

1. Портали. Використовуються для інформаційних ресурсів, основною метою ставлять максимальне спрощення публікації статей і новин. Можуть включати в себе перераховані нижче типи CMS як самостійні модулі. Найбільш відомі представники даного класу: AngelineCMS, Bes-cms, CoolPHP, CPG-Nuke,

вебZE, Xaraya, xNuke, XOOPS і ін;

2. Двигуни без SQL. Це відгалуження в розробці CMS розвинене відносно слабо, так як використання в якості сховища інформації файлів замість таблиць бази даних пов'язане з безліччю важко вирішуваних проблем (таких, як одночасний запис в один файл кількома копіями скрипта). Гідність цих CMS - в доступності для модифікації контенту і можливості розміщення на безкоштовних хостингах. Є кілька реалізацій цієї ідеї: Cute News, DeeLight CMS, Prog-ressive, SAPID;

3. Блог (походить від англ. Weblog, російський термін - «мережевий щоденник») - це сайт, на якому знаходяться особисті замітки автора. В основному нотатками є посилання на сайти, які здаються власникові ресурсу найбільш цікавими, і коментарі до них. Блог може містити не тільки посилання, але і просто електронний щоденник користувача. До цієї категорії можна віднести наступні CMS: b2evolution, bBlog, BLOG: CMS, MyPHPblog, Nucleus, pLog, pMachine Free, Serendipity, Textpattern, WordPress, XHP;

4. Форуми - це інструмент для спілкування на сайті. Повідомлення у форумі в чомусь схожі на поштові - кожне з них має автора, тему і зміст. Але для того, щоб відправити повідомлення в форум, не потрібна ніяка додаткова програма - потрібно просто заповнити відповідну форму на сайті. Принципова властивість форуму полягає в тому, що повідомлення в ньому об'єднані в треди (від англ. Thread - «нитка»). Коли ви відповідаєте в форумі на чиєсь повідомлення, ваш відповідь буде «прив'язаний» до вихідного повідомлення. До форумам, гідним уваги, можна зарахувати FUDforum, openBB, Phorum, phpBB, PunBB, W-Agora, XMB, Zorum, ExBB, IPB, vBulletin;

5. Магазили. До магазинів віднесемо будь-який сайт, з якого можна замовити будь-який товар. В даному випадку в визначення «товару» може входити абсолютно все, включаючи час доступу в Інтернет, хвилини стільникового зв'язку. Абсолютна більшість інтернет-магазинів є нелегальними. CMS, що дозволяють створити віртуальний магазин: MyMarket, osc2nuke, osCommerce, Zen Cart;

6. Групова робота (Groupware) - комплекс програмного забезпечення, що дозволяє організувати роботу підприємства, відносини з клієнтами і замовниками в Інтернеті. Зазвичай являє собою повністю або частково закриту частину сайту з можливістю відслідковувати терміни виконання поставлених завдань, розподіл ролей і тимчасових нормативів. Іноді можна виносити питання на обговорення і рішення вищого керівництва. Як правило, користуються такими CMS: dotProject, eGroupWare, MoreGroupware, phpCollab, PHProjekt

7. Навчання (e-Learning) - дистанційна форма навчання з використанням Інтернету. Онлайнова форма навчання вже не один рік є «маяком», на який орієнтуються освітні системи різних країн світу. Головним стратегічним напрямком є швидке оновлення знань і ефективне використання інформації. Таких систем небагато: ATutor, Claroline, LogiCampus, Moodle, Segue, eLearning 3000 [4].

1.7 Огляд ринку CMS

Успішне функціонування будь-якого сайту (будь-то розробка інтернет-магазину, або навіть створення сайту-візитки), а також реалізації найскладніших і індивідуальних завдань, як правило залежить від правильно підбраного "движка", CMS на якому буде функціонувати сайт.

Почнемо огляд з CMS системи Joomla - це безкоштовний "движок" для управління контентом. Головною особливістю цієї системи є її здатність до розширення свого функціоналу за рахунок додаткових модулів, які можна або завантажити готові, або написати самому.

Особливість Joomla це досить зрозумілий для користувача інтерфейс і здатність підлаштуватися під будь-які функціональні вимоги. Дуже добре

підходить під сайти-візитки і сайт для бізнесу.

Плюси CMS системи Joomla:

- Проста і зручна в адмініструванні
- Універсальна, підходить для вирішення безлічі завдань
- Існує дуже багато плагінів, модулів, розширень
- Чи не складне додавання своїх модулів
- Багатий вибір безкоштовних і платних шаблонів
- Часті оновлення самого движка і додаткових модулів для функціонування сайту
- Початківець користувач зможе самостійно і швидко освоїти адміністративну панель сайту

Минуси CMS системи Joomla:

- Падає швидкість при дуже великих розмірах сайту (наприклад якщо використовувати сайт в ролі інтернет-магазину або блогу)
- Часто піддається до атак хакерів, вимагає додаткової настройки безпеки
- При просуванні сайту потрібна ретельна і тонка настройка

Наступна CMS система це OpenCart. Адже це один з двигунів, який цілеспрямовано створювався під інтернет-магазини.

Плюси CMS системи OpenCart:

- Відкрита платформа.
- Безкоштовний.
- Відмінний і дуже мультимедійний функціонал даної ЦМС дозволить Вам з легкістю налаштувати під себе інтернет-магазин.
- Графічний дизайн цієї системи просто вражає своїм великими можливостями. Тут тобі і кількість користувачів на сайті і найпопулярніший товар і географічна карта світу, для розуміння з якої частини світу робляться покупки

- Широкий спектр модулів
- Реалізовано різні варіанти оплати товару (PayPal, MasterCard, Qiwi, Webmoney, Visa)
- Наявність відмінних SEO модулів,
- Дуже популярний серед програмістів.

Мінуси CMS системи OpenCart:

- Без додаткових модулів ("милиць") має низькі показники взаємодії з користувачем
 - Практично не кроссформатний. Підходить виключно для інтернет-магазинів або онлайн каталогів
 - Легко підлаштувати під себе, але вимагає великих витрат часу і ресурсів

MODx CMS система є непростюю для рядового користувача движком, але останнім часом вона займає одні з лідируючих позицій при розробці. Розробка сайту на MODx займає порівняно небагато часу, при цьому вартість розробки сайту на MODx - середня.

Плюси CMS системи MODx:

- Особливістю MODx є реалізація абсолютно будь-якого типу сайту будь-то інтернет-магазин або сайт-візитка. Все залежить від "рук програміста".
- Дуже добре підходить для SEO просування, тому що дублів сторінок Ви ніколи не побачите, тільки якщо Ви самі їх не створите. Ніякої зайвої "води", все чітко і по справі, ніяких непотрібних рядків коду Ви також не побачите.
- Повністю під Вашим контролем, якщо, звичайно, Ви вмієте поводитися з даної CMS. Досвідчені фахівці зможуть виправити будь-який піксель, який Вам не сподобатися.
- Чистий і "тиха" установка на хостинг без зайвих непотрібних модулів і всіляких доповнень. Встановлюється чисте ядро.

- Простота в шаблонах. Ніяких загадок, звичайний html код і є вашим шаблоном.

Мінус CMS системи MODx:

- Точна і тонка настройка вимагає досить хороших знань програмування .. Але для подальшого адміністрування дуже зручна і проста CMS.

WordPress - це найкраща CMS для ведення своїх особистих блогів і для новинних сайтів. На ній також можна створювати і ресурсовитратні сайти, такі як: інтернет-магазини, онлайн каталоги, бізнес сайти. З легкістю може впоратися з простеньким сайтом-візиткою.

Плюси CMS системи WordPress:

- Широкий вибір безкоштовних і платних шаблонів, що дозволити підібрати унікальний стиль для Вашого сайту.

- Зручний графічний інтерфейс управління сайтом. Після короткого навчання Ви будете себе почувати як риба в воді.

- Широкий спектр готових і перевірених часом модулів

- Відмінна продуктивність, природно, при невеликих обсягах наповнення сайту.

- SEO компонент забезпечить хороші позиції у видачі Google або Yandex.

Мінуси CMS системи WordPress:

- Повільна швидкість переходу по посиланнях, довге завантаження фотографій. На хороших хостингах цього можна і не побачити, тому що цей "движок" дуже ресурсомісткий і на поганому хостингу проблема стане помітнішою.

- За неофіційними даними - взлом системи не становить особливих труднощів. Хороші розробники знають як боротися з подібними нападами хакерів.

1.8 Висновок

У першому розділі було розглянуто що таке система управління вмістом web-сайту та їх класифікації.

Визначили одне з найважливіших якостей хороших сайтів. Це не дизайн, хоча і це важливо, не наявність складних програмних блоків і навіть не швидкість завантаження по модему. Найважливіше властивість сайту з точки зору бізнесу - актуальність: наскільки правдива інформація, на скільки вона актуальна на поточний момент, наскільки точно вона відповідає цілям і задачам бізнесу.

Система управління контентом (CMS) дозволяє управляти змістом і структурою Інтернет-ресурсу в режимі on-line. На практиці це означає, що будь-який, навіть незнайомий з інформаційними технологіями користувач, може абсолютно самостійно управляти вмістом свого сайту, не вдаючись при цьому до послуг технічного персоналу.

Також було виділено 4 критерії класифікації системи управління вмістом:

- за сферою застосування;
- за засобом поширення;
- за рівнем складності;
- за засобом роботи.

Були розглянені популярні CMS системи, та виділені їх плюси та мінуси

Отже остаточно можна сказати, що система управління змістом дозволяє полегшити адміністрування сайту і дозволяє все це робити в режимі онлайн [5].

2 ВИМОГИ ДО СИСТЕМИ УПРАВЛІННЯМ ВМІСТОМ ВЕБ-САЙТУ

Першим етапом створення будь-якого програмного продукту є постановка завдання. Маючи точно сформульовані цілі і завдання, можна приступати до проектування структури бази даних і всього програми в цілому. Розробку програми можна починати, тільки маючи готовий проект, інакше вся робота загрожує закінчитися невдачею і втратою дорогоцінного часу.

Основні вимоги, що пред'являються до розроблюваної CMS, полягають в наступному.

Розроблювана система повинна дозволяти:

- Редагувати існуючі сторінки і створювати нові.
- Редагувати інформаційне наповнення всіх сторінок сайту.
- Бути гнучкою, в плані додавання нових модулів.
- Мати новинний движок.
- Мати онлайн файл-менеджер.
- Бути інтегрована з базою даних.

Згідно з цими вимогами, всі сторінки сайту повинні створюватися засобами CMS. Виняток можуть становити дизайнозалежні блоки, наприклад копірайт внизу сторінки, зміни в яких можуть сильно вплинути на дизайн. Такі блоки, як правило, мало схильні до змін і створюються вигляді статичних HTML-блоків.

Для того щоб забезпечити не тільки зручну навігацію адміністраторам і редакторам ресурсу, але і простоту супроводу ресурсу, з самого початку необхідно спроектувати структуру розташування файлів і папок в директорії Document Root.

Структура і організація, що розробляється під потреби сайту, що включає наступні Web - додатки:

- Динамічно формуються сторінки, з системою адміністрування, яка дозволяє додавати, видаляти і редагувати їх.
- Блоки динамічно сформованих новин, забезпечені системою управління, що дозволяють створювати, видаляти, здійснювати правку новин редактору, якому для цього не обов'язково знати мову HTML.
- Блок «Фотоальбом», що дозволяє відвідувачам переглядати фотографії, що розміщуються адміністратором сайту за допомогою сторінки адміністрування даного Web - додатки.

2.1 Структура сайту

Розглянемо сайт, який має наступну структуру:

- Головна
- Контакти
- Відео
- Новини

З точки зору CMS, кожен унікальний інформаційний сторінку називатимемо сторінкою. А зону сайту з однією або декількома унікальними сторінками будемо називати розділом сторінок.

Таким чином, сайт буде являти собою групу розділів, що містять в собі одну або кілька сторінок.

Кожна сторінка містить в собі блок контенту, який може складатися з тексту, зображень та посилань, таблиць і т.п.

Розроблювана CMS повинна створювати і редагувати:

- Розділи сторінок, призначені для об'єднання сторінок в групи.

- Сторінки - інформаційні сторінки сайту.
- Контент - складові елементи сторінки.

2.2 Структура та елементи сторінок

Якщо спробувати розбити інформаційне наповнення сторінок на логічні елементи, то кожна сторінка може містити:

- Заголовки.
- Абзаци (блоки тексту).
- Зображення.
- Посилання.
- Таблиці.

З точки зору CMS, кожна сторінка містить блок контенту, яким можуть бути:

- Заголовки.
- Блоки тексту разом із зображеннями; зображення часто служать для ілюстрації тексту і в той же час зображення зазвичай мають підпис, тому блоки тексту доцільно об'єднати з зображеннями;

- Посилання, в тому числі на інші сторінки з CMS.
- Таблиці.

У термінології CMS всі ці елементи називаються блоками контенту, так як кожен з них являє собою окремий блок сторінки. Список типів контенту далеко не повний. Розвиваючи CMS, можна додавати нові типи контенту, виділяючи їх зі структури сторінок за будь-якими критеріями.

Контент - це структурний елемент сторінки. Він не володіє оформленням і дизайном. Оформлення присвоюється контенту у вигляді стилів. Кожен вид контенту в залежності від його типу може мати кілька стилів, один з яких повинен бути встановлений йому в системі управління .

Наприклад, «Тема» може мати стилі:

- Простір назв.
- Звичайний заголовок.
- Важливий блок сторінки.
- Тема примітки.

Абзацу можуть бути присвоєні такі стилі:

- Звичайний текст.
- Текст з відступом.
- Підпис до картинки.
- Примітка.

Стилі містять не тільки оформлення, але і логіку поведінки елементів, наприклад, у елемента «Посилання» можуть бути стилі:

- Відкрити посилання в новому вікні.
- Відкрити посилання в поточному вікні.
- Відкрити посилання у спливаючому вікні.

Останній стиль в основному використовується, якщо посилання вказує на окреме зображення.

Набір стилів для кожного елемента можна доповнювати, розвиваючи функціональність CMS [6].

2.3 Висновок

У другому розділі було визначено що, першим етапом створення будь-якого програмного продукту є постановка завдання. Маючи точно сформульовані цілі і завдання, можна приступати до проектування структури бази даних і всієї програми в цілому. Розробку програми можна починати, тільки маючи готовий проект, інакше вся робота загрожує закінчитися невдачею і втратою дорогоцінного часу.

Були виділені основні вимоги для розробки CMS, де всі сторінки сайту повинні створюватися засобами CMS. Виняток можуть становити дизайно-залежні блоки, наприклад копірайт внизу сторінки, зміни в яких можуть сильно вплинути на дизайн. Такі блоки, як правило, мало схильні до змін і створюються вигляді статичних HTML-блоків.

Також були виділені структура сайту та структура сторінок.

3 ОПИС РОЗРОБЛЕНОЇ СИСТЕМИ УПРАВЛІННЯМ СИСТЕМИ ВЕБ-САЙТУ

3.1 Робота з системою адміністрування розробляємої (CMS)

Для адміністрування ресурсу необхідно звернутися до ключового файлу бібліотеки адміністрування. Як правило, цей файл не виводять у вигляді посилань на сторінки ресурсу, в цілях безпеки (а якщо і виводять, то тільки посилання на сторінку авторизації користувачів). При зверненні в адресному рядку браузера до адреси: `http://www.імя_сайта.імя_домена/admin/` або `http://www.імя_сайта.імя_домена/admin/index.php` користувач потрапляє на сторінку авторизації(рис. 3.1) [9].

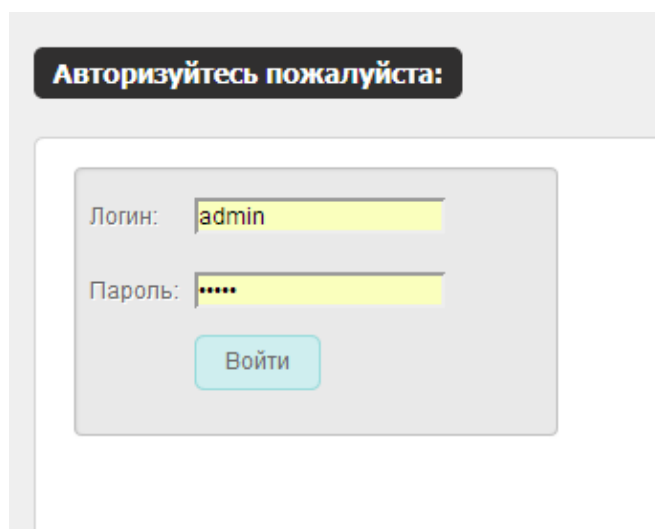


Рисунок. 3.1 Авторизація в CMS

Метадані сторінок і новин включають в себе унікальну посилання, контент (вміст) і інформацію для заголовка і метатегів сторінки сайту: заголовок сторінки (Тег <TITLE>), опис сторінки (метатег <DESCRIPTION>), ключові слова (<KEYWORDS>).

Робота над сторінкою як структурним елементом дозволяє здійснювати над нею наступні дії:

- Створення.
- Редагування (зміна метаданих сторінки).
- Видалення.
- Приховування / відображення в головному меню сайту.
- Редагування сторінки (робота з контентом).

CMS повинна забезпечувати наступні операції з контентом:

- Зручність в написання.
- Зручність в редагування.
- Видалення.
- Приховування / Відображення.
- Спрощення роботи завдяки візуальному редактору.

3.2 Управління розділами(сторінками)

Розділ меню - це гіперпосилання ведуча на певну сторінку сайту. Створення нового розділу здійснюється через систему CMS. На сторінці "Управління сторінками сайту", системи адміністрування (рис. 3.2), виводиться список існуючих розділів (сторінок), доступних для виведення в головне меню.

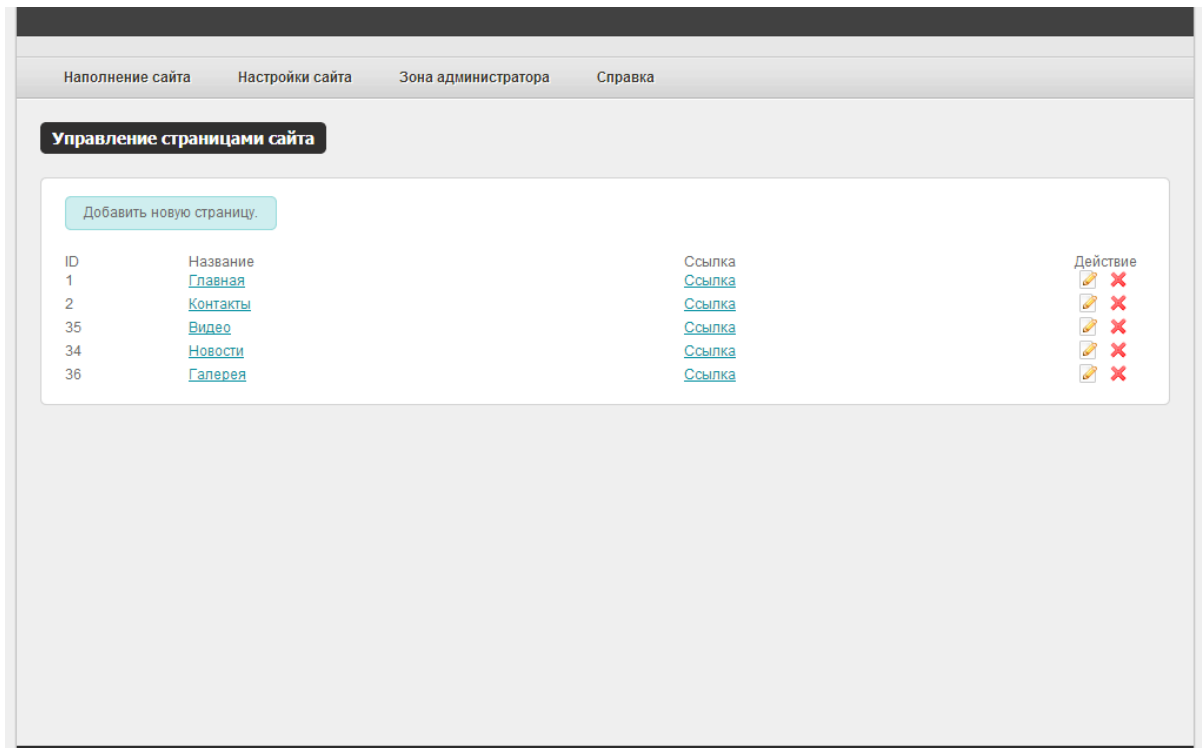


Рисунок 3.2. Управління розділами(сторінками) сайту.

Для створення нового розділу слід перейти за посиланням «Додати нову сторінку». Правда, така форма створення нової сторінки (рис. 5.3), містить шість полів:

- Ім'я сторінки.
- URL сторінки.
- Контент сторінок.
- Ключові слова для сторінки
- МЕТА- теги для сторінки

А так же перемикач типу «Список, що випадає»:

- Відображення в головному меню.

Який відповідає за відображення сторінки як розділу в головному меню сайту [10].

При натисканні кнопки зберегти, введені дані заносяться в базу даних; в разі успішного додавання сторінки виводиться повідомлення про успішно виконану операцію (Рисунок. 3.4).

3.2.1 Редагування розділів(сторінок)

Процес редагування розділів (сторінок) сайту, мало чим відрізняється від створення, за винятком вибору кнопки "Редагувати" (Піктограма) біля вже існуючої записи на сторінці «Управління сторінками сайту» (рис. 3.2).

3.3 Додавання новин

Управління, в тому числі додавання нових новин здійснюється на сторінці «Управління новинами сайту» (рис. 3.5).

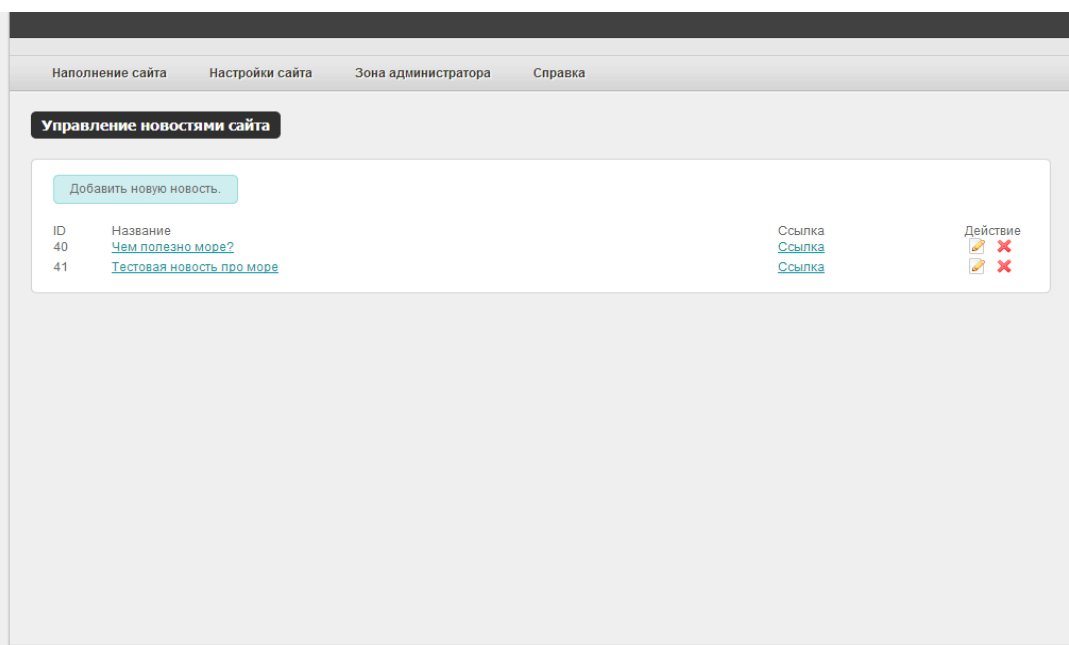


Рисунок 3.5. Сторінка управління новинами

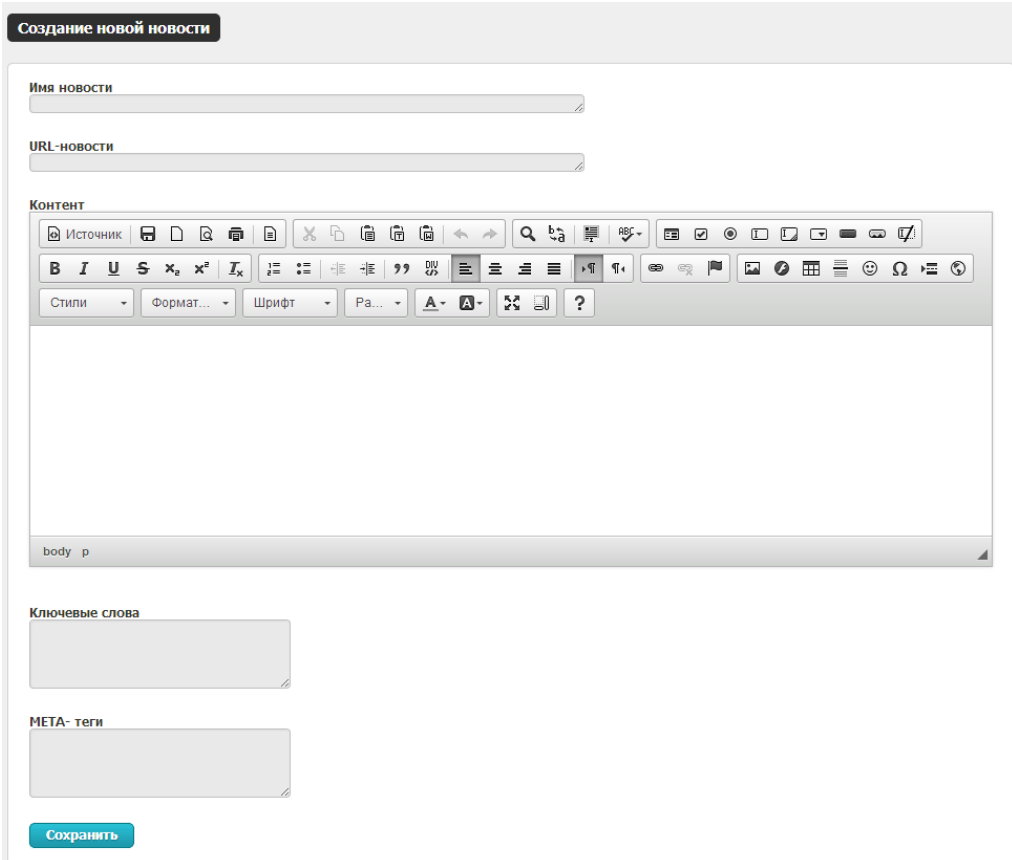
При зверненні до посилання "Додати нову новина" користувач переходить на сторінку додавання нової новини за допомогою онлайн - редактора (Рисунок 3.6). Даний редактор - це продукт, розроблений третьою,

називається CKeditor і безкоштовно розповсюджується у відповідності з ліцензією GNU Lesser General Public. CKeditor має інтуїтивно зрозумілий інтерфейс панелі управління, дуже схожою на панель редагування вмісту відомого редактора, як MS Word від Microsoft.

Сторінка створення нової новини (Рис. 3.6), містить п'ять полів:

- Ім'я новини.
- URL новини.
- Контент новини.
- Ключові слова для новини
- МЕТА- теги для новини

При натисканні кнопки зберегти, введені дані заносяться в базу даних; в разі успішного додавання новини виводиться повідомлення про успішно виконану операцію по аналогії з успішним додаванням сторінки (рис. 3.4).



The screenshot shows a web form titled "Создание новой новости" (Creating a new news item). It contains the following elements:

- A text input field labeled "Имя новости" (News name).
- A text input field labeled "URL-новости" (News URL).
- A section labeled "Контент" (Content) featuring a rich text editor with a toolbar containing various icons for text formatting, alignment, and insertion.
- A text input field labeled "Ключевые слова" (Keywords).
- A text input field labeled "МЕТА- теги" (Meta tags).
- A blue button labeled "Сохранить" (Save) at the bottom left.

Рисунок 3.6. Сторінка додавання нової новини в парі з візуальним-редактором CKeditor.

3.3.1 Редагування новин

Процес редагування новин, мало чим відрізняється від створення, за винятком вибору кнопки "Редагувати" (Піктограма) біля вже існуючої новини на сторінці «Управління новинами сайту» (рис. 5.5).

3.3.2 Розділ довідка

Була розроблена сторінка довідки (рис. 3.7), на якій детально викладені кроки управління CMS.

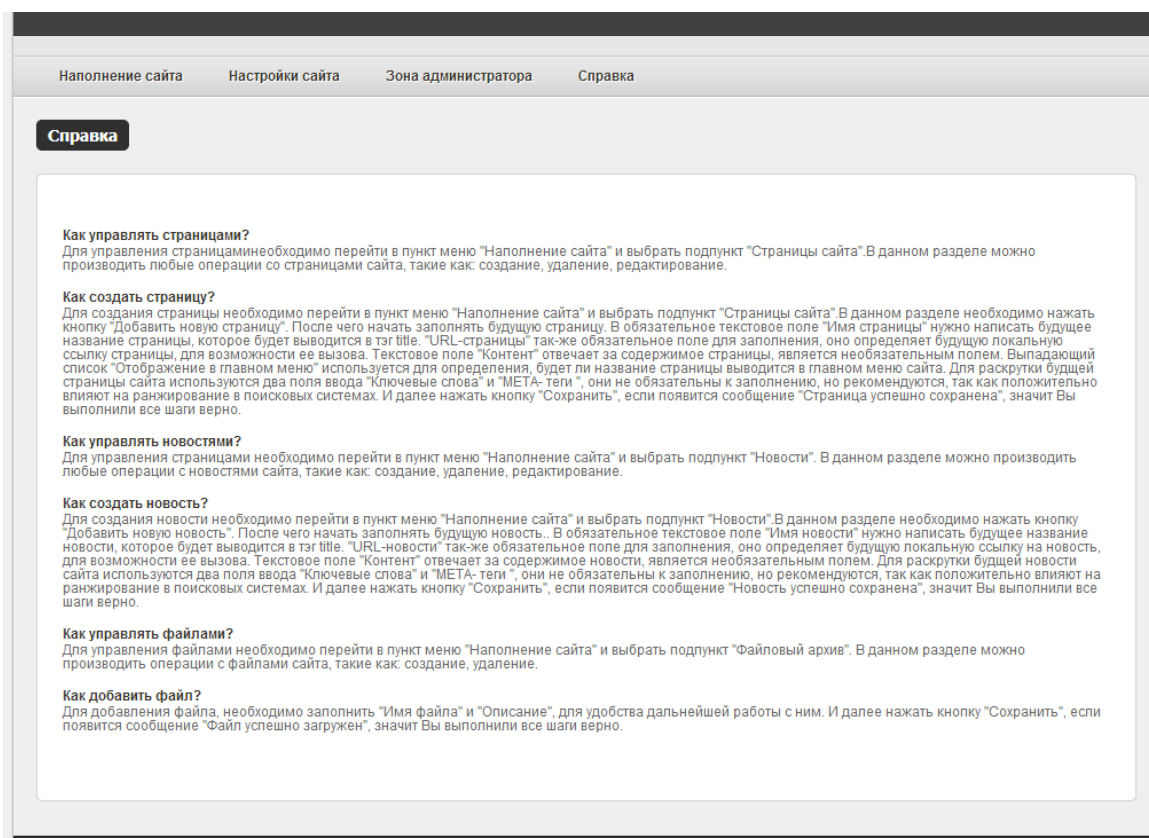


Рисунок 3.7. Довідкова інформація CMS

3.4 Вибір платформи для реалізації проекту

Мовний:

- HTML 5.0;
- PHP версії 5 і вище;
- MySQL.

Програмний:

- Adobe Photoshop CC;
- Денвер (PHP, MYSQL, Apache);
- Notepad ++;
- CKeditor 4.6.2;
- Microsoft WEB-Matrix 3.0.

3.5 Обґрунтування вибору мовних і програмних засобів

Мова HTML дозволяє розмічати електронний документ, який відображається на екрані з поліграфічним рівнем оформлення; результуючий документ може містити найрізноманітніші мітки, ілюстрації, аудіо- та відеофрагменти тощо. До складу мови ввійшли розвинені засоби для створення різних рівнів заголовків, шрифтових виділень, різні списки, таблиці і багато іншого.

Принципи гіпертекстової розмітки

HTML є описовою мовою розмітки документів, в ньому використовуються покажчики розмітки (теги). Тегів модель описує документ як сукупність контейнерів, кожен з яких починається і закінчується тегами, тобто документ HTML є не що інше, як звичайний ASCII - файл, з доданими до нього керуючими HTML - кодами (тегами). Оскільки HTML походить від SGML, в

ньому дозволено використовувати тільки три керуючих символу: горизонтальну табуляцію, переклад каретки і переведення рядка. Це полегшує взаємодію з різними операційними системами.

PHP (Personal Home Pages) - це серверна мова створення сценаріїв (або сторони сервера), розроблений спеціально для web. У HTML-сторінку можна впровадити код PHP, який буде виконуватися при кожному її відвідуванні. Код PHP інтерпретується Web-сервером і генерує HTML або інший висновок, що спостерігається відвідувачем сторінки.

Розробка PHP була розпочата в 1994 році і спочатку виконувалася однією людиною,

Расмусом Лердорфом (Rasmus Lerdorf). Ця мова був прийнятий поруч талановитих людей і зазнав три основні редакції, поки не став широко використовуваним і зрілим продуктом, з яким ми маємо справу сьогодні. До січня 2001 року він використовувався майже в п'ять мільйонів доменів в усьому світі, і їх кількість продовжує швидко рости. Кількість доменів, в яких в даний час використовується PHP, можна з'ясувати на сторінці <http://www.php.net/usage.php>.

PHP - це продукт з відкритим вихідним кодом (Open Source). У користувача є доступ до вихідного коду. Його можна використовувати, змінювати і вільно поширювати іншим користувачам або організаціям.

Спочатку PHP було скороченням від Personal Home Page (Персональна початкова сторінка), але потім ця назва була змінена відповідно до рекурсивним угодою по найменуванню GNU (GNU = Gnu's Not Unix) і тепер означає PHP Hypertext Preprocessor (Препроцесор гіпертексту PHP).

В даний час останньою версією PHP є PHP 7.1

Адреса початкової сторінки для PHP - <http://www.php.net>

В Число конкурентів PHP входять Perl, Active Server Pages (ASP), Java Server Pages (JSP).

PHP має безліч переваг в порівнянні з цими продуктами, серед яких найбільш значними є:

- Висока продуктивність.
- Наявність інтерфейсів до багатьох систем баз даних.
- Вбудовані бібліотеки для виконання багатьох загальних завдань, пов'язаних з web.
- Низька вартість.
- Простота вивчення та використання.
- Переносимість.
- Доступність вихідного коду [11].

MySQL. - дуже швидка, надійна система керування базами даних (СУБД). База даних дозволяє ефективно зберігати, шукати, сортувати і отримувати дані. Сервер MySQL керує доступом до даних, дозволяючи працювати з ними одночасно декільком користувачам, забезпечує швидкий доступ до даних і гарантує надання доступу тільки мають на це право користувачам. Отже, MySQL є багатокористувацьким, багатопотоковим сервером. Він застосовує SQL (Structured Query Language -Мова структурованих запитів), що використовується по всьому світу стандартна мова запитів в бази даних. MySQL з'явився на ринку в 1996 році, але його розробка почалася ще в 1979 р В даний час, після десяти років свого існування, ця система завоювала приз читацьких симпатій журналу Linux Journal.

В даний час пакет MySQL доступний як програмне забезпечення з відкритим вихідним кодом, але в разі потреби можна отримати і комерційні ліцензії.

До конкурентам MySQL, крім інших відносяться, PostgreSQL, Microsoft SQL Server, Oracle.

MySQL має багато переваг, в тому числі високою продуктивністю, низькою вартістю, простотою конфігурації та вивчення, переносимість та доступністю коду.

Web-сервер Apache - це програма, яка виконує функції http-сервера. Саме з її допомогою і буде функціонувати web - сервер. Ця програма виконує всі

необхідні функції, під її керівництвом працює більшість ресурсів мережі.

Дана CMS не виняток, в силу гнучкості і універсальності Apache використовує саме цей сервер [12].

3.6 Розробка бази даних CMS

Функціонування розробляється CMS ґрунтується на використанні бази даних, в якій буде зберігатися поступаємо від адміністративних (в тому числі призначених для користувача) скриптів інформація.

Для зберігання інформації в базі даних необхідно попередньо визначити групи і параметри даних, звести цю інформацію в реляційні таблиці і встановити між ними зв'язки. Крім того, необхідно задати первинні ключі та індекси, нормалізувати структуру. Кінцевим результатом проектування буде схема БД і типовий сценарій SQL на її основі.

На даному етапі розробки для роботи системи використовується п'ять таблиць:

- Admin - таблиця для зберігання користувачів і їх паролів;
- Page - таблиця для зберігання сторінок сайту;
- News - таблиць для зберігання новин сайту;
- Files - таблиця для зберігання даних про завантажені файли;
- Log - таблиця, для зберігання інформації про спроби входу в адміністративну частину сайту: вдалих (1) і невдалих (0).

Тип даних для поля ID в усіх таблицях вибраний виходячи з того, що, для невеликого сайту цілком достатньо зарезервувати 9999 осередків для новин, сторінок і інших даних майбутнього сайту; для поля використовується тип INT; також полю присвоєно властивість "AUTO_INCREMENT", для неможливості створення запису з однаковим ID.

Поле Title в таблицях "Page" і "News" має максимальну довжину 255 символів оскільки число букв в назві статей може бути різним. Використання полів змінної довжини (тип VARCHAR, а не CHAR) дозволяє уникнути зберігання непотрібної інформації, тому довжина поля визначається максимально можливими або достатніми для більшості записів значеннями.

У таблиці Admin зберігаються облікові записи користувачів, необхідні для авторизації. Тобто при введенні в форму авторизації пароль введений користувачем під своїм «логіном», звіряється з паролем занесеним в базу даних. Варто зазначити що паролі в колонці Password зберігаються в зашифрованому (md5 - хеш) вигляді. Це забезпечує механізм захисту від несанкціонованого входу в систему адміністрування сторонніх осіб. Надалі при розвитку системи планується ввести в таблицю поле категорії, для розмежування повноважень між користувачами системи, такими як адміністратор сайту, редактор контенту і т.д.

Таблиця 3.1 – Admin для зберігання користувачів та їх паролів

Поле	Тип	Нуль	Коментарі
id	int(4)	Ні	Унікальний індекс користувача
login	varchar(255)	Ні	Логін користувача
password	varchar(255)	Ні	Пароль користувача
name	varchar(255)	Ні	Ім'я користувача
email	varchar(255)	Ні	E-mail користувача

У таблиці Files для підвищення продуктивності системи всі зображення зберігаються в директорії «www / userfiles», а в базі даних знаходяться їхні імена, шляхи до файлів зображень і їх опис, що дозволяє кешувати все зображення на стороні HTTP-сервера. Максимальна довжина шляху в більшості операційних систем обмежується 255 символами.

Таблиця 3.2 – Files для зберігання даних про завантажені файли

Поле	Тип	Нуль	Коментарі
id	int(4)	Ні	Унікальний індекс користувача
name	varchar(255)	Ні	Назва файлу
description	varchar(255)	Ні	Опис файлу
userfile	varchar(255)	Ні	Напрямок до файлу

У таблиці Log зберігаються логи спроб входу в адміністративну частину сайту. У таблицю записується дата спроби, логін, який намагався авторизуватися, IP адреса і статус авторизації (виконаний вхід або ж авторизація закінчилася невдачею) [13].

Таблиця 3.3 – Log для зберігання інформації про спроби входу

Поле	Тип	Нуль	Коментарі
id	int(4)	Ні	Унікальний індекс користувача
login	varchar(255)	Ні	Логін намагався увійти в адмінку
date	datetime	Ні	Дата спроби увійти в адмінку
ip	varchar(16)	Ні	IP намагався увійти в адмінку
success	int(1)	Ні	Успішний чи ні був вхід (1 або 0)

Таблиця 3.4 – Page для зберігання сторінок сайту

Поле	Тип	Нуль	Коментарі
id	int(4)	Ні	Унікальний індекс користувача
title	varchar(255)	Ні	Ім'я сторінки для відображення в меню
title_url	varchar(255)	Ні	URL для відображення в адресному рядку
content	mediumtext	Ні	Контент сторінки
menu	int(1)	Ні	Відобразити чи ні в головному меню сайту
keywords	varchar(255)	Ні	Ключові слова сторінки
description	varchar(255)	Ні	МЕТА- теги сторінки

Таблиця 3.5 – News для зберігання новин сайту

Поле	Тип	Нуль	Коментарі
id	int(4)	Ні	Унікальний індекс користувача
title	varchar(255)	Ні	Ім'я новини для відображення в списку
title_url	varchar(255)	Ні	URL для відображення в адресному рядку
content	mediumtext	Ні	контент новини
keywords	varchar(255)	Ні	Ключові слова новини
description	varchar(255)	Ні	МЕТА- теги новини
date	timestamp	Ні	Дата створення новини

3.7 Принцип роботи системи

При зверненні до «центральному» файлу `index.php`, в кореневій директорії, рядком: `<include ('config.php');>` підключається "конфігураційний" файл `config.php`. Його завдання становить підключення класів і функцій, а так само підключення обходимо шаблону. Крім цього, файл `config.php` визначає параметри підключення до бази даних:

- Визначає ім'я сервера бази даних.
- Визначає ім'я бази даних.
- Визначає логін і пароль користувача бази даних.
- Визначає кодування підключення до бази даних.

3.8 Функції конфігураційного файлу і їх використання

Конфігураційний файл містить в собі інструкції (константи, класи і функції), для виконання різних дій, необхідних для генерації сторінок сайту, а також адміністративного розділу.

Розглянемо базові функції, завдяки яким відбувається генерація сторінок. Перш за все, необхідно зробити підключення до бази даних. Відбувається це завдяки стандартної функції: `mysql_connect ($hostname, $username, $password);`

Після успішного підключення, починається генерація сторінки, завдяки таким функціям:

- `get_menu ()` - отримує головне меню для сайту (призначений для користувача розділ).
- `page_data_content ($ title_url)` - отримує сторінку з її URL.

Залежно від того, який GET параметр переданий на посилання, відбувається запит до бази даних з ним.

- `page_data_content_id ($ id)` - отримує сторінку по її ID.

Залежно від того, який GET параметр переданий на посилання, відбувається запит до бази даних з ним.

- `news_data_content ($ title_url)` - отримує новина по її URL.

Працює по аналогії з "`page_data_content ($ title_url)`", тільки вибирає з бази даних новини.

- `news_data ($ id)` - отримує новина по її ID.

Працює по аналогії з "`page_data_content_id ($ id)`", тільки вибирає з бази даних новини.

- `insert_data ()` - додавання даних в базу даних.

Застосовується для створення нових новин, сторінок і іншого. Записує в базу даних введені значення полів.

- `update_data ()` - оновлення даних в базі даних.

Застосовується для оновлення вже існуючих новин, сторінок і іншого. Перезаписує в базу даних тільки оновлені значення полів.

- `file_info ()` - отримує інформацію про всіх файлах з бази даних.

Застосовується для отримання в розділ "Управління файлами" таблиці значень вже існуючих записів в базі даних.

- `get_page_admin ()` - отримує інформацію про сторінки з бази даних.

Застосовується для отримання в розділ "Управління сторінками сайту" таблиці значень вже існуючих записів в базі даних.

- `get_news_admin ()` - отримує інформацію про новини з бази даних.

Застосовується для отримання в розділ "Управління новинами сайту" таблиці значень вже існуючих записів в базі даних.

Як вже було сказано вище, в файлі конфігурації підключений «шаблонизатор»:

```
include ($ _ SERVER [ 'DOCUMENT_ROOT' ]. ' / template / test / style.php');
```

Це PHP скрипт з підключеними CSS стилями. Виробляє парсинг контенту і вставляє його в заданий шаблон, інакше кажучи, формує тіло HTML - документа. Сам шаблон являє собою невеликі шматки HTML - розмітки, з синтаксичними конструкціями, які розуміє шаблонизатор. Даний метод дозволяє змінювати дизайн сторінок, при менших витратах часу, на відміну від методу «ручний верстки сторінок».

Шаблони можна розглядати як «розширення» програмного коду. Шаблони не тільки автоматизують виснажливий процес кодування, але і забезпечують структурний поділ проекту в робочих групах. Роль такого поділу зростає зі збільшенням обсягів проекту і чисельності груп, а також з ускладненням архітектури проекту, причому не тільки на стадії програмування, але і при подальшому супроводі програми.

Сказане варто пояснити на конкретному прикладі. Припустимо, у нас є команда розробників, що складається з web-дизайнерів і програмістів. В ідеалі група web-дизайнерів трудиться над створенням привабливого і зручного сайту, а група програмістів в цей час працює над ефективністю і широтою можливостей web-додатки. На щастя, шаблони помітно спрощують подібне структурування процесу.

Існує кілька різних підходів для створення шаблонів за допомогою мови PHP:

- впровадження HTML в код PHP;
- включення файлів в сторінку.

Хоча перша схема більш зрозуміла і простіше реалізується, вона також більшою мірою обмежує свободу дій. Головна проблема полягає в тому, що код PHP змішується з компонентами HTML, що утворюють макет сторінки. Виникаючі при цьому проблеми пов'язані не тільки з необхідністю потенційної підтримки одночасного доступу до сторінки і її модифікації, а й з підвищеною вірогідністю помилок при безпосередньому перегляді і редагуванні сторінок.

Друга схема в багатьох ситуаціях виявляється набагато зручніше першої. Проте, хоча структура «заголовок - основна частина - колонтитул» добре

підходить для структурування щодо малих сайтів з чітко визначеним форматом, зі збільшенням обсягів і складності проекту ці обмеження виявляються дедалі помітнішою. Спроби вирішення цих проблем привели до розробки нової схеми застосування шаблонів, більш складної в порівнянні з двома першими, але і володіє значно більшою гнучкістю. У цій схемі поділяються два головні компоненти web-додатки: дизайн і програмування. Подібний розподіл забезпечує можливість паралельної розробки (web-дизайн та програмування) без необхідності постійної координації протягом усього робочого циклу. Більш того, воно дозволяє в майбутньому модифікувати один компонент, не впливаючи на роботу іншого.

Для використання системи в повному обсязі, дизайнер може одночасно працювати з декількома шаблонами, що володіють одними і тими ж загальними атрибутами. У таких ситуаціях застосування технології об'єктно-орієнтованого програмування (ООП) виявляється особливо ефективним. З цієї причини всі функції побудови і виконання операцій з шаблонами оформлені у вигляді окремих файлів.

Нижче представлений якийсь базовий шаблон, створений на основі цієї технології.

Шаблон блока «news»

```
<?
// Отрывок кода из файла config.php
if(empty($_GET['news'])) $news = 'URL';
else $news = $_GET['news'];
$news_id= news_data($news);

function news_data($id)
{
    $query = sprintf(" SELECT * FROM news WHERE news.id = '%s' ",
                    mysql_real_escape_string($id));
    $result = mysql_query($query);
    $row = mysql_fetch_array($result);
    return $row;
}
?>
```

```
// Отрывок кода из файла style.php
<table>
  <tr>
    <td ><div><? echo $news_id['date'];?></div>
      <br>
      <strong><? echo $news_id['title'];?></strong><br>
      <br>
      <? echo $news_id['content'];?>
      <br>
      <br></td>
  </tr>
</table>
```

Рядок (`echo $news_id['date'], echo $news_id['title'], echo $news_id['content']`), інтерпретуються як відповідні імена змінних, а функція "echo" виводить їх на екран. Дизайнер буде сторінку на свій розсуд; все, що від нього потрібно, - включати у відповідні місця документа ці ключові рядки.

Кожна функція вирішує конкретну задачу, що відповідає тій чи іншій операції в процесі обробки шаблону. На найпростішому рівні цей процес можна розділити на чотири стадії:

- Реєстрація файлів - реєстрація всіх файлів, що обробляються сценаріями шаблонів.
- Реєстрація змінних - реєстрація всіх змінних, які повинні замінюватися своїми значеннями в зареєстрованих файлах.
- Обробка файлів - заміна всіх змінних, що знаходяться між обмежувачами, в зареєстрованих файлах.
- Висновок файлу - висновок оброблених зареєстрованих файлів в браузері [14].

3.9 Висновок

В третьому розділі надано опис розробленої системи управління вмістом веб-сайту.

До неї входять:

- Робота з системою адміністрування CMS
- Управління розділами(сторінками)
- Редагування розділі(сторінок)
- Додавання новин
- Редагування новин
- Розділ довідка.

Був обґрунтован вибор мовних і програмних засобів

Також була розроблена база даних CMS, яка на даному етапі складається із п'яти таблиць:

- Admin - таблиця для зберігання користувачів і їх паролів;
- Page - таблиця для зберігання сторінок сайту;
- News - таблиць для зберігання новин сайту;
- Files - таблиця для зберігання даних про завантажені файли;
- Log - таблиця, для зберігання інформації про спроби входу в адміністративну частину сайту: вдалих (1) і невдалих (0).

Був описан принцип роботи система та функції конфігураційного файлу і їх використання.

4 ОХОРОНА ПРАЦІ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної роботи бакалавра було розроблено інформаційну комп'ютерну систему вістом веб-сайту, і як результат було створено даний проект. Так як в процесі проектування використовувалося електричне обладнання, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера на якому буде розроблена дипломна робота.

4.1 Правові та організаційні основи охорони праці

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави, і особистої відповідальності працівників.

Державна політика в галузі охорони праці визначається відповідно до Конституції України Верховною Радою України і спрямована на створення належних, безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Відповідно до статті 3 Закону України «Про охорону праці» (далі – Закону) законодавство про охорону праці складається з Закону, Кодексу законів про працю України, Закону України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності" та прийнятих відповідно до них нормативно-правових актів, норм міжнародного договору (ратифіковані Конвенції і Рекомендації МОТ,

директиви Європейської Ради).

На законодавчому рівні визначено такі пріоритетні напрямки з безпеки праці:

- кожен працівник несе безпосередню відповідальність за порушення зазначених Законом, нормами і правилами вимог;
- напрямки реалізації конституційного права громадян на їх життя і здоров'я в процесі трудової діяльності:
- пріоритет життя і здоров'я працівників по відношенню до результатів виробничої діяльності підприємства;
- соціальний захист працівників, повне відшкодування збитків особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань;
- комплексне розв'язання завдань охорони праці;
- підвищення рівня промислової безпеки шляхом забезпечення суцільного технічного контролю за станом виробництв, технологій та продукції, а також сприяння підприємствам у створенні безпечних та нешкідливих умов праці;
- соціальний захист працівників, повне відшкодування збитків особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань;
- використання економічних методів управління охороною праці, участь держави у фінансуванні заходів щодо охорони праці;

Користувачі персональних комп'ютерів, для яких ця робота є головною, підлягають медичним оглядам: попереднім — під час влаштування на роботу і періодичним — протягом професійної діяльності раз на два роки. Жінок з часу встановлення вагітності та в період годування дитини грудьми до роботи з ПК не допускають.

Наявні трудові відносини між працівниками і роботодавцями в Україні за темою дипломного проекту регулюються Кодексом законів про працю (КЗпП)

України, відповідно до якого права працюючої людини на охорону праці охороняються всебічно та норми охорони праці неухильно інтегровані до правил внутрішнього розпорядку організації/підприємства [15]

4.2 Вимоги до приміщень

Згідно з розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Для зручності спільної роботи з іншими працівниками (обговорення ідей, з'ясування проблем і т.д.) в кімнаті є дивани і журнальний стіл, обставлені живими квітами. Також робочий процес пов'язаний з багатьма документами, теками, журналами для чого приміщення облаштоване принтером і шафою для зручності. Задля дотримання визначеного рівня мікроклімату в будівлі встановлено систему опалення та кондиціонування.

Для забезпечення потрібного рівного освітленості кімната має вікно та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

4.3 Вимоги до організації місця праці

Робочий стіл на досліджуваному місці також містить достатньо простору для ніг. Крісло, що використовується в якості робочого сидіння, є підйомно

поворотним, має підлокітники і можливість регулювання за висотою і кутом нахилу спинки, також воно м'яке і виконане з екологічної шкіри, що дає можливість працювати у комфорті. Екран монітору знаходиться на відстані 0.8 м, клавіатура має можливість регулювання кута нахилу 5-15°. Отже, за всіма параметрами робоче місце відповідає нормативним вимогам. Приміщення кабінету знаходиться на третьому поверсі дев'яти поверхової будівлі і має об'єм 78 м³, площу – 20 м². У цьому кабінеті обладнано одне місце праці .[16]

Температура в приміщенні протягом року коливається у межах 18–24°C, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум в лабораторії знаходиться на рівні 50 дБА. Система вентилявання приміщення — природна неорганізована, а опалення — централізоване.

Розміщення вікон забезпечує природне освітлення з коефіцієнтом природного освітлення не менше 1,5%, а загальне штучне освітлення, яке здійснюється за допомогою восьми люмінесцентних ламп, забезпечує рівень освітленості не менше 200 Лк.

У кабінеті є електрична мережа з напругою 220 В, яка створює небезпеку ураження електричним струмом. ПК та периферійні пристрої можуть бути джерелами електромагнітних випромінювань, аерозолів та шкідливих речовин (часток тонеру, оксидів нітрогену та озону).

За ступенем пожежної безпеки приміщення належить до категорії В. Кабінет оснащений переносним вуглекислотним вогнегасником ВВК-5 .

Наявна аптечка для надання долікарської допомоги, а також у кабінеті роблять вологе прибирання та щоденно провітрюють приміщення.

4.4 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.[17]

4.4.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.1). Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга $U=+220V \pm 5\%$;
- робочий струм $I=2A$;
- споживана потужність $P=350 \text{ Вт}$.

Таблиця 4.1 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів(види робіт)	Кількісна оцінка	Нормативні документи
1	2	3	4
фізичні:			
- підвищена температура поверхонь обладнання	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	2	[18]

- підвищений рівень шуму на робочому місці	-//-	2	[18]
- підвищений рівень вібрації	-//-	2	[18], [19]
- підвищена або знижена вологість повітря	//-	2	[18]
- підвищена або знижена рухливість повітря	-//-	1	[18]
- підвищений рівень іонізуючого випромінювання в робочій зоні	-//-	2	[18], [20]
- підвищений рівень електромагнітного випромінювання	-//-	2	[20]
- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	-//-	4	[21],[22]
- підвищений рівень статичної електрики	-//-	2	[21]
- підвищена напруженість електричного поля	-//-	2	[20]
- підвищена напруженість магнітного поля	-//-	2	[20]
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[23]
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	[16]
- підвищена яскравість світла	порушення умов праці (організації місця праці-налагодження моніторів)	1	[16]
- понижена контрастність	-//-	1	[16]
хімічні:			

<p>- загазованість повітря робочої зони, яка впливає на організм людини через органи дихання та надає токсичну і канцерогенну дію</p>	<p>від експлуатації сканерів, принтерів для роботи – O₃, оплавлення електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів, транзисторів й інше в ЕОМ та системах кондиціонування повітря - CO, CO₂, SO₂, P₂O₅, H₂S, HCl, H, NH₃, ClF₃, F₂O₂, F₂O₃, SeO₂. SeF₆, TeF₆, COCl₂, SO₂F₂, інш.</p>	3	[23], [24], [25], [26]
психологічні:			
<p>- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)</p>	<p>- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи</p>	4	[15], [16]
<p>- фізичні (статичне – сидіння)</p>	<p>порушення умов праці (організації місця праці-сидіння користувача,) та організації робочого часу - безпервна робота)</p>	2	[15], [16]

4.5 Пожежна безпека

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціонування,. Небезпека загоряння пов'язана з особливістю комп'ютерів - із значною кількістю щільно розташованих на монтажній платі і блоках

електронних вузлів і схем, електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів. Надійна робота окремих елементів і мікросхем в цілому забезпечується тільки в певних інтервалах температури, вологості і при заданих електричних параметрах. При відхиленні реальних умов експлуатації від розрахункових можуть виникнути пожежонебезпечні ситуації.

Висока щільність елементів в електронних схемах призводить до значного підвищення температури окремих вузлів (80...100 °C). При проходженні електричного струму по провідниках і деталей виділяється тепло, що в умовах їх високої щільності може привести до перегріву, і може служити причиною запалювання ізоляційних матеріалів. Слабкий опір ізоляційних матеріалів дії температури може викликати порушення ізоляції і привести до короткого замикання між струмоведучими частинами обладнання (шини, електроди). Також ймовірна небезпека внаслідок перевантаження напруги, розрядки зарядів статичної електрики, пошкодження обладнання та електропроводки. Електростатичний розряд виникає під час тертя двох ізолюваних матеріалів. Розряд статичної електрики може виникнути під час роботи вентилятора або комп'ютера. Кабельні лінії є найбільш пожежонебезпечними місцем. Наявність пального ізоляційного матеріалу, ймовірних джерел запалювання у вигляді електричних іскор і дуг, розгалуженість і недоступність роблять кабельні лінії місцем найбільш ймовірного виникнення і розвитку пожежі. Для зниження займистості і здатності поширювати полум'я кабелі покривають вогнезахисними покриттями. Проектом передбачено прокладати проводку: приховано, під знімною підлогою розділяючи негорючими діафрагмами, в малодоступних місцях.

Для гасіння пожеж в офісному приміщенні пропонується використовувати порошкові або вуглекислотні вогнегасники, так як вони є універсальними. Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця (батареї опалення, водопровідні труби, кабелі із

заземленим відкритим екраном), надійно захищені діелектричними щитками та/або сітками з метою недопущення потрапляння працівника під напругу. Дане приміщення оснащено системою автоматичної пожежної сигналізації, має 1 вогнегасник ВП-5 із зарядом вогнегасної речовини 8-12 кг, відповідно до вимог чинного законодавства України. Проходи до засобів пожежогасіння вільні, не захарашуються та у разі потреби забезпечувати евакуацію всіх людей, які перебувають у приміщенні через один евакуаційний вихід з дверима на шляху евакуації, що відчиняться в напрямку виходу з будівлі від робочого місця. В приміщенні наявна затверджена «План-схема евакуації з кабінету (приміщення)».

Пожежна безпека при застосуванні ЕОМ забезпечується:

- 1) системою запобігання пожежі,
- 2) системою протипожежного захисту,
- 3) організаційно-технічними заходами.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі і важкогорючі) не надається технічно можливим. Тому проектом передбачаються способи і засоби запобігання утворення (або внесення) в горюче середовище джерел запалювання, таких як:

- 1) застосування електроустаткування, відповідної пожежонебезпечної і вибухонебезпечної зонам відповідно до ПУЕ;
- 2) застосування в конструкції швидкодійних засобів захисного відключення можливих джерел запалення;
- 3) виключення можливості появи іскрового розряду в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення.

Приміщенню, у якому розташоване робоче місце, присвоюється II ступень вогнестійкості.

Потенційними джерелами запалювання можуть бути:

- 1) іскри і дуги короткого замикання;
- 2) електрична іскра при замиканні і розмиканні ланцюгів;
- 3) перегріву від тривалого перевантаження,
- 4) відкритий вогонь і продукти горіння,
- 5) наявність речовин, нагрітих вище за температуру самозаймання,
- 6) розрядна статична електрика.

Причинами можливого загоряння і пожежі можуть бути:

- 1) несправність електроустановки;
- 2) конструктивні недоліки устаткування;
- 3) коротке замикання в електричних мережах;
- 4) запалювання горючих матеріалів, що знаходяться в безпосередній близькості від електроустановки. [26]

4.6 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів

фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.7 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт Іа. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають. [17]

Дане приміщення обладнане системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією. У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря[17]. Рівні позитивних і негативних іонів у повітрі мають відповідати. Для забезпечення оптимальних параметрів мікроклімату в приміщенні проводяться перерви в роботі співробітників, з метою його провітрювання. Існують спеціальні системи кондиціонування, які забезпечують підтримання в приміщенні балансу оптимальних параметрів мікроклімату.

Таблиця 4.2 – Норми мікроклімату робочої зони об'єкту

Період року	Категорія робіт	ТемператураС ⁰	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

Контроль параметрів мікроклімату в холодний і теплий період року здійснюється не менше 3-х разів на зміну (на початку, середині, в кінці).

4.8 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

Освітленість приміщення має велике значення при роботі на ПЕОМ. Вона багато в чому визначається колірною і мережевий обстановкою. Для зменшеного поглинання світла стеля і стіни вище панелей (1,5-1,7м.). Якщо вони не облицьовані звукопоглинальним матеріалом, фарбуються білою водоемульсійною фарбою (коефіцієнт відбиття повинен бути не менше 0,7).

Для забарвлення стіни панелей рекомендується віддавати перевагу світлим фарбам.

Основний потік природного світла при цій повинен бути зліва. Не допускається спрямування основного світлового потоку природного світла праворуч, ззаду і спереду працівника на ПЕОМ.

Робота на ПЕОМ може здійснюватися за таких видах освітлення:

- загальному штучному освітленні, коли відео монітори розташовуються по периметру приміщення або при центральному розташуванні робочих місць у два ряди по довжині кімнати з екранами, звернені в протилежні сторони;

- суміщене освітлення (природне + штучне) тільки при одному і трьох рядном розташуванні робочих місць, коли екран і поверхню робочого столу знаходяться перпендикулярно світла несучій стіні. При цьому штучне освітлення буде виконане стельовими або підвісними люмінесцентними світильниками, рівномірно розміщеними по стелі рядами паралельно світловим прорізам так, щоб екран відео монітора знаходився в зоні захисного кута світильника, і його проекції не доводилися на екран. Працюючі на ПЕОМ не повинні бачити відображення світильників на екрані. Застосовувати місцеве освітлення при роботі на ПЕОМ не рекомендується.

Природне освітлення, коли робочі місця з ПЕОМ розташовуються в один ряд по довжині приміщення на відстані 0,8 - 1,0 м від стіни з віконними прорізами, і екрани знаходяться перпендикулярно цієї стіни. Основний потік природного світла при цій повинен бути зліва. Не допускається спрямування основного світлового потоку природного світла праворуч, ззаду і спереду працює на ПЕОМ. Оптимальна відстань очей до екрана відео монітора повинна становити 60-70 см, допустиме не менше 50 см. Розглядати інформацію ближче 50 см не рекомендується.

У проєкті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла в світильниках загального освітлення, оскільки люмінесцентні лампи мають високу потужність (80 Вт), тривалий термін служби (до 10000 годин), спектральний складом випромінюваного світла, близький до сонячного. При експлуатації ЕОМ виконується зорова робота IVв розряду точності (середня точність). При цьому нормована освітленість на робочому місці (Ен) рівна 200 лк. Джерелом природного освітлення є сонячне світло.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення [27].

Розрахунок освітлення.

Для побутових приміщень світловий коефіцієнт приймається не менше за 1/10:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10} \right) \cdot S_n, \quad (4.1)$$

де S_b – площа віконних прорізів, m^2 ;

S_n = площа підлоги, m^2

$$S_n = a \cdot b = 4 \cdot 5 = 20 \text{ м}^2, \quad (4.2)$$

$$S = 1/8 \cdot 20 = 2,5 \text{ м}^2.$$

Приймаємо 1 вікно площею $S = 2,5 \text{ м}^2$.

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 5 м, ширина 5 м, світильниками ЛПО2П, оснащеними лампами типу ЛБ (дві по 80 Вт) з світловим потоком 5400 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників n виробляється по формулі:

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (4.3)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, m^2 ; $S = 20 \text{ м}^2$;

Z – поправочний коефіцієнт світильника ($Z = 1,15$ для ламп розжарювання та ДРЛ; $Z = 1,1$ для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу, отримуємо:

$$n = \frac{300 * 20 * 1.1 * 1.5}{5400 * 0.575 * 2} \approx 1.6 \quad (4.4)$$

4.9 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при V приміщення $> 40 \text{ м}^3$ на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП.

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні

4.10 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Застосовують різні електричні захисні засоби від ураження струмом:

а) Ізолюючі - ізолюють людини від струмоведучих або заземлених частин, а так-же від землі. Вони діляться на основні та додаткові.

б) Основні - володіють ізоляцією, здатної довго витримувати робоче напругу електроустановки і тому ними дозволяється стосуватися струмоведучих частин, знаходячи-трудящих під напругою. До них відносяться: в електроустановках до 1000 Вт - діелектричної рукавички, ізолюючі штанги, ізолюючі і електровимірювальні кліщі і т.д .; понад 1000 Вт - ізолюючі штанги, і електровимірювальні кліщі, а також кошти для ремонтних робіт під напругою понад 1000Вт.

в) Запобіжні - володіють ізоляцією нездатною витримати робоча напруга електроустановки, і тому вони не можуть самостійно захищати людину від ураження струмом під цим напругою. Їх значення - посилити захисні дії основних і ізолюючих засобів, разом з якими вони повинні застосовуватися, при чому при використанні основних захисних засобів достатньо застосування одного запобіжного захисного засобу. До запобіжних відносяться засоби в електроустановках до 1000 Вт - діелектричні калоші килимки, а також ізолюючі підставки.

Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом, приміщення в якому проводяться всі роботи відносяться до першого класу (без підвищеної небезпеки). Під час роботи використовуються електроустановки з напругою живлення 36 В, 220 В, та 360 В. Опір контура заземлення повинен мати не більше 4 Ом.

Розрахунок проводять за допомогою методу коефіцієнта використання (екранування) електродів. Коефіцієнт використання групового заземлювача η – це відношення діючої провідності цього заземлювача до найбільш можливої його провідності за нескінченно великих відстаней між його електродами. Коефіцієнт використання вертикальних заземлювачів η_v в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4...0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача η_c .

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів $R_{шт.з.}$:

$$R_{шт.з.} = \frac{R_d \cdot R_{пр.з.}}{R_{пр.з.} - R_d}, \quad (4.5)$$

де $R_{пр.з.}$ – опір природних заземлювачів; R_d – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то $R_{шт.з.} = R_d$.

Підставивши числові значення у формулу, отримуємо:

$$R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом} \quad (4.6)$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ , Ом·м. Приблизне значення питомого опору глини приймаємо $\rho=40$ Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту, $\rho_{розр.}$, Ом·м, визначається відповідно для вертикальних заземлювачів $\rho_{розр.в.}$, і горизонтальних $\rho_{розр.г.}$, Ом·м за формулою:

$$\rho_{\text{розр.}} = \psi \cdot \rho, \quad (4.7)$$

де ψ – коефіцієнт сезонності для вертикальних заземлювачів I кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $\rho_{\text{розр.в}}=1,7$ і горизонтальних $\rho_{\text{розр.г}}=5,5$ Ом·м.

$$\rho_{\text{розр.в}} = 1,7 \cdot 40 = 68 \text{ Ом} \cdot \text{м} \quad (4.8)$$

$$\rho_{\text{розр.г}} = 5,5 \cdot 40 = 220 \text{ Ом} \cdot \text{м}$$

4) Розраховується опір розтікання струму вертикального заземлювача $R_{\text{в}}$, Ом, за.

$$R_{\text{в}} = \frac{\rho_{\text{розр.в}}}{2 \cdot \pi \cdot l_{\text{в}}} \cdot \left(\ln \frac{2 \cdot l_{\text{в}}}{d_{\text{ст}}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_{\text{в}}}{4 \cdot t - l_{\text{в}}} \right), \quad (4.9)$$

де $l_{\text{в}}$ – довжина вертикального заземлювача (для труб - 2–3 м; $l_{\text{в}}=3$ м);

$d_{\text{ст}}$ – діаметр стержня (для труб - 0,03–0,05 м; $d_{\text{ст}}=0,05$ м);

t – відстань від поверхні землі до середини заземлювача, яка визначається за ф.:

$$t = h_{\text{в}} + \frac{l_{\text{в}}}{2}, \quad (4.10)$$

де $h_{\text{в}}$ – глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м} \quad (4.11)$$

$$R_B = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом} \quad (4.12)$$

5) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання η_B :

$$n = \frac{2 \cdot R_B}{R_d} = \frac{2 \cdot 18,5}{4} = 9,25 \quad (4.13)$$

I визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки $\eta_B = 0,57$ (табличне значення).

б) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання n_B , шт:

$$n_B = \frac{2 \cdot R_B}{R_d \cdot \eta_B} = \frac{2 \cdot 18,5}{4 \cdot 0,57} = 16,2 \approx 16 \quad (4.14)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача l_c , м:

$$l_c = 1,05 \cdot L_B \cdot (n_B - 1), \quad (4.15)$$

де L_B – відстань між вертикальними заземлювачами, (прийняти за $L_B = 3\text{м}$);

n_B – необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м} \quad (4.16)$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) R_Γ , Ом:

$$R_\Gamma = \frac{\rho_{\text{розр.}\Gamma}}{2 \cdot \pi \cdot l_c} \cdot \ln \frac{2 \cdot l_c^2}{d_{\text{см}} \cdot h_\Gamma}, \quad (4.17)$$

де $d_{\text{см}}$ – еквівалентний діаметр смуги шириною b , $d_{\text{см}} = 0,95b$, $b = 0,15$ м;

h_Γ – глибина закладання горизонтальних заземлювачів (0,5 м);

l_c – довжина з'єднувальної стрічки горизонтального заземлювача l_c , м

$$R_\Gamma = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом} \quad (4.18)$$

9) Визначається коефіцієнт використання горизонтального заземлювача η_c відповідно до необхідної кількості вертикальних заземлювачів n_b .

Коефіцієнт використання з'єднувальної смуги $\eta_c = 0,3$ (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{зар}} = \frac{R_b \cdot R_\Gamma}{R_b \cdot \eta_c + R_\Gamma \cdot n_b \cdot \eta_b} \leq R_d. \quad (4.19)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{\text{зар}} < 4$ Ом, а саме:

$$R_{\text{заг}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_{\text{д}} \quad (4.20)$$

3) При виникненню пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як:

- іскри і дуги коротких замикань;
- перегрів провідників, резисторів та інших радіодеталей ПЕОМ, від тривалої перевантаження та наявності перехідного опору;
- іскри при розмиканні і розмиканні ланцюгів;
- розряди статичної електрики;
- необережному поводженню з вогнем, а також вибухи газо-повітряних і паро-повітряних сумішей.

Важливу увагу слід звернути на пожежну безпеку підприємства в цілому і окремих його приміщень. В приміщеннях не повинно накопичуватися сміття, непотрібний папір, мотлох та ін. речі, які не використовуються у виробничому процесі. Наявний вільний аварійний вихід за межі приміщення в разі пожежі, бути передбачені вогнегасники. Вони повинні бути в робочому стані і перевірятися згідно з нормами. У приміщеннях повинна бути пожежна сигналізація, вогнегасник. У разі виникнення пожежі необхідно повідомити в найближчу пожежну частину, убезпечити інших працівників і по можливості прийняти кроки по запобіганню можливих наслідків та усуненню пожежі[28].

4.11 Висновки до розділу 4

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Була наведена схема, розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

ВИСНОВКИ

У дипломному проекті був розроблений програмний продукт, представляє собою набір скриптів, за допомогою яких відбувається управління веб-контентом і призначений для швидкого створення простих сайтів за короткий термін.

У першому розділі було розглянено що таке система управління вмістом web-сайту та їх класифікації, а також існуючі CMS системи.

У другому розділі було визначено що, першим етапом створення будь-якого програмного продукту є постановка завдання. Маючи точно сформульовані цілі і завдання, можна приступати до проектування структури бази даних і всієї програми в цілому.

В третьому розділі був наданий опис розробленої системи управління вмістом веб-сайту. Також була розроблена база даних CMS, яка на даному етапі складається із п'яти таблиць.

Залишається додати тільки те, що CMS - це закономірний продукт розвитку інтернет-технологій, що відображає високі вимоги до інтернет-представництв будь-якого роду і призначений не для розробника, а для кінцевого користувача.

Крім того, не секрет, що технічний персонал багатьох компаній, зайнятий обслуговування внутрішньо корпоративних бізнес-процесів, не в змозі охопити багато нюансів використовуваних на інтернет-ринку технологій. Все складніше і складніше відповідати зростаючим стандартам в той час, як сайтобудівництво перетворилося на серйозну індустрію, зайнявши самостійне положення в ряду ІТ.

ПЕРЕЛІК ДЖЕРЕЛ І ПОСИЛАНЬ

1. Види сайті [Електронний ресурс] //URL:
<http://fotogid.com/uk/1028.html>
2. Визначення CMS [Електронний ресурс]// URL:
<http://webstudio2u.net/ua/programming/96-cms.html>
3. Статичний сайт, динамічний сайт[Електронний ресурс] // Веб-студія: інформаційний сайт. URL: <http://webstudio2u.net/ru/design-web/391-static-or-dynamic.html>
4. М.А.Курилов, С.В.Терещенко. Класифікація систем керування вмістом web-ресурсів та їх використання для розробки сайта дистанційного навчання// ІШтучний інтелект . 2010. №3. С. 648-654.
5. Обзор бесплатных CMS [Електронний ресурс] // URL:
<http://sitepark.ua/obzor-besplatnykh-cms-dlya-sajta-analiz-i-sravnenie>
6. Основнык требования к CMS[Електронний ресурс]// URL:
<http://thewall.by/osnovnye-trebovaniya-k-cms-s-uchetom-seo/>
7. Себеста Р. В. Основные концепции языков программирования. / Р. В. Себеста. – М.: «Вильямс», 2001. – 672 с.
8. Встановлення денвера(PHP, MySQL, Apache) [Електронний ресурс]// URL: <https://blog.denisbondar.com/post/apache24php56win7>
9. Lerman J. Programming Entity Framework. 2nd Edition. / J. Lerman. – O'Reilly, 2010. – 920 p
10. Лаура Томсон и Люк Веллинг. Разработка Web-Приложений на PHP и MySQL. 2-е издание: Пер. С англ. – М.Издательский дом «Вильямс», 2009.-800с
11. Кузнецов М.В., Симдянов И.В., Голышев С.В. PHP 5. Практика разработки Web-сайтов. / М.В. Кузнецов, И.В. Симдянов, С.В. Голышев. – СПб.: «БХВ-Петербург», 2005. – 960 с.

12. Веллинг Л. MySQL. Учебное пособие.: Пер. с англ. / Л. Веллинг, Л. Томсон. – М.: "Вильямс", 2005. – 304 с.
13. Кузнецов С. Д. Основы баз данных. 2-е изд. / С. Д. Кузнецов. – М.: «БИНОМ», 2007. – 484 с
14. Романюк О.Н. Веб-дизайн і комп'ютерна графіка. Навчальний посібник / О.Н. Романюк, Д.І. Кательніков, О.П. Косоветь. – В.: ВНТУ, 2007. -
15. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно- обчислювальних машин
16. ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин
17. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень.
18. ДСТУ ГОСТ 12.1.012-90 ССБТ. Вибрационная безопасность. Общие требования.
19. ГОСТ 12.1.006-84 ССБТ Электромагнитные поля радиочастот. Общие требования безопасности. Допустимые уровни на рабочих местах и требования к проведению контроля.
20. ГОСТ 12.4.030-81 ССБТ. Электробезопасность.Защитное заземление. Зануление
21. ГОСТ 13109-97 „Электрическая энергия. Совместимость технических средств электромагнитных. Нормы качества электроэнергоснабжения общего назначения”
22. НПАОП 40.1-1.21-98 Правила безпечної експлуатації електроустановок споживачів
23. ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування
24. ГОСТ 12.1.005-88 ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны

25. ГОСТ 12.1.044-89 ССБТ. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения
26. НАПБ Б.03.002-2007 Норми визначення категорії приміщень будинків та зовнішніх установок за вибухопожежною та пожежною безпекою
27. ДБН В.2.5-28:2015 Освітлення у приміщеннях.
28. НПАОП 40.1-1.01-97 Класифікація приміщень за ступенем небезпеки ураження електричним струмом.

Додаток А.

Лістинг програми

```

<?
//Данные подключения к БД MySQL
$hostname = 'localhost';
$username = 'plus500';
$password = '147852';

// Подключаемся к серверу MySQL
$db = mysql_connect($hostname, $username, $password) or die('Не удалось
подключится к серверу БД!');

// Устанавливаем нужную кодировку общения с сервером
mysql_query ("set_client='utf8'");
mysql_query ("set character_set_results='utf8'");
mysql_query ("set collation_connection='utf8_general_ci'");
mysql_query ("SET NAMES utf8");

// Выбираем нужную БД
mysql_select_db('plus500') or die('БД не найдена!');

/*      Функции файла config.php, используемые для генерации страниц
сайта: */
/*      Функции файла config.php, используемые для генерации страниц
сайта: */

//Получаем меню для сайта
function get_menu()
{
$query = "SELECT * FROM page WHERE menu = 1"; //Запрос на вывод всех
значений меню из базы
$result = mysql_query($query);
$result = db_result_to_array($result); //Обрабатываем запрос, для приведения его
в вид массива
return $result;
}

//Функция построчной выборки результатов из базы
function db_result_to_array($result)

```

```

{
$res_array = array();
$count = 0;
while($row = mysql_fetch_array($result)) //Запускаем цикл пока не кончатся
результаты из таблицы
    {
        $res_array[$count] = $row;
        $count++;
    }
return $res_array;
}

```

```

//Получаем информацию о странице
function page_data($title)
{
$query = sprintf(" SELECT * FROM page WHERE page.title_url = '%s' ",
mysql_real_escape_string($title));
$result = mysql_query($query);
$row = mysql_fetch_array($result);
return $row;
}

```

```

//Функция получения динамические значения из базы
function get_data_admin($table,$id)
{
    $query = sprintf(" SELECT * FROM $table WHERE $table.id = '%s' ",
        mysql_real_escape_string($id));
    $result = mysql_query($query);
    $row = mysql_fetch_array($result);
    return $row;
}

```

```

// Функция для получения страниц сайта в админке(в список)
function get_page_admin()
{
    $query = "SELECT * FROM page";
    $result = mysql_query($query);
    $result = db_result_to_array($result);
    return $result;
}

```

```
//Функция обновления динамических данных
function update_data($table, $id, $row, $data)
{
    $i = 0;
    foreach($data as $item=>$key):
        mysql_query(" UPDATE $table SET $row[$i] = '$key' WHERE id = '$id'
");
        $i++;
    endforeach;
}
```

```
//Функция получения динамических данных о странице на основе URL
function page_data_content($title)
{
    $query = sprintf(" SELECT * FROM page WHERE page.title_url = '%s' ",
        mysql_real_escape_string($title));
    $result = mysql_query($query);
    $result = db_result_to_array($result);
    return $result;
}
```

```
//Функция получения динамических данных о странице на основе ID
function page_data_content_id($id)
{
    $query = sprintf(" SELECT * FROM page WHERE page.id = '%s' ",
        mysql_real_escape_string($id));
    $result = mysql_query($query);
    $result = db_result_to_array($result);
    return $result;
}
```

```
//Функция удаления динамических данных по ID
function delete_data($table, $id) { mysql_query(" DELETE FROM $table
WHERE id='$id' "); }
```

```
//Функция ДОБАВЛЕНИЯ динамических данных
function insert_data($table, $row, $data)
{
    $row = implode(',',$row);
    $data = implode(',',$data);
```

```

mysql_query("INSERT INTO $table ($row) VALUES($data) ");
}

//Функция получения динамических данных о ФАЙЛЕ
function file_info($title)
{
    $query = sprintf(" SELECT * FROM files WHERE files.userfile = '%s' ",
        mysql_real_escape_string($title));
    $result = mysql_query($query);
    $result = db_result_to_array($result);
    return $result;
}

// Функция для получения НОВОСТЕЙ сайта в админке(в список)
function get_news_admin()
{
    $query = "SELECT * FROM news";
    $result = mysql_query($query);
    $result = db_result_to_array($result);
    return $result;
}

//Функция получения динамических данных о НОВОСТИ
function news_data_content($title)
{
    $query = sprintf(" SELECT * FROM news WHERE news.title_url = '%s' ",
        mysql_real_escape_string($title));
    $result = mysql_query($query);
    $result = db_result_to_array($result);
    return $result;
}

//Получаем информацию о НОВОСТИ
function news_data($id)
{
    $query = sprintf(" SELECT * FROM news WHERE news.id = '%s' ",
        mysql_real_escape_string($id));
    $result = mysql_query($query);
    $row = mysql_fetch_array($result);
    return $row;
}

//Скрипт получения меню административной панели
session_start(); //Открываем сессию для чтения

```

```
if (isset($_SESSION['user_id'])) { //Проверка не пуста ли сессия
```

```
//Если пуста, отображаем страницу авторизации с сообщением, для устрашения
:)
```

```
else { header("Location: index.php?error=1"); }
```

```
?>
```

```
<ul id="nav">
```

```
<li>
```

```
<a href="#">Наполнение сайта</a>
```

```
<div>
```

```
<ul>
```

```
<li><a href="content-page.php">Страницы сайта</a></li>
```

```
<li><a href="content-news.php">Новости</a></li>
```

```
<li><a href="content-files.php">Файловый архив</a></li>
```

```
</ul>
```

```
</div>
```

```
</li>
```

```
<li>
```

```
<a href="#">Настройки сайта</a>
```

```
<div>
```

```
<ul>
```

```
<li><a href="main-config-site.php">Основные</a></li>
```

```
</ul>
```

```
</div>
```

```
</li>
```

```
<li>
```

```
<a href="#">Зона администратора</a>
```

```
<div>
```

```
<ul>
```

```
<li><a href="admin-user.php">Управление администраторами</a></li>
```

```
<li><a href="admin-last-ip.php">Последние входы</a></li>
```

```
<li><a href="admin-private-ip.php">Ограничение доступа</a></li>
```

```
</ul>
```

```
</div>
```

```
</li>
```

```
<li>
```

```
<a href="faq.php">Помощь</a>
```

```
</li>
```

```
</ul>
```

<?

//Авторизация в административном разделе:

```

if ($_GET['exit']==1)
{
session_start();
session_unset(); // Удаляем все переменные сессии
session_destroy(); // Удаляем сессию
}

session_start();
if (isset($_SESSION['user_id']))
{header("Location: index.php");}

if (isset($_POST['login']) && isset($_POST['password']))
{
$login = mysql_real_escape_string($_POST['login']); // Экранизация
СПЕЦСИМВОЛОВ
$password = md5($_POST['password']);
// делаем запрос к БД и ищем админа с таким логином и паролем
$query = "SELECT * FROM `admin` WHERE `login`='{ $login}' AND
`password`='{ $password}' LIMIT 1";
$sql = mysql_query($query) or die(mysql_error());

// если такой пользователь нашелся
// то мы ставим об этом метку в сессии (допустим мы будем ставить ID
пользователя)
if (mysql_num_rows($sql) == 1)
{
session_start();
$row = mysql_fetch_assoc($sql);
$_SESSION['user_id'] = $row['id'];
$_SESSION['user_login'] = $row['login'];
$_SESSION['user_name'] = $row['name'];
$_SESSION['user_privilege'] = $row['privilege'];
$_SESSION['user_email'] = $row['email'];

//Пишел логи удачного входа
$query = "INSERT INTO `log` (`login`, `date`, `ip`, `success`) VALUES
('$login', '$date', '$ip', 1)";
$sql = mysql_query($query) or die(mysql_error());

```



```

header("Location: content-page.php");      // Редиректим в админку
}

//если такой пользователь не нашелся, то выводим сообщение об ошибке в
авторизации
else
{
    $query = "INSERT INTO `log` (`login`, `date`, `ip`, `success`) VALUES
('$login', '$date', '$ip', 0)";
    $sql = mysql_query($query) or die(mysql_error()); //Пишел логи
неудачного входа
    die('Логин или пароль не верны. ');
}

if ($_GET['error']==1)
{echo "Для получения доступа к этой странице необходимо
авторизоваться!";}

//Вывод имени, IP, последней даты входа в административном разделе:
include("../config.php");
session_start();
$login= $_SESSION['user_login'];
$query = "SELECT * FROM `log` WHERE `login`='{ $login}' AND
`success`=1' order by `date` desc limit 1,2";
$sql = mysql_query($query) or die(mysql_error());
$sql_as = mysql_fetch_assoc($sql);
$lastdate= $sql_as['date'];
$lastip= $sql_as['ip'];

//Управление страницами сайта в административном разделе:
if ($_GET['comand']=='new' & empty($_GET['save']))
{
?>
<form action="content-page.php?comand=new&save=1&t=page"
method="post" enctype="multipart/form-data" id="add-edit" >

    <label>Имя страницы </label><br /><textarea name="title" cols="100"
rows="1"/></textarea>
    <br /><br /><br />

    <label>URL-страницы </label><textarea name="title_url" cols="100"

```

```

rows="1">
    </textarea><br /><br /><br />

    <label>Контент </label><br /><textarea name="content" cols="55"
rows="11">
    </textarea><br /><br /><br />

    <label>В главном меню </label><br /><select name="menu">
    <option value="1">Отображать</option><option value="2">Не
отображать</option>
    </select><br /><br />

    <label>Ключевые слова </label><br /><textarea name="keywords"
cols="44" rows="5">
    </textarea><br /><br /><br />

    <label>META- теги </label><br /><textarea name="description"
cols="44" rows="5">
    </textarea><br /><br /><br />

    <input type="submit" class="button" value="Сохранить" />
</form>

<?    }

if ($_GET['comand']=='delete')
{
    $id = $_GET['id'];
    $table = $_GET['t'];
    $delete_data = delete_data($table, $id);
    echo "Страница успешно удалена!";
}

if ($_GET['comand']=='edit' & empty($_GET['save']))
{
    $id = $_GET['id'];
    $table = $_GET['t'];
    $page_data = get_data_admin($table, $id);
?>

<form action="content-
page.php?comand=edit&save=1&t=page&id=<?=$page_data[id];?>"
method="post" enctype="multipart/form-data" id="add-edit" >

```

```

        <label>Имя страницы </label><br /><textarea name="title" cols="100"
rows="1"/>
        <?=$page_data['title'];?></textarea><br /><br /><br />

        <label>URL-страницы </label><br /><textarea name="title_url"
cols="100" rows="1">
        <?=$page_data['title_url'];?></textarea><br /><br /><br />

        <label>Контент </label><br /><textarea name="content" cols="55"
rows="11">
        <?=$page_data['content'];?></textarea><br /><br /><br />

        <label>Отображение в главном меню </label><br />В данный момент
запись

        <?
            if($page_data['menu']==1){echo "отображается в главном меню.";}
            else {echo "не отображается в главном меню."};
        ?>

        <br />Действие: <select name="menu"> <option
value="1">Отображать</option> <option value="0">Не
отображать</option></select><br /><br />
        <label>Ключевые слова </label><br /><textarea name="keywords"
cols="44" rows="5">
        <?=$page_data['keywords'];?></textarea><br /><br /><br /><label>МЕТА-
теги </label><br />
        <textarea name="description" cols="44"
rows="5"><?=$page_data['description'];?></textarea>

        <input type="submit" name="add" class="button" value="Сохранить"
/>
    </form>

    <? }
    if ($_GET['comand']=='new' & $_GET['save']==1)
    {
        $count = 0;
        $table = $_GET['t'];

        foreach($_POST as $ArrKey => $ArrStr)
        {
            $row[$count] = $ArrKey;
            $data[$count] = "".$_POST[$ArrKey]."";

```

```

$count++;
}

unset($row[$count]);
unset($data[$count]);
insert_data($table, $row, $data);
echo "Новая страница успешно добавлена!";
}
if ($_GET['comand']=='edit' & $_GET['save']==1)
{
$count = 0;
$table = $_GET['t'];
$id = $_GET['id'];

    foreach($_POST as $ArrKey => $ArrStr)
    {
        $row[$count] = $ArrKey;
        $data[$count] = $_POST[$ArrKey];
        $count++;
    }

update_data($table, $id, $row, $data);
echo "Страница успешно отредактирована!";
}

if (empty($_GET['comand']) & empty($_GET['id']) & empty($_GET['t']))
{
$data = get_page_admin();
?>

<td><input type="submit" style="cursor:pointer;"
onclick="location.href='content-page.php?comand=new'" class="login-
buttons" value="Добавить новую страницу." /></td><br><br>

<table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>
<td class="table_headlines">ID</td>
<td class="table_headlines">Название</td>
<td class="table_headlines">Ссылка</td>
<td class="table_headlines2" width="60">Действие</td>
</tr>

<?

```

```

foreach($data as $item):
?>
<tr>
    <td class="articles_type"><?=$item['id'];?></td>
    <td class="articles_topic"><a href="#"><?=$item['title'];?></a></td>

    <td class="articles_comment">
    <a href="../index.php?view=<?=$item['title_url'];?>"
target="_blank">Ссылка</a></td>
    <td class="articles_actions">
    <a href="content-page.php?comand=edit&t=page&id=<?=$item['id'];?>">
    </a>
    <a href="content-page.php?comand=delete&t=page&id=<?=$item['id'];?>"
    >
    </a></td>
</tr>
<?
endforeach;
}
?>
</table>
<?
//Управление новостями в административном разделе:
if ($_GET['comand']=='new' & empty($_GET['save']))
{
?>

<form action="content-news.php?comand=new&save=1&t=news"
method="post" enctype="multipart/form-data" id="add-edit" >

    <label>Имя новости </label><br /><textarea name="title" cols="100"
rows="1"/>
    </textarea><br /><br /><br />

    <label>URL-новости </label><br /><textarea name="title_url"
cols="100" rows="1">
    </textarea><br /><br /><br />

    <label>Контент </label><br /><textarea name="content" cols="55"
rows="11">
    </textarea><br /><br /><br />

    <label>Ключевые слова </label><br /><textarea name="keywords"
cols="44" rows="5">

```

```

</textarea><br /><br /><br />

<label>МЕТА- теги </label><br /><textarea name="description"
cols="44" rows="5">
</textarea><br /><br /><br />

<input type="submit" class="button" value="Сохранить" />

</form>

<?
}
if ($_GET['comand']=='delete')
{
$хid = $_GET['id'];
$хtable = $_GET['t'];
$хdelete_data = delete_data($хtable, $хid);
echo "Страница успешно удалена!";
}
if ($_GET['comand']=='edit' & empty($_GET['save']))
{
$хid = $_GET['id'];
$хtable = $_GET['t'];
$хnews_data = get_data_admin($хtable, $хid);
?>

<form action="content-
news.php?comand=edit&save=1&t=news&id=<?=$news_data[id];?>"
method="post" enctype="multipart/form-data" id="add-edit" >

    <label>Имя страницы </label><br />
    <textarea name="title" cols="100"
rows="1"/><?=$news_data['title'];?></textarea>

    <label>URL-страницы </label><br />
    <textarea name="title_url" cols="100"
rows="1"/><?=$news_data['title_url'];?></textarea>

    <label>Контент </label><br />
    <textarea name="content" cols="55"
rows="11"/><?=$news_data['content'];?></textarea>

    <label>Ключевые слова </label><br />
    <textarea name="keywords" cols="44"

```

```
rows="5"><?=$news_data['keywords'];?></textarea>
```

```
<label>МЕТА- теги </label><br />
```

```
<textarea name="description" cols="44"
```

```
rows="5"><?=$news_data['description'];?></textarea>
```

```
<input type="submit" name="add" class="button" value="Сохранить" />
</form>
```

```
<?
```

```
}
```

```
if ($_GET['comand']=='new' & $_GET['save']==1)
```

```
{
```

```
$count = 0;
```

```
$table = $_GET['t'];
```

```
foreach($_POST as $ArrKey => $ArrStr)
```

```
{
```

```
$row[$count] = $ArrKey;
```

```
$data[$count] = "".$_POST[$ArrKey]."";
```

```
$count++;
```

```
}
```

```
unset($row[$count]);
```

```
unset($data[$count]);
```

```
insert_data($table, $row, $data);
```

```
echo "Новая страница успешно добавлена!";
```

```
}
```

```
if ($_GET['comand']=='edit' & $_GET['save']==1)
```

```
{
```

```
$count = 0;
```

```
$table = $_GET['t'];
```

```
$id = $_GET['id'];
```

```
foreach($_POST as $ArrKey => $ArrStr)
```

```
{
```

```
$row[$count] = $ArrKey;
```

```
$data[$count] = $_POST[$ArrKey];
```

```
$count++;
```

```
}
```

```

update_data($table, $id, $row, $data);
echo "Страница успешно отредактирована!";
}

if (empty($_GET['comand']) & empty($_GET['id']) & empty($_GET['t']))
{
$data = get_news_admin();

?>

<td><input type="submit" style="cursor:pointer;"
onclick="location.href='content-news.php?comand=new'" class="login-
buttons" value="Добавить новую новость." /></td>

<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td class="table_headlines">ID</td>
<td class="table_headlines">Название</td>
<td class="table_headlines">Ссылка</td>
<td class="table_headlines2" width="60">Действие</td>
</tr>

<?
foreach($data as $item):
?>

<tr>
<td class="articles_type"><?=$item['id'];?></td>

<td class="articles_topic"><a href="#"><?=$item['title'];?></a></td>

<td class="articles_comment">
<a href=" ../index.php?view=<?=$item['title_url'];?>"
target="_blank">Ссылка</a></td>

<td class="articles_actions">
<a href="content-news.php?comand=edit&t=news&id=<?=$item['id'];?>">
</a>
<a href="content-
news.php?comand=delete&t=news&id=<?=$item['id'];?>" >
</a></td>
</tr>

```



```

<?
    endforeach;
}
?>
</table>

<?
    //Управление файлами в административной панели:
?>
<table width="100%" border="1">
    <tr>
        <td>Имя файла</td>
        <td>Описание</td>
        <td>Предпросмотр файла(URL)</td>
        <td>Удалить</td>
    </tr>
<?
    $dir = '../userfiles/';
    $dh = opendir($dir);
    while(false !== ($file = readdir($dh)))
    {
        $data_file_info = file_info($file);

        foreach($data_file_info as $item):
?>

        <tr>
            <td><?=$item['name'];?></td>
            <td><?=$item['description'];?></td>
            <td><? echo "<a href='\".$dir.\"".$file.\">\".$file.\"</a>"; ?></td>
            <td><a href="content-files.php?comand=delete&id=<?=$item['id'];?>" >
                </a>
        </td>
    </tr>
<?
    if($_GET['comand']=='delete')
    {
        delete_data(files,$item['id']);
        echo "Файл успешно удален!";
    }
?>

    </td>
</tr>

<?

```

```

        endforeach;
    }
    closedir($dh);
?>
</table>

<form action="content-files.php" method="post" enctype="multipart/form-
data">
<label>Имя файла </label><br /><textarea name="name" cols="100"
rows="1"/></textarea><br />
<label>Описание </label><br /><textarea name="description" cols="100"
rows="1"/></textarea><br />
<p><input type="file" name="userfile" id="file"></p>
<p><input type="submit" class="login-buttons" value="Загрузить новый
файл"></p>
</form>

```

```

<?
if(!empty($_FILES))
{
    $allowed_filetypes = array('.jpg','.gif','.bmp','.png');
    $max_filesize = 52428800;
    $upload_path = '../userfiles/';
    $filename = $_FILES['userfile']['name']; // В переменную $filename заносим
точное имя файла (включая расширение).
    $ext = substr($filename, strpos($filename,'.'), strlen($filename)-1); // В
переменную $ext заносим расширение загруженного файла.

```

```

    if(!in_array($ext,$allowed_filetypes)) die('Данный тип файла не
поддерживается.');
```

```

    // Теперь проверим размер загруженного файла и если он больше
максимально допустимого, то прерываем выполнение программы и выдаем
сообщение.

```

```

    if(filesize($_FILES['userfile']['tmp_name']) > $max_filesize) die('Файл
слишком большой.');
```

```

    // Проверяем, доступна ли на запись папка, определенная нами под
загрузку файлов (папка files). Если вдруг недоступна, то выдаем сообщение, что
на папку нужно поставить права доступа 777.

```

```

    if(!is_writable($upload_path)) die('Невозможно загрузить файл в папку.
Установите права доступа - 777.');
```

```

    // Загружаем файл в указанную папку и прописываем значения в базу

```

```

if(move_uploaded_file($_FILES['userfile']['tmp_name'],$upload_path .
$filename))
{
    $count = 0;
    $table = 'files';

    foreach($_POST as $ArrKey => $ArrStr)
    {
        $row[$count] = $ArrKey;
        $data[$count] = "".$_POST[$ArrKey]."";
        $count++;
    }

    unset($row[$count]);
    unset($data[$count]);
    insert_data($table, $row, $data);
    $last_id=mysql_insert_id(); // Получение ID только что добавленной
информации о файле
    mysql_query(" UPDATE $table SET userfile = '$filename' WHERE id =
'$last_id' ");
    echo 'Ваш файл успешно загружен <a href="" . $upload_path . $filename .
"">смотреть</a>';
}
else
{
    echo 'При загрузке возникли ошибки. Попробуйте ещё раз!';
}
?>

```

Додаток Б.
Презентація дипломного проекту

Інформаційна комп'ютерна
система управління вмістом
веб-сайту

КЕРІВНИК ДИПЛОМНОГО ПРОЕКТУ: ДОЦЕНТ ЩЕРБАКОВА
МАРИНА ЄВГЕНІВНА

СТУДЕНТ: КУРКЧИ АНТОН ПАВЛОВИЧ

1

Актуальність теми і мета проекту

- Більше 78% web-розробників у світі не користуються CMS при управлінні сайтом, тим самим вони ускладнюють собі життя и багато часу в пустоту, на непотрібну роботу.
- Розробити систему управління вмістом веб-сайту, яка полегшить життя розробникам, та людям, які не володіють навичками програмування.

2

ЩО ТАКЕ CMS?

Система управління контентом (Content Management System - CMS) - це автоматизований редакторський комплекс, що дозволяє управляти змістом і структурою Інтернет-ресурсу в режимі online. На практиці це означає, що будь-який, навіть незнайомий з інформаційними технологіями користувач, може абсолютно самостійно управляти вмістом свого сайту, не вдаючись при цьому до послуг технічного персоналу.

1

Основними вимога до розробленої CMS були:

- Редагувати існуючі сторінки і створювати нові;
- Редагувати інформаційне наповнення всіх сторінок сайту;
- Мати онлайн файл-менеджер;
- Бути інтегрованою з базою даних.

4

Вибір платформи для реалізації проекту

Мовний:

- HTML 5.0;
- PHP версії 5 і вище;
- MySQL.

Програмний:

- Adobe Photoshop CC;
- Денвер (PHP, MYSQL, Apache);
- Notepad ++;
- CKeditor 4.6.2;
- Microsoft WEB-Matrix 3.0.

5

Авторизація в CMS

При зверненні в адресному рядку браузера до адреси:

`http://www.імя_сайта.імя_домена/admin/` або `http://www.імя_сайта.імя_домена/admin/index.php` користувач

потрапляє на сторінку

авторизації.

6

Управління розділами(сторінками)

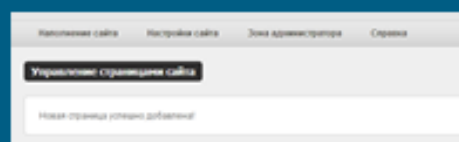
На сторінці "Управління сторінками сайту", системи адміністрування, виводиться список існуючих розділів (сторінок), доступних для виведення в головне меню.

ID	Название	Ссылка	Действие
1	Главная	Ссылка	✓ ✗
2	Контакты	Ссылка	✓ ✗
35	Видео	Ссылка	✓ ✗
34	Новости	Ссылка	✓ ✗
36	Служба	Ссылка	✓ ✗

7

Створення нового розділу(сторінки)

При натисканні кнопки зберегти, введені дані заносяться в базу даних; в разі успішного додавання сторінки виводиться повідомлення про успішно виконану операцію



8

База даних CMS

У базі даних, для роботи системи, використовується п'ять таблиць:

- Admin - таблиця для зберігання користувачів і їх паролів;
- Page - таблиця для зберігання сторінок сайту;
- News - таблиць для зберігання новин сайту;
- Files - таблиця для зберігання даних про завантажені файли;
- Log - таблиця, для зберігання інформації про спроби входу в адміністративну частину сайту: вдалих (1) і невдалих (0).

9

Висновок

У результаті було визначено, що таке CMS, були виконані основні вимоги до системи, та був розроблений програмний продукт, який представляє собою набір скриптів, за допомогою яких відбувається управління веб-контентом і призначений для швидкого створення простих сайтів за короткий термін.