

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

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
« _____ » _____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

_____ Методи побудови сайтів ведення блогів на базі технологій PHP-MySQL _____

Освітньо-кваліфікаційний рівень “Магістр”
Спеціальність 8.05010101 “Інформаційні управляючі системи та технології”

Науковий керівник роботи:

(підпис)

Щербаков Є.В.

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Я.О.Критська

(ініціали, прізвище)

Студент:

(підпис)

Роза Аділсон

(ініціали, прізвище)

Група:

_____ ІУС-16дм _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень Магістр
Напрямок підготовки _____
(шифр і назва)
Спеціальність 8.05010101 “Інформаційні управляючі системи та технології”
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____
I.C. Скарга-Бандурова
« _____ » _____ 20__ р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Роза Аділсон

(прізвище, ім'я, по батькові)

1. Тема роботи Методи побудови сайтів ведення блогів на базі технологій PHP-MySQL

керівник проекту (роботи) доц. Щербаков Є.В.
(прізвище, м. 'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від « _____ » _____ 2017 р. № _____

2. Строк подання студентом роботи 24.01.2018

3. Вихідні дані до роботи Матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Технології PHP-MYSQL для побудови сайтів

2. Аналіз технологій створення сайтів і блогів

3. Розробка сайтів і блогів на основі PHP-MYSQL

4. Охорона праці та безпека в надзвичайних ситуаціях

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
Основна частина	Щербаков Є.В.		
Охорони праці та безпека в надзвичайних ситуаціях	Критська Я. О.		

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	Отримання та аналіз завдання	05.09.2017–30.09.2017	
2	Пошук літературних джерел	01.10.2017–15.10.2017	
3	Вивчення літературних джерел	16.10.2017–30.10.2017	
4	Ознайомлення з предметним середовищем	01.11.2017–30.11.2017	
5	Розробка створення веб-сайту і блогу	01.12.2017–12.12.2017	
6	Оформлення пояснювальної записки	14.12.2017–27.12.2017	
7	Оформлення презентації	28.12.2017–05.01.2018	
8	Розробка заходів з охорони праці	06.01.2018–09.01.2018	
9	Представлення результатів науковому керівнику	10.01.2018–12.01.2018	
10	Підготовка та подання магістерської роботи до захисту	13.01.2018–26.01.2018	

Студент

_____ (підпис)

Роза Аділон

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Щербаков Є.В.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Роза Аділсон. Методи побудови сайтів ведення блогів на базі технологій PHP-MySQL.

Метою даної магістерської атестаційної роботи є дослідження та практичне використання методів побудови сайтів ведення блогів на базі технологій PHP-MySQL. В ході виконання роботи були проаналізовані сучасні веб-технології та був створений сайт-блог торгово-сервісного центру «Шангир». Розроблений сайт орієнтований на залучення нових і обслуговування старих клієнтів сервісного центру. При розробці сайту була використана система управління контентом Drupal.

КЛЮЧОВІ СЛОВА: ВЕБ-САЙТ, БЛОГ, СИСТЕМА УПРАВЛІННЯ КОНТЕНТОМ, МОВА ПРОГРАМУВАННЯ, БАЗА ДАНИХ, СИСТЕМА КЕРУВАННЯ БАЗАМИ ДАНИХ, HTML, CSS, JAVASCRIPT, PHP, MYSQL.

АННОТАЦИЯ

Роза Адилсон. Методы построения сайтов ведения блогов на базе технологий PHP-MySQL.

Целью данной магистерской аттестационной работы является исследование и практическое использование методов построения сайтов ведения блогов на базе технологий PHP-MySQL. В ходе выполнения работы были проанализированы современные веб-технологии и был создан сайт-блог торгово-сервисного центра «Шангир». Разработанный сайт ориентирован на привлечение новых и обслуживание старых клиентов сервисного центра. При разработке сайта была использована система управления контентом Drupal.

КЛЮЧЕВЫЕ СЛОВА: ВЕБ-САЙТ, БЛОГ, СИСТЕМА УПРАВЛЕНИЯ КОНТЕНТОМ, ЯЗЫК ПРОГРАММИРОВАНИЯ, БАЗА ДАННЫХ, СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ, HTML, CSS, JAVASCRIPT, PHP, MYSQL.

SUMMARY

Adilson Ribeiro. Methods of construction blogging sites based on PHP-MySQL technologies.

The purpose of this master's attestation work is to study and use the methods of building blogging sites on the basis of PHP-MySQL technologies. In the course of the work, modern web technologies were analyzed and a website-blog of the trade and service center "Shangir" was created. The developed site is focused on attracting new and servicing old customers of the service center. When developing the site, the content management system Drupal was used.

KEYWORDS: WEB-SITE, BLOG, CONTENT MANAGEMENT SYSTEM, PROGRAMMING LANGUAGE, DATABASE, DATABASE MANAGEMENT SYSTEM, HTML, CSS, JAVASCRIPT, PHP, MYSQL.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
1 ТЕХНОЛОГІЇ PHP-MYSQL ДЛЯ ПОБУДОВИ САЙТІВ	9
1.1 Мова програмування PHP.....	10
1.1.3 Можливості PHP.....	10
1.1.4 Чому потрібно вибирати PHP?.....	10
1.1.5 Переваги PHP	11
1.2 Система керування базами даних MySQL	12
1.2.1 Можливості MySQL	12
1.2.2 Чому потрібно вибирати MySQL?	13
1.2.3 Переваги MySQL	14
1.4 Сайти та блоги в мережі Інтернет.....	14
1.4.1 Визначення веб-сайту	14
1.4.2 Типи сайтів за їх призначенням	15
1.4.3 Соціальні мережі	19
1.4.4 Інструменти розробки сайтів і блогів.....	20
1.5 Системи управління контентом CMS.....	27
1.6 Постановка мети і завдань магістерської роботи.....	29
2 АНАЛІЗ ТЕХНОЛОГІЙ СТВОРЕННЯ САЙТІВ І БЛОГІВ.....	30
2.1 Мови розробки веб-сторінок	30
2.1.1 Мова HTML.....	30
2.1.2 CSS - каскадні таблиці стилів.....	31
2.1.3 Мова програмування JavaScript	34
2.1.4 Серверні мови програмування	34
2.2 Огляд і порівняльний аналіз базових технологій створення сайтів	38
2.2.1 Порівняння технологій PHP і ASP.NET	38
2.2.2 Аналіз та вибір веб-сервера	39
2.2.3 Аналіз систем керування базами даних (СКБД)	40
2.2.4 Порівняльний аналіз СКБД MySQL та ORACLE.....	42
2.3 Аналіз переваг і недоліків систем управління контентом (CMS).....	43
2.3.1 Класифікація систем управління контентом	45
2.3.2 Аналіз використання систем управління контентом	46
2.4 Переваги та недоліки вибору технологій PHP-MYSQL	48
3 РОЗРОБКА САЙТІВ І БЛОГІВ НА ОСНОВІ PHP-MySQL.....	52
3.1 Розробка сайтів з використанням технологій PHP-MySQL	52
3.1.1 Складові частини сайту, розроблюваного з використанням PHP-MySQL	52
3.1.2 Проектування, визначення та наповнення бази даних сайту	53
3.1.3 Створення файлу core.php - ядра сайту	54
3.1.4 Створення файлу index.php - головного контролера сайту	55
3.1.5 Створення компонентів виведення сторінок	56
3.1.6 Створення шаблону дизайну всього сайту	58

3.2 Розробка блогів з використанням технологій PHP-MySQL	59
3.2.1 Створення бази даних блогу	59
3.2.2 Створення контролера статей	59
3.2.6 Відображення списку статей	60
3.2.3 Додавання нових статей	61
3.3 Створення сайту-блогу для ТСЦ «Шангир»	64
3.3.1. Мета і завдання створення сайту-блогу для ТСЦ «Шангир»	64
3.3.2 Підготовка системи управління контентом Drupal для створення веб-сайту	64
3.3.3 Дизайн веб-сайту	67
3.3.4. Створення сторінок веб-сайту	68
3.3.5 Використані модулі CMS Drupal	71
3.3.6 Використані сторонні модулі	72
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ	74
4.1 Вступ	74
4.2 Правила охорони прці пі час експлуатації електронно- обчислювальних машин	74
4.3 Пожежна безпека	84
4.3 Влияние компьютеров на окружающую среду	85
4.4 Висновки по розділу	86
ВИСНОВКИ	87
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	88
ДОДАТОК А.	91
Початковий код шаблону «Corporate Clean»	91
ДОДАТОК Б.	96
Код «JivoSite», який вставляється в тіло документа html	96
ДОДАТОК В.	97
Перелік графічних матеріалів магістерської роботи	97

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

PHP	–	Hypertext Preprocessor
SQL	–	Structured Query Language
БД	–	база даних
СКБД	–	система керування базами даних
WWW	–	World Wide Web
HTML	–	Hypertext Markup Language
CSS	–	Cascading Style Sheets
ПЗ	–	програмне забезпечення
CMS	–	система управління контентом
ТСЦ	–	торгово-сервісний центр

ВСТУП

Веб-технології [1] повністю перевернули наше розуміння інформації та обчислювальної техніки в цілому. Зрештою виявилось, що традиційні параметри обчислювальних систем, такі як пропускна здатність, ємності пристроїв зберігання, не враховували головне, а саме, візуальну взаємодію з людиною, тобто системний інтерфейс.

Розвиток Інтернет-технологій призвело до появи нової гілки в Інтернеті. Сайти почали реєструватися під тисячами доменів, з'явилися цілі портали, де люди з усього світу можуть спілкуватися, отримувати відповіді на будь-які питання, здійснювати різноманітні операції в Інтернеті.

На даний момент важко уявити навчальний заклад, велику організацію, успішний бізнес, що не мають виходу в мережу і своєї веб-сторінки. Одночасно зростає і число "приватних осіб", що використовують Інтернет для своїх потреб - спілкування, пошук інформації [2]. Процес цей стрімко розвивається, використання Інтернету набагато збільшує інформативні можливості людини. А цей прогрес буде продовжуватися і сприяти розвитку молодого покоління, яке навчається в технічних вузах.

Створивши свій сайт або блог в Інтернеті, кожен з нас може зробити свій внесок в розвиток даної технології.

Для реалізації цього необхідно вирішити такі задачі:

- ознайомитись з використанням в веб-розробці сучасних Інтернет-технологій;
- дослідити програмне забезпечення для проектування, розробки і експлуатації сайтів та блогів;
- проаналізувати стратегії розробки та дизайну сайтів і блогів.
- проаналізувати методи і способи подання на веб-сторінках різних видів інформації;
- в відповідності до результатів, отриманих в результаті проведених досліджень, розробити сайт-блог для торгово-сервісного центру (ТСЦ) «Шангир».

Мета дипломної роботи. Метою дипломної роботи є дослідження та практичне використання методів побудови сайтів ведення блогів на базі технологій PHP-MySQL.

Тема є актуальною, так як за прогнозами на даний момент, більше 70% сімей використовують мережу Інтернет щодня. 86% Інтернет-користувачів з країн, що розвиваються, використовують Інтернет для зв'язку з друзями і членами сім'ї. Користувачі воліють використовувати Інтернет для доступу за необхідною інформацією: медичною, політичною, за інтересами і ін.).

При створенні сайту, потрібно зважити всі плюси і мінуси. Визначитися з метою, призначенням сайту, збалансувати його функціональне навантаження.

Об'єкт дослідження. Розробка веб-сайтів, блогів та веб-додатків.

Предмет дослідження. Методи побудови сайтів ведення блогів з використанням технологій PHP-MySQL.

1 ТЕХНОЛОГІЇ PHP-MYSQL ДЛЯ ПОБУДОВИ САЙТІВ

Кожна людина, яка більш-менш розуміється в розробці програм, скаже, що мова програмування PHP та СКБД MySQL є на сьогоднішній день є дуже важливими у всесвітній павутині, так як з їх використанням побудована значна частина програмного забезпечення Інтернет, в тому числі, системи управління контентом (Content management system - CMS).

Кожна CMS являє собою програмну основу, забезпечену призначеним для користувача інтерфейсом. Інструментарій CMS багато в чому полегшує процес адміністрування сайту. Але тоді навіщо займатися вивченням PHP з нуля? Адже можна просто зайти в адміністративну частину ресурсу і зробити всі маніпуляції за допомогою вбудованих в неї засобів [3, 7].

Почнемо з того, що не у багатьох вистачає терпіння, часу або сил займатися вивченням PHP і MySQL. До цього людина повинна прийти сама. Але той, хто зважиться опанувати ці дисципліни, отримає наступне.

Затребувану професію - на ринку праці тямущі програмісти завжди були нарозхват. Але навіть недосвідчені новачки з хорошими теоретичними знаннями PHP і невеликим досвідом в веб-програмуванні зможуть знайти роботу з гідною оплатою.

Віддалену роботу - для того щоб займатися програмуванням не обов'язково рано вставати, добиратися на роботу і дотримуватися дресскоду. Все це можна залишити за порогом, і створювати програмний код, сидячи вдома в кімнатних тапочках.

Економію грошей - якщо ви є власником сайту, то звичайно знаєте, що послуги професійних веб-програмістів коштують дорого. А для початківця свою справу в Інтернеті дорога кожна копійка. Самостійне навчання MySQL і PHP дозволить заощадити кошти і пустити їх в потрібне русло [10].

Як уже зазначалося, більшість сучасних движків CMS працюють на основі PHP. Хоча їх відмовостійкість і знаходиться на високому рівні, але вплив людського фактора ніхто не відміняв. Тому код CMS іноді поводить себе не так, як задумав веб-програміст.

І щоб виправити цю помилку, потрібно шукати фахівця, платити йому великі гроші. А потім виявляється, що вся заковика була лише в одному рядку коду, яку можна було легко виправити навіть зі знанням лише базових основ PHP.

Цікаве хобі - веб-програмування це не тільки засіб заробляння грошей або їх економії. Це ще й захоплююче заняття, яке поєднує в собі як технічну роботу, так і творчу складову.

1.1 Мова програмування PHP

Мова PHP [11] відноситься до скриптових мов програмування. Найчастіше він застосовується для створення веб-додатків і веб-сервісів. Дана мова підтримується більшістю хостингів в Інтернеті. На його основі написана велика частина динамічних сайтів.

У динамічному сайті на відміну від статичного, створеного лише на html, вміст всіх сторінок формується у відповідь на користувальницький запит. Структура ресурсу формується на основі файлів шаблону, а вміст (найчастіше) завантажується з баз даних.

1.1.3 Можливості PHP

На PHP можна робити все, що можна робити за допомогою CGI-програм: обробляти дані з форм, генерувати динамічні сторінки, одержувати і посилати куки (cookies). Крім цього в PHP включена підтримка багатьох баз даних (табл 1.1), що робить написання веб-додатків з використанням БД доволі простим.

Таблиця 1.1- Неповний перелік підтримуваних в PHP баз даних

Adabas D	InterBase	Solid
dBase	mSQL	Sybase
Empress	MySQL	Velocis
FilePro	Oracle	Unix dbm
Informix	PostgreSQL	

Додатково до всього PHP підтримує протоколи IMAP, SNMP, NNTP, POP3 і, природно, HTTP, а також має можливість працювати з сокетом (sockets) і спілкуватися по інших протоколах.

1.1.4 Чому потрібно вибрати PHP?

Розробникам веб-додатків немає необхідності говорити, що веб-сторінки - це не тільки картинка і текст. Гідний уваги сайт повинен підтримувати досить високий рівень інтерактивності з користувачем: пошук інформації, продаж продуктів, конференції і т. п.

Традиційно все це реалізувалося CGI-скриптами, написаними на Perl. Але CGI-

скрипти дуже погано масштабуються. Кожен новий виклик CGI, вимагає від ядра породження нового процесу, а це займає процесорний час і витрачає оперативну пам'ять.

PHP пропонує інший варіант - він працює як частина веб-сервера, і цим самим схожий на ASP від Microsoft.

Синтаксис PHP дуже схожий на синтаксис C або Perl. Люди, знайомі з програмуванням, дуже швидко починають писати програми на PHP. У цій мові немає строгої типізації даних і немає необхідності в діях по виділенню / звільненню пам'яті.

Програми, написані на PHP, легко прочитати. PHP - код легко візуально прочитати і зрозуміти, на відміну від Perl-програм [12].

1.1.5 Переваги PHP

Система програмування на базі мови PHP, включаючи бібліотеки та інструментальні засоби розробки програм, має такі переваги:

є вільним програмним забезпеченням, поширюваним під особливою ліцензією (PHP license);

- легка в освоєнні на всіх етапах;
- підтримується великою спільнотою користувачів і розробників;
- має розвинену підтримку баз даних;
- є величезна кількість бібліотек і розширень мови;
- може використовуватися в ізолюваному середовищі;
- пропонує нативні засоби організації веб-сесій, програмний інтерфейс розширень;
- є досить повною заміною пропрієтарного середовища ASP (Active Server Pages) від Microsoft;
- може бути розгорнута майже на будь-якому сервері;
- портована під велику кількість апаратних платформ і операційних систем.

Недоліками системи програмування на PHP є:

- не підходить для створення десктопних додатків або системних компонентів;
- має слабкі засоби для роботи з виключеннями;
- глобальні параметри конфігурації впливають на базовий синтаксис мови, що ускладнює настройку сервера і розгортання додатків;
- об'єкти передаються за значенням, що бентежить багатьох програмістів, які звикли до передачі об'єктів по посиланню, як це робиться в більшості інших мов;

- веб-додатки, написані на PHP, часто мають проблеми з безпекою.

1.2 Система керування базами даних MySQL

Система керування базами даних (СКБД) MySQL - компактний багатопотоковий сервер баз даних. MySQL характеризується великою швидкістю, стійкістю і легкістю у використанні.

MySQL був розроблений компанією ТсХ для швидкої обробки дуже великих баз даних. Компанія стверджує, що використовує MySQL з 1996 року на сервері з більш ніж 40 БД, які містять 10.000 таблиць, з яких близько 500 мають більше 7 мільйонів рядків.

MySQL є ідеальним рішенням для малих і середніх додатків. Початкові тексти сервера компілюються на багато платформ [11, 12]. Найбільш повно можливості сервера проявляються на Unix-серверах, де є підтримка багатопотоковості, що дає значний приріст продуктивності.

На поточний момент MySQL усе ще в стадії розробки, хоча версії 3.22 і більш пізні повністю працездатні.

MySQL-сервер є безкоштовним для некомерційного використання. Для комерційного використання необхідно придбати ліцензію, вартість якої становить 190 EUR.

1.2.1 Можливості MySQL

MySQL підтримує мову запитів SQL у стандарті ANSI 92, і крім цього має багато розширень до цього стандарту, яких немає в жодній іншій СКБД.

Короткий перелік можливостей MySQL:

- Підтримується необмежена кількість користувачів, що одночасно працюють з базою даних.
- Кількість рядків в таблицях може досягати 50 млн.
- Швидке виконання команд. Можливо MySQL найшвидший сервер з існуючих серверів.
- Проста і ефективна система безпеки.

MySQL дійсно дуже швидкий сервер, але для досягнення цього розроблювачам довелося пожертвувати деякими вимогами до реляційних СКБД. В MySQL відсутні:

- Підтримка вкладених запитів, типу `SELECT * FROM table1 WHERE id IN (SELECT id FROM table2)`.

- Не реалізована підтримка транзакцій. Натомість пропонується використовувати LOCK / UNLOCK TABLE.
- Немає підтримки тригерів і збережених процедур.

За словами розробників саме ці пункти дали можливість досягти високої швидкодії. Їх реалізація істотно б знизила швидкість сервера. Ці можливості не є критичними при створенні веб-додатків, що в поєднанні з високою швидкістю і малою ціною дозволило серверу набути велику популярність.

1.2.2 Чому потрібно вибрати MySQL?

MySQL є системою «клієнт-сервер». Це означає, що MySQL складається з одного SQL-сервера, який підтримує різні функції (власне, він і виконує всю роботу з базами даних), кількох різних програм-клієнтів (вони забезпечують лише інтерфейс між користувачем і сервером), адміністративних інструментальних засобів і декількох інтерфейсів програмування. Клієнти взаємодіють з сервером, використовуючи власний мережевий протокол (рис 1.2).

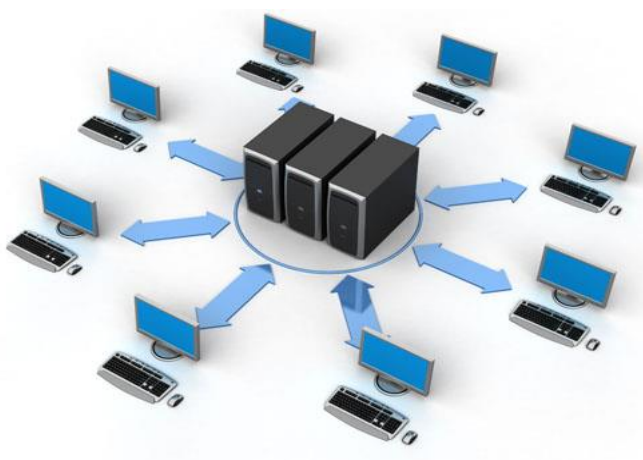


Рисунок 1. 2 – Схема роботи з сервером

Така схема дозволяє великому числу користувачів одночасно працювати з сервером. В тому числі, спільно обробляти одні й ті ж дані, не заважаючи один одному. Така взаємодія буде детально розглянута в наступних розділах.

Зазвичай клієнт і сервер використовуються на різних комп'ютерах, але припустимо варіант установки обох частин пакета на одному комп'ютері. Навіть в рамках єдиного комп'ютера все одно застосовується мережу для взаємодії клієнта і сервера. До речі, слід зазначити, що клієнт і сервер можуть працювати під різними операційними системами, це нічому не заважає. MySQL вже доступний більш ніж під десятком операційних систем.

1.2.3 Переваги MySQL

Використання СКБД MySQL дає розробнику такі переваги:

- Легкість установки і наявність візуальних засобів управління.
- Підтримка, прямо або побічно, великого SQL-функціоналу.
- Наявність вбудованих засобів безпеки.
- Масштабованість і робота з великими обсягами даних.
- Ігнорування певних стандартів збільшує швидкість роботи і продуктивність MySQL.

Недоліки MySQL:

- MySQL далеко не універсальна і володіє певними обмеженнями функціоналу, які «подобються» далеко не всіх програм.
- MySQL не так надійна, як хотілося б, у порівнянні з іншими СКБД.
- Незважаючи на те, що технічно MySQL являє собою програмне забезпечення з відкритим кодом, процес її розробки фактично припинений.

1.4 Сайти та блоги в мережі Інтернет

Інформація, доступна користувачам Інтернету, розташовується на веб-серверах, які використовують спеціальне програмне забезпечення. Велика частина цієї інформації організована в вигляді веб-сайтів, кожен сайт має свою адресу в Інтернеті [44].

1.4.1 Визначення веб-сайту

Веб-сайт - це дані та програми, представлені в певному вигляді, які розміщуються на веб-сервері і мають одну адресу. Перегляд веб-сайтів здійснюється з використанням спеціальних програм, які називаються браузером. [1, 2]: Залежно від адреси сайту, який ми введемо в рядку "Адреса", браузер завантажить відповідну інформацію.

Веб-сайт складається з пов'язаних між собою HTML-сторінок. Веб-сторінка являє собою текстовий файл з розширенням .htm, який містить текстову інформацію і HTML-теги, що визначають в якому вигляді ця інформація буде подаватися у вікні браузера. Вся графічна, аудіо- та відео інформація безпосередньо в веб-сторінку не входить і являє собою окремі файли з розширеннями .gif (графіка), .jpg (графіка), .mid (звук), .mp3 (звук), .avi (відео) . У HTML-кодї сторінки містяться тільки вказівки на такі файли.

Кожна сторінка веб-сайту також має свою веб-адресу, яка складається з адреси сайту і імені файлу, відповідного даній сторінці. З цього випливає, що веб-сайт - це

інформаційний ресурс, що складається з пов'язаних між собою гіпертекстових документів, розміщених на веб-сервері, і має індивідуальну адресу. Подивитися веб-сайт може будь-яка людина, що має комп'ютер, підключений до Інтернету.

Комерційні ресурси - одна з найбільш поширених категорій сайтів. Призначення комерційних ресурсів полягає в продажі товарів Інтернет-користувачам. Комерційні сайти діляться на кілька підкатегорій. Основними з них є:

- сайт-вітрина;
- промо-сайти;
- Інтернет магазини;
- сайти-візитки.

1.4.2 Типи сайтів за їх призначенням

Залежно від мети створення можна виділити такі типи сайтів:

- некомерційні;
- комерційні.

Залежно від доступу до сайтів користувачами їх умовно можна розділити на:

- закриті, які недоступні для людей, що не входять в певне коло;
- напівзакриті, які будуть доступні тільки після процедури реєстрації;
- відкриті, які доступні будь-якому користувачеві.

Від того як оформлений і який функціонал у сайтів, можна виділити наступні типи сайтів:

- флеш-сайти;
- динамічні;
- статичні.

Залежно від кількості користувачів виділяють наступні типи сайтів:

- тематичні або вузькоспрямовані;
- Інтернет-портали;
- сайти-візитки, персональні сторінки.

За фізичним розташуванням сайти можна розділити на наступні типи:

- локальні;
- загальнодоступні.

Одними з найпопулярніших ресурсів в мережі Інтернет є інформаційні сайти, вони доносять до користувача абсолютно будь-яку інформацію. Сайти інформаційної спрямованості можуть бути одно-тематичними або багато-тематичними.

Одно-тематичні сайти - ресурси, у яких практично всі сторінки присвячені одній певній проблематиці або темі, в свою чергу багато-тематичні сайти захоплюють широке коло інформаційної спрямованості і можуть бути присвячені великій кількості тем і напрямків. Рівень відвідувань таких сайтів дуже великий, користуються ними як великі організації, так і прості користувачі.

1.4.2.1 Новинні сайти

Завдання новинних сайтів полягає в донесенні до користувача різних новин, які відбувалися, відбудуться або відбуваються в даний момент. При цьому, новинні сайти можуть бути присвячені одній певній тематиці, або розповідати Інтернет-спільноті про новини з різних життєвих сфер людини.

1.4.2.2 Блоги

У мережі Інтернет блоги, на даний момент, зустрічаються досить часто, при цьому рейтинг їх популярності стає дедалі більшим. Блоги, так само як і новинні сайти, призначені для того, щоб донести ту чи іншу інформацію Інтернет-спільноті, яку автор описує по-своєму.

Блоги можна розділити на дві категорії, кожна з яких виконує свою певну задачу. Блоги можна умовно розділити на:

- корпоративні;
- особисті.

Корпоративні блоги, є додатковим ресурсом, який створюється компаніями, підприємствами чи підприємцями для розкрутки бренду. Корпоративний блог часто є доповненням до комерційного або корпоративного сайту. На них власники поширюють своїм потенційним та існуючим клієнтам останні тенденції в певній галузі, новини і т. і.

Особисті блоги - це категорія сайтів, до якої належать ресурси, створені окремими користувачами. Метою даних ресурсів є донесення великому колу користувачів точку зору автора з певної теми. Дуже часто блоги створюють фрілансери, публікуючи на них інформацію щодо своєї сфери діяльності.

Створення такого ресурсу підвищує популярність і дає можливість бути більш помітним серед конкурентів, залучати нових клієнтів, ділитися різною інформацією з Інтернет-користувачами і багато іншого.

Залежно від того, як багато тем розглядається, блоги можна розділити умовно на дві категорії:

- вузько тематичні, на яких розглядається кілька тем, що мають тісний зв'язок один з одним;
- загально тематичні, на яких розглядаються абсолютно різні теми.

До другої групи належать сайти корпоративної спрямованості.

1.4.2.3 Корпоративні сайти

Корпоративні Інтернет-ресурси створюються підприємствами або компаніями для виконання одного з двох певних завдань: зміцнення зв'язку між філіями та відділеннями або підвищення іміджу в очах потенційних і існуючих клієнтів. Залежно від того, яку з вищезгаданих проблем повинен вирішувати корпоративний ресурс, прийнята така класифікація сайтів:

- іміджевий;
- інформаційний.

Функцією іміджевих сайтів є рекламна діяльність. На таких сайтах описується історія підприємства, компанії або торгової марки, надаються докладні поточні відомості, контактна інформація. Також на даних сайтах містяться відомості про вироблені і які розповсюджуються товари, послуги, що надаються. На них також можуть публікуватися різні новини, інформація про акції, що надаються знижки - все те, що характеризує компанію перед клієнтом .

Вони відрізняються оригінальним ексклюзивним дизайном, поєднанням нестандартних елементів оформлення і рішень.

Автоматизація та зміцнення зв'язку між відділами та філіями є основною функцією інформаційних корпоративних сайтів, а також обіг документів, управління персоналом і багато іншого. Рядовим користувачам Інтернету, як правило, дані сайти не надають відкритого доступу, так як вони призначені виключно для співробітників однієї організації.

1.4.2.4 Сайти-вітрини

Сайт-вітрина - вид сайту, основним призначенням якого є не реалізація товару, а лише його реклама. Найбільшого поширення сайти вітрини отримали серед компаній і підприємств, які займаються випуском товарів. З його допомогою неможливо здійснити покупку, так як на даному сайті тільки надається опис товару. Як правило на них вказуються місця, де можна купити зацікавлений товар.

1.4.2.5 Промо-сайти

Промо-сайти займаються рекламуванням певних послуг або товарів, але особливу увагу зупиняючи на їх перевагах. Структура таких сайтів досить проста, характерними особливостями є невеликий обсяг і нетривалі тривалість життя. Інформація на сторінках таких сайтів може бути різноманітною - від графічних матеріалів до контактних даних компанії. Промо-сайт є прекрасним рішенням масштабної рекламної компанії певного товару.

1.4.2.6 Сайти-візитки

Сайти-візитки є популярними серед молодих і початкових фірм, індивідуальних підприємців і призначені для того, щоб коротко представити інформацію про свого власника великому колу користувачів. Як правило вони складаються з 3-5 сторінок.

Сайти-візитки мають нестандартну структуру, яка ділить його на кілька розділів, в яких містяться дані про компанію, які продаються нею товари, послуги, що надаються, контактні дані. До переваг сайтів-візиток можна віднести:

- високу швидкість створення;
- змістовний і короткий виклад всієї інформації;
- залучення партнерів і клієнтів;
- низька вартість;
- зручність і простота в управлінні.

1.4.2.7 Сайти-портфоліо

За рахунок реклами робіт, виконуваних організацією, сайти-портфоліо призначені привертати увагу широкої цільової аудиторії, щоб клієнт, дивлячись на конкретний результат зміг зробити для себе певні висновки і вирішити, чи потрібно звертатися за послугами до даної компанії або індивідуальному підприємцю. Більш популярними вони є серед фотосалонів, веб-студій, окремих розробників сайтів.

Сайти-портфоліо і сайти-візитки сьогодні усе частіше створюються приватними особами, особливо фрілансерами, які намагаються привернути увага нових замовників і клієнтів, розрекламувавши свою роботу, надавши результат її виконання.

1.4.2.8 Інтернет-магазини

Основним напрямком діяльності Інтернет-магазинів є продавати різні категорії товарів в режимі онлайн. На таких сайтах, користувач може вибрати зацікавивший його товар, дізнатися про нього докладну інформацію і, в разі необхідності, здійснити

замовлення, натиснувши декілька кнопок.

Найчастіше на одному сайті пропонуються всі товари і послуги, які надає дане підприємство. Значно рідше створюються кілька ресурсів, на яких представлена одна певна категорія поширюваної продукції.

Інтернет-магазини створюються з метою розширити коло потенційних покупців і можливості свого сервісу.

До четвертої групи належать соціальні проекти. До соціальних проектів відносяться спеціалізовані соціальні мережі або мережі загальної тематики, сайти-спільноти, форуми і т.д.

1.4.2.9 Форуми

Форуми - спеціалізовані сайти, на яких користувачі глобальної мережі Інтернет можуть безперешкодно обговорювати різні новини, проблеми, життєві сфери суспільства і т. і. Як і інші категорії Інтернет-сайтів, форуми поділяються на дві підгрупи.

- вузько направлені, обговорюються одна або кілька проблем, тісно переплетених між собою;
- загально спрямовані, обговорюються проблеми, абсолютно різні за тематикою і часом зовсім не пов'язані один з одним.

1.4.3 Соціальні мережі

Кожна людина, знайома з всесвітньою павутиною, напевно чула що-небудь про соціальні мережі, а лєвова частка з них мають власні акаунти. Найбільші соціальні мережі вже встигли зібрати багатомільйонну аудиторію. Основне завдання таких веб-ресурсів - надати Інтернет-користувачам можливість знаходити один одного і вести між собою спілкування в режимі онлайн [2, 44].

Соціальні мережі діляться на дві групи:

- вузько спрямовані;
- загально спрямовані.

Вузько спрямовані соціальні мережі об'єднують в собі людей, які пов'язані між собою одним або кількома загальними інтересами (хобі, професія). Соціальні мережі загальної спрямованості збирають в одне коло для спілкування людей, які зайняті в абсолютно різних сферах і мають різні інтереси і хобі.

До п'ятої групи належать різні веб-сервіси. Найбільш поширеними є:

- каталоги - сайти, на яких містяться структуровані посилання, що ведуть на інші веб-ресурси, і їх короткий опис;

- поштові сервіси - надають можливість роботи з електронною поштою;
- пошукові системи - надають можливість шукати різну інформацію з урахуванням пошукових запитів;
- дошки оголошень - веб-ресурси, на яких користувачі мережі Інтернет розміщують оголошення різних тематик;
- сайти-хостинги - дані сайти дають можливість тимчасового зберігання файлів, деякі з них надають можливості перед завантаженням переглянути вміст файлу у вікні браузера, зображень.

1.4.4 Інструменти розробки сайтів і блогів

Типовий процес розробки сайту, складається з упорядкованої послідовності дій, які в підсумку призведуть до запуску сайту.

Якщо представити візуально етапи розробки веб-сайту, то це виглядає так, як представлено на (рис.1.3).



Рисунок 1. 3 - Етапи розробки веб-сайту

Більш детально, кожен з етапів розшифровується наступним чином:

- перший контакт - постановка мети і подальший її супровід;
- планування - як, ким, чим і в які терміни будуть проектуватися окремі частини сайту;
- контент - це вміст сайту, його інформаційне наповнення;
- дизайн - розробка візуального оформлення сайту;
- програмування – розробка програм сайту на мові PHP;
- запуск сайту в роботу.

1.4.4.1 Основи використання PHP та MySQL

Отже, для початку нам потрібно локальний сервер, для того щоб можна було запускати PHP-скрипти і підключатися до бази даних (БД). Якщо ще немає локального сервера, то спочатку слід його установити і налаштувати [12]: Якщо вже є цей сервер, то за допомогою компоненти PhpMyAdmin визваємо головну сторінку СКБД (рис. 1. 4), яка дозволяє управляти (додавати, видаляти, змінювати) записами в БД.

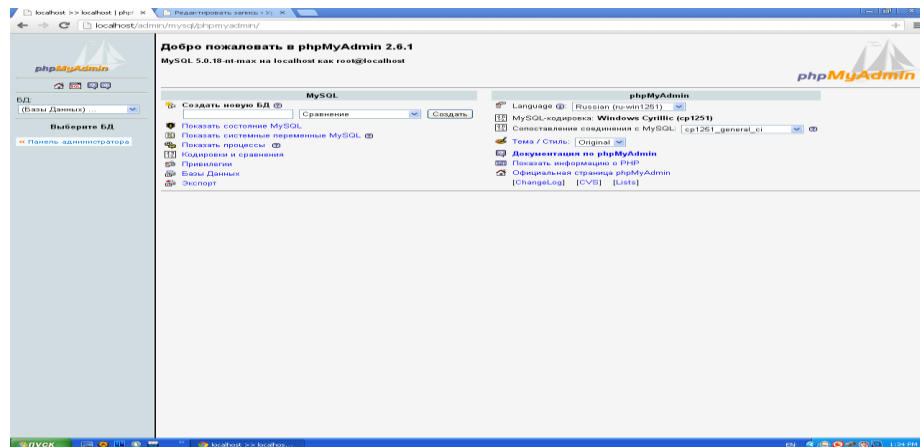


Рисунок 1. 4 – Головна сторінка СКБД

Звідси можна створити свою БД і зберегти в ній необхідні записи. Це дуже зручно при створенні сайтів, тому що інформація структурована і ви можете дуже швидко отримати будь-який запис з БД.

Для початку роботи, нам буде потрібно створити саму базу даних. У неї буде своє ім'я, за яким ми потім будемо її шукати. Загальна структура бази даних приведена на (рис. 1.5).

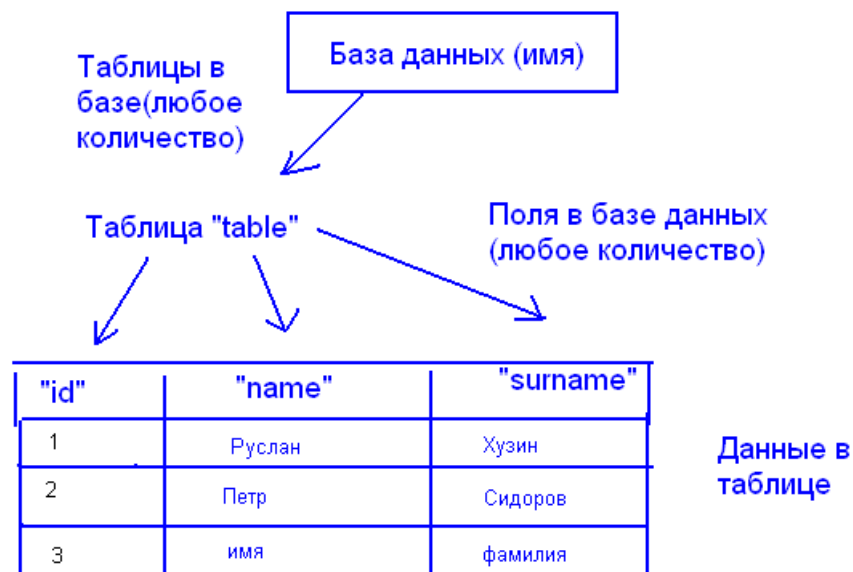


Рисунок 1.5 - Структура базы даних

Для того щоб створити БД, впишіть в поле введення під рядком «Створити нову БД» (на головній сторінці PhpMyAdmin) будь-яке ім'я (НЕ кирилиця!) Для майбутньої бази. Я створюю базу даних з ім'ям «База». Тепер, слідує схемі побудови БД, нам потрібно зробити таблицю. Для цього після створення бази з'явиться вікно, показане на (рис. 1.6).

Рисунок 1.6 – Вікно створення нової таблиці БД

Створюємо таблицю з ім'ям users і з трьома полями в ній. Далі PhpMyAdmin попросить нас задати структуру для кожного поля (рис. 1.7).

Поле	Тип	Длины/ Значения*	Сравнение	Атрибуты	Ноль	По умолчанию**	Дополнительно			
id	INT	5			not null		auto_increment	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
name	VARCHAR	100			not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
sumame	VARCHAR	100			not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Рисунок 1.7 – Вікно з атрибутами для кожного поля

Тут в стовпці «Поле» потрібно вказати ім'я, в «Тип» вказуємо тип даних які будуть там зберігатися (INT - це число, VARCHAR - рядок або невеликий текст). У «Додатково» вказуємо полю з ім'ям «Id» параметр «Auto_increament», що означає, воно буде збільшуватися кожного разу при вставці будь-яких даних, і задаємо йому параметр «Первинний ключ» (вибираємо відповідну радіо-кнопку). Це означає, що по значенні в одній колонці можна однозначно ідентифікувати поле. «Однозначно ідентифікувати поле» - значить навіть якщо ім'я та прізвище будуть однакові, стовпець «id» матиме різні значення (так як він автоматично збільшується на одиницю кожного разу). Тепер натискаємо «Зберегти». Тим самим ми створили таблицю users, в якій будемо зберігати імена та прізвища (id не потрібно вставляти, MySQL зробить все за нас).

Після цього вибираємо в меню «Вставити» (рис. 1. 8) і пишемо потрібні значення в поля таблиці бази даних.

Рисунок 1. 8 – Вікно для введення значень

Для видалення або редагування даних в таблиці треба перейти в меню «Огляд» і натиснути червоний хрестик навпроти запису таблиці (рис. 1. 9).

Рисунок 1. 9 - Вікно для видалення або редагування значень

Для редагування значень в таблиці треба натиснути на олівець і замінити необхідні поля.

1.4.4.2 Робота з базою даних в PHP

Отже, для початку потрібно підключитися до вже створеної БД. Для цього використовується ось такий код:

```
<?php
    $connect = mysql_connect ( "localhost", "root", "" );
```

Тут ми підключилися до бази даних, використовуючи функцію `mysql_connect()`, в

якій задано три аргументи:

- 1) сервер бази даних,
- 2) логін користувача БД,
- 3) пароль користувача БД.

У вищевикладеному коді, було виконане підключення до сервера localhost, використовуючи логін root, який не вимагає пароля (тому він не вказаний).

Після цього, потрібно задати, яка база даних буде використовуватись в подальшому коді PHP:

```
$db = mysql_select_db ( "baza");
// Тут ми вказуємо,
// яку БД будемо використовувати ("baza")
?>
```

Після підключення до бази, ми можемо проводити всі ті операції які виконували в СКБД phpmyadmin. Тобто вставку, видалення, зміна і отримання різної інформації.

Вставка записів в базу даних SQL на PHP виконується такими інструкціями:

```
$query = "INSERT INTO `таблиця` ( `вказуємо_поле`, `вказуємо_поле`) VALUES (
'будь-яке значення', 'будь-яке значення') " ;
$result = mysql_query ( $ query ) ;
if ( $result == true ) {
    echo "Успішно!" ;
} else {
    echo " Помилка ! <br>" . mysql_error () ;
}
```

Якщо все пройшло успішно, то mysql_query() поверне результат true, а інакше - false.

Полів можна вказувати хоч скільки, головне щоб всі вони існували в БД. Тобто, наприклад, існує таблиця в якій є поля «name», «surname» і «city». Для вставки в неї значень будемо використовувати такий код:

```
$query = "INSERT INTO `users` ( `name`, `surname`, `city`) VALUES ( 'Ruslan',
'Н uzin', 'Kokshetau') " ;
$result = mysql_query ( $ query ) ;
```

```

if ( $result == true ) {
    echo "Успішно!" ;
} else {
    echo " Помилка ! <br>" . mysql_error () ;
}

```

Видалення записів з бази даних на PHP виконується функцією SQL DELETE. Для цього буде приблизно ось такий код:

```

$query = "DELETE FROM `users` WHERE `name` = 'Ruslan'";
$result = mysql_query ( $query ) ;
if ( $result == true ) {
    echo "Успішно!" ;
} else {
    echo " Помилка ! <br>" . mysql_error () ;
}

```

Тобто ми видалили з таблиці users всі рядки, в яких в стовпчику name значення дорівнює рядку "Ruslan".

Також ми можемо вносити зміни в уже існуючі записи в таблиці. Наприклад, нам треба замінити значення стовпця name в рядку, де стовпчик surname має значення "Huzin". Для цього виконуємо такий код :

```

$query = "UPDATE `users` SET `name` = 'myname' WHERE `surname` = 'Huzin' " ;
$result = mysql_query ( $ query ) ;
if ( $result == true ) {
    echo "Успішно!" ;
} else {
    echo " Помилка ! <br>" . mysql_error () ;
}

```

Отримання значень з бази даних виконується наступним чином. Наприклад, потрібно дістати з таблиці users весь рядок, де стовпчик name дорівнює "Ruslan". Для цього потрібен трохи інший код, ніж в перших прикладах:

```

$query = "SELECT * FROM WHERE `name` = 'Ruslan' ";
$result = mysql_query ( $ query );
if ( $result == true ) {
    echo "Успішно!" ;
} else {
    echo " Помилка ! <br>" . mysql_error () ;
}
$data = mysql_fetch_array ( $result ) ;
/* Тепер в змінній $data зберігаються дані з таблиці */

```

Для того що звернутися до якогось елемента прочитаного запису, потрібно написати:

```
$data [ 'ім'я стовпця' ]
```

Тобто, щоб прочитати прізвище з отриманого запису (де ім'я було "Ruslan"), треба виконати:

```
echo $data[ 'surname' ] ;
```

Для вибірки відразу декількох записів з таблиці, потрібен цикл для їх виведення. Якщо, наприклад, записів зі значенням "Ruslan" в стовпці `name` буде декілька то можна написати наступний код:

```

$query = "SELECT * FROM WHERE `name` = 'Ruslan' ";
$result = mysql_query ( $query );
while( $data = mysql_fetch_array ( $result ))
{
    echo $data [ 'name' ]. "<br>". $data [ 'surname' ]. "<hr>";
}

```

Тепер на екран виведуться всі записи у яких значення в стовпці name дорівнює "Ruslan".

1.5 Системи управління контентом CMS

Для розробки сайтів часто застосовуються системи CMS (Content Management System - система управління контентом), які використовуються для забезпечення і організації спільного процесу створення, редагування і управління контентом (тобто вмістом). Головною метою таких систем є можливість збирати в єдине ціле і об'єднати на основі ролей і завдань все різноманітні джерела знань та інформації, доступні як всередині організації, так і за її межами, а також можливість забезпечення взаємодії співробітників, робочих груп та проектів з створеними ними базами знань, інформацією та даними так, щоб їх легко можна було знайти, прочитати і повторно використовувати звичним для користувача чином [3].

Існують різноманітні системи управління сайтом. Серед них зустрічаються платні і безкоштовні, побудовані за різними технологіями. Кожен сайт має панель управління, яка є тільки частиною всієї програми, але достатня для управління сайтом.

Термін контент-менеджер позначає рід професійної діяльності - редактор сайту.

Велика частина сучасних систем управління вмістом реалізується у вигляді візуального (WYSIWYG) редактора - програми, яка створює HTML-код зі спеціальної спрощеної розмітки, що дозволяє користувачеві простіше форматувати текст.

У наш час технологія CMS містить в собі більшість необхідних зв'язків, модулів і плагінів «прямо з коробки». Що б ви не хотіли зробити - Інтернет магазин, корпоративний сайт або блог - як правило, в будь-якої CMS це вже реалізовано. Їх є багато, як платних, так і безкоштовних, наприклад, Бітрікс, Joomla!, Shop-script, WordPress, OpenCart та інші.

Основні особливості CMS системи:

- надання інструментів для створення, редагування, видалення вмісту сайту;
- управління вмістом: зберігання, контроль версій, дотримання прав доступу і т. ін.;
- налаштування призначеного для користувача візуального інтерфейсу;
- подання інформації у вигляді, зручному для навігації і пошуку.

В системі управління контентом можуть перебувати найрізноманітніші дані: документи, відео, фотографії, контакти, наукові дані і т. д. Така система часто використовується для зберігання, управління, перегляду і публікації документації.

До появи CMS-систем, для того, щоб створити сайт, необхідно було знати масу технологій і мов веб-програмування, наприклад, таких як: HTML, CSS, PHP, а також володіти деякими навичками дизайнера і мати загальне уявлення про процес розробки веб-проекту [15].

Тепер будь-якій людині стало доступне створення Інтернет-сайтів, які по дизайну і функціональності нічим не поступаються продуктам професійних веб-студій.

Однак стартові набори таких CMS систем не завжди можуть відповідати всім поставленим вимогам. Якщо спочатку існують специфічні запити до функціоналу майбутнього сайту, наявність нестандартних рішень, краще відразу звернутися до хорошого фахівця-розробника, або до людини, яка розбирається в специфіці CMS-системи і яка допоможе довести стандартний функціонал до необхідних вимог. Нижче подаються короткі описи найбільш поширених CMS [7].

Joomla! (Джумла) - система, розроблена на PHP і використовує базу даних MySQL. Дана система підходить під будь-які завдання. Має зрозумілу для початківця-користувача панель адміністрування. Створено багато модулів, плагінів і розширень для Джумли, що дозволяють робити на її основі різні сайти: від простого сайту-візитки до Інтернет-магазину.

Wordpress (Вордпрес) - система розроблена на PHP, в якості сервера бази даних використовує MySQL. Переваги системи - це її простота адміністрування, підтримка RSS, багато розширень і доповнень. Система підійде для ведення блогу на сайті.

Бітрікс (Bitrix) - комерційна система, розроблена компанією 1С, орієнтована на корпоративні сайти, інформаційні та довідкові портали, соціальні мережі, Інтернет-магазини, сайти ЗМІ. Найпоширеніша комерційна система в російськомовному сегменті Інтернету. Підтримує наступні СКБД: MySQL, Oracle, MS SQL.

Drupal (Друпал) - дана CMS написана на мові PHP і використовує як сховище даних реляційні бази даних, такі як MySQL, PostgreSQL та інші. Ядро Drupal має дуже добре опрацьовану структуру, де передбачена можливість переробити всю систему під інші цілі, написавши додаткові модулі, що дозволяє з її допомогою створювати великі портали і специфічні веб-додатки. Це робить Drupal найбільш популярною системою управління контентом серед професійних розробників [9, 45].

1.6 Постановка мети і завдань магістерської роботи

Метою даної магістерської атестаційної роботи є дослідження та практичне використання методів побудови сайтів ведення блогів на базі технологій PHP-MySQL.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- ознайомитись з використанням в веб-розробці сучасних Інтернет-технологій;
- дослідити програмне забезпечення для проектування, розробки і експлуатації сайтів та блогів;
- проаналізувати стратегії розробки та дизайну сайтів і блогів.
- проаналізувати методи і способи подання на веб-сторінках різних видів інформації;
- в відповідності до результатів, отриманих в результаті проведених досліджень, розробити сайт-блог для торгово-сервісного центру (ТСЦ) «Шангир».

Розроблюваний сайт-блог повинен забезпечити для ТСЦ «Шангир»:

- підтримку старих і створення нових зв'язків на ринку;
- забезпечення цілодобової підтримки клієнтів і партнерів;
- створення сприятливої думки про ТСЦ «Шангир».

Для вирішення цих завдань розроблюваний сайт-блог повинен:

- надавати інформацію про ТСЦ «Шангир», його діяльність, режим роботи, нові надходження, новини та ін.;
- забезпечити можливість пошуку по змісту сайту;
- забезпечити зв'язок користувачів сайту з компанією.

2 АНАЛІЗ ТЕХНОЛОГІЙ СТВОРЕННЯ САЙТІВ І БЛОГІВ

2.1 Мови розробки веб-сторінок

2.1.1 Мова HTML

Термін HTML (Hypertext Markup Language) означає «мова гіпертекстової розмітки».

Перша версія мови HTML була розроблена в 1989 році Тімом Бернерс-Лі для популярного в минулому браузеру Mosaic. Можна зрозуміти чому в той час ні для мови, ні для браузера застосування не було, але часи змінюються. У 1993 році з'явився HTML +, і ця версія також залишилася практично непоміченою [14]. Широке використання гіпертексту дала версія 2.0, яка з'явилася в червні 1994 року, який був роком зростання популярності в світі World Wide Web. Зміни (більшість з них), які були включені у версію 2, використовуються і до цього дня.

У версії 3.0 HTML, яка з'явилася через рік, була можливість включення математичних символів (знак інтеграла, нескінченність, фракції і т. і.), які стали елементами мови. За цією версією був розроблений браузер Арена, але цей проект виявився глухим кутом і не набув поширення.

У 1996 році було оновлення HTML 3.2. Це було новаторське рішення, були введені фрейми, які стали тепер дуже популярними серед розробників веб-сторінок. Навіть зараз на основі цієї специфікації можна реалізувати цікаві дизайнерські рішення. [15]. Майже всі сучасні браузери підтримують версію 3.2.

На додаток до офіційної специфікації мови, яка постійно розвивається, виробники браузерів від W3C (W3 Консорціум) вводили їх власні елементи (розширення).

Потім деякі з цих пунктів після всесвітнього визнання появлялись в описі наступних версій мови. Інноваційне рішення - «фрейми» не були включені в специфікацію 3.2. Але браузери підтримували фрейми і багато книг по HTML містили опис без згадки про те, що це нестандартне рішення. Згодом, стандарт став де-факто. У версії 4 вони вже були включені в базову мову.

Навпаки елементи APPLELET і SCRIPT, як розширення мови HTML версії 3.2. не зіграли тієї ролі, на яку вони претендували спочатку. Тому що різні версії браузерів інтерпретували різні мовах програмування, такі як, наприклад, Java, JavaScript, Visual Basic (VBScript).

Офіційна специфікація HTML 4 (Dynamic HTML) з'явилася в 1997 році, в той час вже було ясно, що подальший розвиток гіпертексту буде забезпечуватись за допомогою

програмування сценаріїв. Це виявилось трохи більш ефективним, ніж вносити нові елементи мови [18].

2.1.2 CSS - каскадні таблиці стилів

Каскадні таблиці стилів CSS або (від англійського Cascading Style Sheets) - це досить пізня інновація в гіпертекстову розмітку. Якщо сам HTML був опублікований в 1989 році, таблиці стилів CSS - тільки в 1997 році.

Крім того, таблиця стилів формально не є частиною HTML. Це пов'язано з тим, що всесвітня мережа була створена вченими як засіб для обміну текстовими документами, і HTML була мовою, за допомогою якого писали ці документи. Для вчених, основний зміст документа був більш важливішим його зовнішнього вигляду.

Таким чином, перша версія HTML не включала в себе засоби для розміщення на веб-сторінці графіки, не кажучи вже про форматування тексту. Але час минав, і в Інтернеті з'явилися користувачі, які зажадали красивого дизайну веб-сторінок. Веб-дизайнери, в свою чергу, затребували, щоб розробники стандартних інструментів HTML зробили все, щоб полегшити роботу по дизайну.

Так що всі каскадні таблиці стилів в даний час стандартизуються специфікацією CSS3, яка повністю підтримує CSS1, а також CSS2 частково.

Каскадні таблиці стилів, або просто файли стилів (CSS - Cascading Style Sheets) - набір правил, що описують форматування різних фрагментів HTML коду і знаходиться окремо від нього. Правило форматування одного елемента HTML або ж групи елементів, або фрагменту HTML-коду, називається стиль. Таблиці стилів записуються на спеціальній мові, а CSS-правила зберігаються в файлах з розширенням .css, хоча вони можуть бути вбудовані в саму веб-сторінку [8, 15, 18].

В цілому стандарт CSS визначає три способи налаштування стилів для елементів сторінок:

- Зовнішні таблиці стилів. Стилі зберігаються в окремому файлі з розширенням .css, які додаються до веб-сторінки за допомогою спеціального тегу <link>.
- Внутрішні (або вбудовані) таблиці стилів. Стилі в цьому випадку мають той же формат, що і зовнішні, але в заголовку веб-сторінок поміщаються в спеціальний тег <style>.
- Лінійний (або інтегрований) стиль. Визначення стилю поміщається прямо в потрібний тег, використовуючи спеціальний атрибут стилю.

Спадкування стилів. Спадкування – це механізм, за допомогою якого стилі

застосовуються не тільки до вибраних елементів, але також до їх нащадків. Наприклад, якщо колір з правила:

```
h1 {color: gray;}
```

застосований до елемента h1:

```
<h1>Meerkat <em>Central</em></h1>
```

то цей колір буде застосований до всього тексту в h1 і до тексту, який поміщений в дочірні елементи цього h1. Якби значення властивостей не успадковувалися елементами - нащадками, текст em був би чорним, як передбачено в таблиці стилів браузер за замовчуванням, і цей текст довелося б фарбувати за допомогою окремого правила.

Спадкування добре працює зі списками і багатьма іншими елементами веб-сторінки.

Краще всього зрозуміти принцип роботи спадкування можна з деревовидного представлення документа. Якщо значення якоїсь властивості застосовується до елемента - вершини піддерева, то це значення передається вниз по дереву елементів - нащадків до тих пір, поки не залишиться нащадків, [21] які могли б спадкувати це значення. Значення властивості ніколи не передається вгору по ієрархії, тобто елемент ніколи не передає значення своїм предкам.

При розгляді принципу спадкування треба мати на увазі, що деякі властивості з міркувань здорового глузду не успадковуються. Це такі властивості, як margin, width, border і їм подібні.

Каскадування стилів. Застосування стильових правил починається з елемента – вершини дерева елементів сторінки і відповідно до правил спадкування розповсюджується до нижніх елементів дерева. Для кожного з елементів, представлення яких повторно визначається таблицями стилів, браузер або об'єднує визначення стилів з різних таблиць стилів або перекриває їх визначення. Природно, що при стилізації елементів два або більше оголошень, отриманих з різних джерел, можуть конфліктувати.

Для вирішення таких конфліктів для кожного елемента накопичуються всі призначені для нього і успадковані конфліктуючі стильові оголошення, з яких вибирається одне найбільш важливе (пріоритетне) в наступній послідовності:

- Стилі користувача. Якщо користувач надав свою таблицю стилів, стилі з цієї таблиці мають перевагу над правилами з інших джерел.

- Лінійні стилі. Якщо є стилі, задані за допомогою атрибута `style` елемента вони завжди застосовуються незалежно від інших чинників. Цим пояснюється рекомендація: ніколи не можна використовувати лінійні стилі у фінальній редакції HTML-сторінки.
- Тип пристрою виведення (`media`). Очевидно, якщо тип пристрою виведення, заданий для таблиці стилів атрибутом `media` або директивою `@media`, не збігається з пристроєм, на якому в даний момент відображається елемент, стиль не використовується.
- Важливе оголошення стилю. Включення специфікатора `!important` в оголошенні стилю безпосередньо перед комою дає перевагу цьому стилю над всіма іншими. Якщо в конфлікт вступають два оголошення стилю з цим специфікатором, перевага віддається стилю з таблиці стилів користувача.
- Специфічність. Чим вище контекстуальна специфічність правила, тим вище його пріоритет каскадування. Чим більше ідентифікаторів, класів і/або тегів входить в селектор правила, тим вище його специфічність. При обчисленні специфічності правила кожен ідентифікатор в селекторі додає 100, кожен клас – 10, кожен тег – тільки 1. Таким чином, правило:

```
#copy b p {color: red}
```

має специфічність 102, тоді як специфічність правила:

```
p {color: lime}
```

оцінюється тільки в 1. Виходячи з цього колір абзацу буде червоним.

- Порядок слідування. Якщо конфліктують правила, рівні по переліченим попереднім 5 показникам, віддається перевага правилу, переліченому останнім. При цьому треба мати на увазі, що лінійні стилі завжди виграють.
- Стилi, успадковані від предків. На цьому рівні застосовуються стилі, успадковані від батьківських елементів.
- Стилi браузера за замовчуванням. Ці стилі мають найнижчий пріоритет і застосовуються, якщо для елементів відсутні інші оголошення стилів.

2.1.3 Мова програмування JavaScript

На даний період часу мова програмування JavaScript, поряд з Java, PHP, Python, C#, C++, Ruby, CSS, C та Objective-C, входить в першу десятку більшості рейтингів найбільш використовуваних мов програмування. Наприклад, випущений в середині 2016 року гібридний рейтинг RedMonk (<http://redmonk.com/sograzy/2016/07/20/language-rankings-6-16/>) ставить мову програмування Javascript на перше місце серед 20-ти найбільш популярних мов програмування [18].

Мову програмування JavaScript можна застосовувати для обчислень, побудови меню, перевірки правильності заповнення форм, зміни зображень і так далі. Програми на JavaScript зазвичай виконуються браузером або компілюються в звичайні додатки на комп'ютерах клієнтів, і в цьому випадку мова називається клієнтською мовою JavaScript. У разі ж, коли програми на JavaScript використовуються на веб-сервері для обробки запитів, обчислень, роботи з даними і формування документів HTML, мова називається серверною мовою JavaScript [22].

Мова JavaScript була створена в Netscape в перші роки існування Інтернет, а «JavaScript», що є торговою маркою Sun Microsystems (тепер Oracle), використовується для опису реалізації мови, виконаної Netscape (тепер Mozilla). Netscape представила мову для стандартизації в Європейську асоціацію виробників комп'ютерів ECMA (European Computer Manufacturer's Association), а стандартизована версія мови почала носити назву «ECMAScript».

Протягом минулого десятиліття практично всіма браузерами була завершена реалізація стандарту мови ECMAScript 3. Мова ECMAScript 4 була у стадії розробки протягом багатьох років, але виявилася дуже амбітною і специфікація на неї не була випущена. Декілька років тому з'явився специфікація мови ECMAScript 5, а потім специфікація ECMAScript 6, і практично всі виробники браузерів в даний час цей стандарт мови JavaScript реалізували.

2.1.4 Серверні мови програмування

Сучасні серверні технології створення і підтримки веб-сайтів орієнтовані на платформи, що дозволяють ефективно управляти інформаційним наповненням даними, які надходять до відвідувачів сайту. Як правило, такі рішення базуються на серверних технологіях типу ASP, ASP.NET, JSP, PHP і використовують готові потужні засоби для створення корпоративних сайтів, орієнтованих на впровадження зазначених технологій. Розглянемо докладніше популярні інформаційні технології.

Створення веб-сторінок фрагментами серверного коду може виконуватись за допомогою технологій ASP, ASP.NET (Active Server Pages) [23]. Це розроблена Microsoft комерційно доступна технологія, за допомогою якої веб-майстер може самостійно формувати динамічно обновлювані веб-сторінки. Характерною особливістю цієї технології є можливість відділення функціональної частини розробки від процесів створення дизайну. ASP-сторінки можуть містити HTML-текст, змішаний зі сценаріями як JavaScript і VBScript. В процесі обробки запиту нової сторінки його виконує сервер і динамічно генерує браузеру потік HTML-тексту, який відображається на екрані монітора. ASP-технологія Microsoft отримала подальший розвиток в технологіях JSP, PHP і ін. Технологія JSP (Java Server Pages) - це технологія створення серверних сторінок Java. Специфікація JSP є розширенням Java Servlet API для генерації динамічних веб-сторінок на веб-сервері. Така крос-платформа є альтернативою технології ASP корпорації Microsoft.

Специфікація Sun під назвою JSF (Java Server Faces) реалізує технологію JSP, що описує правила створення веб-додатків зі зручним для користувача інтерфейсом і орієнтована на розробку серверних компонентів створення візуального інтерфейсу [25].

Однією з перших технологій створення веб-додатків, які виконуються сервером, була Common Gateway Interface (CGI) технологія [26]. Вона дозволила розробку і виконання серверних додатків, звернення до яких відбувається за допомогою зазначеного в URL імені (і параметрів).

Залежно від обраного протоколу вхідною інформацією таких веб-додатків вважають безпосередньо код HTTP-заголовка або запит пошукової системи. CGI-додатки - це консольні додатки, які генерують HTML-код для його передачі браузеру. Серед інших популярних технологій, що реалізують створення веб-сторінок з фрагментами коду, який виконується на сервері, можна виділити некомерційну, вільно поширювану технологію PHP (Personal Home Pages) [27]. Ця технологія заснована на використанні CGI-додатків, які інтерпретують впроваджений в HTML-сторінку код на скриптовій мові.

Головною особливістю мови PHP є її практичність. PHP надає програмісту інструменти для швидкого і ефективного вирішення поставлених задач. Вона відрізняється винятковою гнучкістю до потреб розробника. хоча PHP традиційно рекомендують використовувати в поєднанні з HTML-кодом, проте PHP з таким же успіхом інтегрується і з JavaScript, WML, XML та іншими мовами Інтернет-програмування (табл. 2.1)

Таблиця 2.1 – Характеристики серверних мов програмування

Характеристики	C# (ASP.NET)	Java	Python	Perl	Ruby	PHP
Безкоштовна ліцензія	-	+	+	+	+	+
Вартість додаткового ПЗ (ОС, БД)	> 40 т.р.	-	-	-	-	-
Кросплатформеність	-	+	+	+	+	+
Достатня кількість довідкових матеріалів	+	+	+	-	-	+
Легкість у вивченні	-	-	-	-	+	+
Багато готових систем управління контентом	-	-	-	-	-	+

Для створення динамічних веб-сайтів використовують серверні мови веб-програмування. Найбільш використовуваними мовами є наступні: C#, Java, PHP, Python, Perl, Ruby. Серверні програми, написані на цих мовах, дозволяють автоматично, за запитом користувача, генерувати статичні HTML-сторінки і передавати їх користувачеві.

Мова C# та технологія ASP.NET, [24, 26]. яка була описана вище, розроблені компанією Microsoft і працюють на платформі Windows з використанням веб - сервера IIS (Internet Information Services) та СУБД Microsoft SQL Server.

Python застосовується як інтерпретуюча мова для скриптів різного призначення. Задачі на мові Python описуються з використанням псевдокоду, що дозволяє зменшити обсяг програми. Цю мову можна використовувати як в великих, так і в малих проектах, інтенсивно застосовується такими корпораціями як Google і Yandex.

Java - повністю об'єктно-орієнтована мова програмування, розроблена компанією Sun Microsystems і вданий час підтримується ORACLE. За допомогою Java можна реалізувати проект будь-якої складності. Java здатна працювати на будь-якому комп'ютері, завдяки використанню віртуальної Java-машини.

Perl - високорівнева інтерпретуюча мова динамічного програмування загального призначення. Основна особливість цієї мови - це широкі можливості при роботі з текстом, з регулярними виразами.

Ruby - динамічна об'єктно-орієнтована мова програмування, яка характеризується динамічною типізацією і автоматичним управлінням пам'яттю. Мова Ruby використовується в веб-розробці в складі відкритого веб-фреймворку Rails, частіше званого Ruby on Rails.

PHP - мова сценаріїв, що виконується на стороні веб-сервера, розроблен Рasmusом Лердорфом як інструмент створення динамічних та інтерактивних веб-сайтів.

Мова PHP досить гнучка і потужна, тому ця мова використовується в проектах будь-якого масштабу - від простого блогу до найбільших веб-додатків в Інтернеті. В даний час підтримується більшістю хостинг-провайдерів і є одним з лідерів серед мов програмування, що застосовуються для створення динамічних веб-сайтів. Код PHP інтерпретується на веб-сервері і генерує HTML-сторінки або інші документи, які запитуються користувачами сайтів.

Аналізуючи дані таблиці 2.1, треба зробити висновок, що для розробки сайтів ведення блогів буде оптимальним використовувати серверну мову програмування PHP через наступні її переваги:

- PHP поширюється під ліцензією «PHP License», розробленою «PHP Group»; ця ліцензія схвалена «Open Source Initiative» (OSI).
- PHP підтримує можливість роботи з багатьма операційними системами, має спеціальний репозитарій PECL, в якому міститься понад 140 готових розширень, таких як: робота з графікою, робота з різними СКБД, і т. і.
- PHP має велике співтовариство її прихильників, тому на основі цієї мови розроблено багато готових систем керування вмістом (табл. 2.2).
- PHP є основною мовою програмування у багатьох провайдерів, які надають послуги хостингу для розміщення веб-сайтів.

Таблиця 2.2 Характеристики популярних систем управління контентом

Характеристики	1С Bitrix	WordPress	Joomla!	Drupal
Безкоштовна ліцензія	-	+	+	+
Мови Програмування	ASP.NET , PHP	PHP	PHP	PHP
База даних	MySQL, MS SQL, Oracle	MySQL	MySQL	MySQL, PostgreSQL
Найбільш підходить як	Багато-функційна	Сайт-блог	Сайт-візитка	Багато-функційна
Продуктивність	Висока	Висока	Середня	Висока
Можливість розширення	+	-	+	+

2.2 Огляд і порівняльний аналіз базових технологій створення сайтів

У наш час в питанні, яку ж мову програмування використовувати для створення сайту, у розробників веб-сайтів є не один і не два вибори. Є багато варіантів: Perl, PHP, ASP, ASP.NET, JSP. Найпоширенішими (з великим відривом від інших) сьогодні є PHP і ASP.NET. Крім цього, для зберігання даних використовуються бази даних, такі як MySQL або ORACLE.

2.2.1 Порівняння технологій PHP і ASP.NET

PHP (Hypertext Preprocessor) - мова для написання серверних сценаріїв (скриптів). Інтерпретатор мови безкоштовний, з відкритим початковим кодом, створені версії для різних веб-серверів - перш за все, для Apache і Internet Information Services. Актуальна на сьогоднішній день версія інтерпретатора – PHP 7.

ASP.NET (Active Server Pages.NET) - засоби для розробки веб-додатків на платформі Microsoft ASP.NET - розцінюється не як мова програмування, а скоріше технологія, яка перш за все включає в себе багато компонентів. Поточна версія на останній день - ASP.NET Core 2.0. Слід зазначити, що ASP.NET радикально відрізняється від свого попередника.

Зрозуміти принципи роботи PHP набагато простіше, ніж ASP.NET. Це одна з головних причин, чому PHP зараз більш поширений. «Поріг входження» у PHP набагато нижче, і новачкові набагато простіше зрозуміти його принципи роботи, ніж складну модель ASP.NET [18].

Перевагою є простота вивчення, однак вона ж є і недоліком: це призводить до того, що низький рівень розробників на PHP зустрічається набагато частіше, ніж на ASP.NET. Якщо у програміста недостатньо досвіду або знань - то мову PHP він все одно освоїть, в той же час як при використанні ASP.NET виникнуть труднощі. Це означає що, звертаючись до PHP-розробника, шанс отримати низькоякісний продукт вище, ніж в разі використання ASP.NET.

До безперечних достоїнств PHP я б відніс його «низькорівневність».

На відміну від PHP, ASP.NET - спроба Microsoft застосувати при розробці веб-додатків ті ж принципи, що і при розробці Windows-додатків. Наприклад, на сторінці створюється кнопка, для якої пишеться обробник події, який при натисканні змінює текст у якого-небудь серверного елемента, і так далі. Але така модель не зовсім застосовна при розробці веб-додатків. Протокол HTTP - це сеансовий протокол, який працює за

принципом запит-відповідь. А Windows-програми працюють безперервно (проводячи аналогію, можна сказати, що вони використовують постійне з'єднання).

Тому для реалізації аналогічного підходу довелося застосовувати багато абстракцій і прийомів. Всі ці абстракції покликані приховати «сеансовість» протоколу HTTP. Перевагою їх є те, що значно прискорюється процес розробки веб-додатків. Але якщо програміст починає вивчення веб-розробки з ASP.NET - досить імовірно, що він не відразу зрозуміє справжні (фізичні) принципи роботи веб-додатків. Наприклад, якщо потрібно отримати від відвідувача сайту які-небудь дані і обробити їх (наприклад, записати в базу даних), PHP- розробнику доводиться створити HTML-форму, відповідний скрипт, який отримає від неї дані, обробить їх, і поверне відвідувачеві відповідь, також у вигляді HTML-розмітки - так і працюють всі веб-додатки: запит - відповідь. Якщо аналогічне завдання вирішує ASP.NET-розробник, то форма за нього вже створена, він лише додає поле для введення і кнопку, за якою закріплює обробник події, щоб виконати потрібні дії по обробці введених даних.

2.2.2 Аналіз та вибір веб-сервера

Веб-сервер - це будь-яке управляюче комп'ютером програмне забезпечення, яке дає йому можливість виконувати запити на документи або інші дані. Програми, які запитують і відображають документи (такі як браузер), називаються клієнтами. Терміни "на стороні сервера" і "на стороні клієнта", що використовуються, наприклад, при роботі з картами-зображеннями, відносяться до тієї машини, яка керує процесом. Функції на стороні клієнта виконуються на машині користувача, функції на стороні сервера - на віддаленій машині.

Веб-сервер - спеціальне програмне забезпечення, яке забезпечує доступ користувачам до інформації, яка зберігається на серверному комп'ютері. Саме на веб-серверах знаходяться сайти і веб-сторінки.

Серверне програмне забезпечення - пакет програм обчислювальної системи, що виконують функції зберігання, обробки і збору даних за запитом клієнта, надаючи йому доступ до певних ресурсів або послуг.

Веб-сервери відповідають на запити браузерів (клієнтських програм), знаходять задані файли і повертають документи або результати сценаріїв.

Веб-браузери і сервери спілкуються по протоколу Hypertext Transfer Protocol (HTTP, протокол передачі гіпертексту). Веб-сервер аналізує прийняті HTTP-запити від клієнтів, зазвичай веб-браузерів, і формує HTTP-відповіді, які можуть супроводжуватися HTML-документами, зображеннями, файлами, медіа-потокком або іншими даними [37].

Хостинг - послуга з надання обчислювальних потужностей для фізичного розміщення інформації на сервері, який, зазвичай, має цілодобовий вихід в мережу Інтернет.

Як правило, разом з хостингом сайтів, надаються послуги поштових сервісів, послуги баз даних, послуги DNS (Domain name system) і підтримка інших сервісів.

Доменне ім'я - це ім'я, що служить для ідентифікації областей - одиниць адміністративної автономії в мережі Інтернет - в складі вищої по ієрархії такої області. Кожна з таких областей називається доменом. Спільний простір імен Інтернету функціонує завдяки DNS - системі доменних імен. Доменні імена дають можливість адресації Інтернет-вузлів і розташованих на них мережевих ресурсів (веб-сайтів, серверів електронної пошти, інших служб) в зручній для людини формі [16].

HTTP-сервер Apache - веб-сервер з відкритим початковим кодом. З квітня 1996 року, є найпопулярнішим HTTP-сервером в Інтернеті; У квітні 2017 року відсоток його використання на веб-серверах дорівнював 58%.

Основними перевагами Apache вважаються надійність і гнучкість конфігурації. Все це дозволяє користатися зовнішніми модулями, які використовуються для аутентифікації користувачів, підключати бази даних, змінювати повідомлення про помилки і т. д.

Недоліком є відсутність зручного стандартного інтерфейсу для адміністратора.

Сервер був написаний на початку 1995 року, і вважається, що його назва походить від імені групи племен північноамериканських індіанців - апачів. Пізніше, в версії 2.x, сервер був переписаний і не містив NCSA код, але назва прижилася. В даний час в розробці знаходиться версія 2.4, а версії 1.3 і 2.0 випускаються тільки для підтримки безпеки.

Веб-сервер Apache розробляється і підтримується відкритим співтовариством розробників під егідою Apache Software Foundation, і використовується в багатьох програмних продуктах, в тому числі Oracle і IBM WebSphere.

2.2.3 Аналіз систем керування базами даних (СКБД)

Робота з базами даних є однією з головних складових процесу програмування сайту динамічного типу. Бази даних для сайтів використовують з метою зберігання різнопланової інформації. Спрощено бази даних являють собою певний набір взаємозалежних таблиць. Розміри таблиць у базах є різними, а їх кількість – довільною. Бази даних на сервері акумулюють необхідну для роботи сайту інформацію статистичного характеру.

До найпопулярніших мережево-орієнтованих баз даних відносять MS SQL Server,

MySQL, Oracle Database та інші. Вибір потрібної системи керування базою даних (СКБД) обумовлюється вимогами до інформаційних характеристик та функціональних можливостей системи.

Однією з найпоширеніших систем керування базами даних у наш час вважається MySQL [28], яка є альтернативою комерційним системам. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL – компактний багатопотоковий сервер баз даних. Серед переваг сервера MySQL виділимо простоту у встановленні та використанні, підтримку паралельної роботи значної кількості користувачів. Кількість рядків (записів) у таблицях даних може сягати 50 млн. Висока швидкість виконання команд, наявність простої та ефективної системи безпеки обумовлюють зростання популярності MySQL [22]. Проте основною перевагою СКБД MySQL є можливість її безкоштовного використання.

Об'єктно-реляційна система керування базами даних Oracle Database компанії Oracle орієнтована на операційні системи Windows, Unix, Linux та MacOS. Oracle Database, на відміну від MySQL, має більш широку сферу застосування [29]. СКБД Oracle Database широко відома як у нашій країні, так і у світі. На ній базується багато сучасних інформаційних систем. Однак, у процесі програмування сайтів для невеликих і середніх компаній через свою високу вартість вона використовується нечасто. Крім того, досить складно знайти хостинг з підтримкою такої СКБД.

Мова SQL (Structured Query Language – мова структурованих запитів) – декларативна мова програмування засобів інтерфейсної взаємодії користувача з базами даних, що реалізує процеси формування запитів, оновлення і керування реляційними базами даних, створення схеми бази даних і її модифікації, систему контролю за доступом до інформаційних ресурсів. SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати в якості інструкцій для керування даними. Крім того, стандарт SQL включає функції визначення процесів зміни, перевірки і захисту даних. Табл. 2.3 акумулює результати критеріального аналізу сучасних СКБД.

Таблиця 2.3 – Порівняльна характеристика СКБД

Характеристики	MS SQL Server	Oracle	MySQL
Надійність	+	+	+
Швидкодія	-	+	+
Простота	-	-	+

Продовження табл..2.3.

Зручність використання	+/-	+	+
------------------------	-----	---	---

Проаналізувавши базові характеристики розглянутих СКБД, можна зробити висновок про зручність використання засобів MySQL для створення динамічних веб-сторінок, оскільки сервер MySQL забезпечується підтримкою з боку різноманітних мов програмування, зокрема і PHP. MySQL задовольняє основним вимоги до СКБД у процесі розробки веб-сайту, так як вона є надійною, має високу швидкість, просту у встановленні та зручна у використанні. Завдяки своїй доступності, швидкості та відмовостійкості MySQL забезпечує ефективний доступ до баз даних з Інтернету.

2.2.4 Порівняльний аналіз СКБД MySQL та ORACLE

Починаючи з п'ятої версії MySQL підтримує збережені процедури і функції, обробники помилок, курсори, тригери, представлення. Щоб встановити собі на комп'ютер MySQL можна скачати або початкові коди, або готовий виконуваний модуль для використовуваної ОС. В MySQL використовується трохи «підрізаний» стандартний SQL. MySQL підтримує дуже великий список платформ. Треба сказати, що майже всі хостери надають хостинг разом з MySQL - тому вона і є такою популярною. ORACLE підтримує всі вищеописані функції, але в ній є ще ряд особливих функцій: об'єктно-орієнтовані властивості, автономні транзакції, послідовності, пакети, аналітичні функції. Для роботи можна використовувати як стандартний SQL, так і мову програмування PL/SQL - в неї додано багато додаткових функцій, що спрощують роботу програміста.

Крім того, останні на даний момент версії СКБД ORACLE 11g та ORACLE 12c - це перша в світі база даних, спеціально розроблена для роботи в мережах розподілених обчислень Grid. ORACLE [29, 30], також, як і MySQL, підтримує дуже багато різних платформ.

MySQL можна скачати разом з веб-сервером і інтерпретатором PHP в пакеті Denwer (або, наприклад - Top Server), потім досить встановити - і все буде працювати. Створювати, змінювати і видаляти об'єкти можна за допомогою кліків мишею. Для початківців дана система - ідеальна. За замовчуванням користувач має ім'я root і порожній пароль. Тому обов'язково треба створити нового користувача і задати йому необхідні привілеї і пароль. Всі налаштування автоматично здійснюються в СКБД MySQL [15]. Щоб встановити ORACLE - доведеться трохи повозитися, але цей процес все ж досить простий.

Там теж велика частина функцій доступна по клацанню, але треба знати - де що лежить. Зате потім, коли все буде вивчено - відкриваються величезні можливості по створенню, редагуванню і видалення всіх об'єктів бази даних. За замовчуванням, адміністратор СКБД має логін і пароль MANAGER. Після входу також треба створити свого користувача. В плані користування - MySQL виграє, так як вона дуже проста. Виходячи з цього були обрані мова PHP і СКБД MySQL, тому що вони прості до вивчення, найбільш поширені, підтримуються всіма хостинговими компаніями і працюють на будь-яких платформах.

jQuery - бібліотека JavaScript, що фокусується на взаємодії JavaScript і HTML. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого елементу DOM, звертатися до атрибутів і вмісту елементів DOM, маніпулювати ними. Також бібліотека jQuery надає зручний API для роботи з AJAX. Зараз розробка jQuery ведеться командою jQuery на чолі з Джоном Резігом.

Комбінація jQuery і AJAX забезпечують потужну функціональність. За допомогою jQuery і AJAX ви можете зробити запит і отримати інформацію в різних форматах, включаючи XML, HTML і навіть звичайний текст. Для обміну даними можна використовувати формат JSON. Дані, отримані за AJAX-запитом, ми можемо використовувати в нашій HTML-сторінці.

jQuery робить існуючі браузерні AJAX API могутнішими і простішими у використанні. Створювати виклики AJAX звичайним способом, використовуючи JavaScript, трохи важко, так як необхідно враховувати, що для різних браузерів потрібні різні підходи до створення об'єкта XMLHttpRequest. Крім того, відправляти дані, наприклад, з форм, стає складніше, якщо, використовується звичайна мова JavaScript для виклику AJAX.

jQuery забезпечує просту і потужну функціональність, яка розширює AJAX-методи JavaScript і забезпечує більш гнучкий підхід.

2.3 Аналіз переваг і недоліків систем управління контентом (CMS)

Система управління контентом (CMS) – комплекс комп'ютерних програм, який використовується для організації та забезпечення процесу створення, управління і редагування вмісту сайту. Сьогодні існує багато CMS: від популярних, за якими стоять величезні компанії-розробники, до саморобних, нікому не відомих.

CMS бувають абсолютно різні, деякі системи орієнтовані тільки на вирішення конкретних завдань (ведення блогів, Інтернет-магазини, форуми), інші є універсальними і надають розробникам зручне середовище проектування і програмування для розробки чого завгодно [4]. Частина CMS складатися з багатьох функціональних блоків і модулів,

інші монолітні, неподільні, та ще й зашифровані. Одні системи поставляються безкоштовно і з можливістю внесення своїх доробок, а деякі надаються за гроші і не допускають можливість редагування ядра.

В основному сьогоднішні масові CMS використовують генерацію HTML/XML-документів з використанням двох скриптових мов: PHP - на стороні сервера, JavaScript - на стороні клієнта (браузера). При цьому можливості стандартного серверного програмного забезпечення, що складається з трьох основних частин - мережевого сервера, транслятора PHP, системи керування базами даних (зазвичай, MySQL) - можуть використовуватися на різних рівнях.

Існують різноманітні системи управління контентом, серед яких зустрічаються платні і безкоштовні, популярні і непопулярні, побудовані за різними технологіями. Кожен сайт має панель управління, яка є тільки частиною всієї системи, але достатня для управління ним.

Комерційні коробкові CMS - продукти, створені комерційними організаціями з метою отримання прибутку від продажу ліцензії і / або технічної підтримки. Серед відомих платних систем управління можна виділити: 1С-Бітрікс, UMI.CMS, NetCat, HostCMS і інші.

Безкоштовні CMS з'явилися, як альтернативне бізнес-рішення, з тією лише різницею, що в першому випадку компанія заробляє на продажу ліцензій, а в другому - на підтримці та впровадженні будь-якого веб - ресурсу. За своєю функціональністю безкоштовні CMS не поступаються платним аналогам і також популярні серед розробників. До таких систем відносяться: Joomla!, WordPress, Drupal, MODx, OpenCart та інші [9].

Приватні розробки невеликих студій, вузькоспеціалізовані CMS і інші рішення можна віднести до непопулярних CMS.

Використання CMS при розробці сайтів має свої переваги та недоліки.

До переваг можна віднести:

- можливість самостійного наповнення;
- можливість розподіленого адміністрування;
- структурованість інформації;
- розширення можливостей за допомогою готових модулів;
- автономність (клієнт не буде залежати від веб-студії»).

Недоліками використання CMS є:

- подорожчання розробки сайту за рахунок більш складного програмування;
- наступні доопрацювання сайту, пов'язані з доопрацюваннями CMS;

- плата за ліцензію (у разі платної CMS);
- підвищена оплата хостинг-майданчика;
- вимогливість до ресурсів (сайт довше завантажується);
- наявність вразливостей;
- низький рівень безпеки (частіше для безкоштовних CMS).

Як видно, основним аспектом установки CMS є отримання можливості самостійного редагування веб-ресурсу.

2.3.1 Класифікація систем управління контентом

Чіткої класифікації систем управління контентом в даний час не існує, оскільки такий вид програмних продуктів ще відносно новий. За типом розробки їх можна поділити наступним чином [46].

1. Модульні системи. Являють собою порівняно недорогі рішення, орієнтовані на представницькі сайти для середнього бізнесу і корпоративні веб-ресурси, що розробляються як "первинні" проекти. Вони призначені для створення і підтримки корпоративних або тематичних веб-сайтів. Дуже багато виробників контент-систем будують свої продукти за модульним принципом. У цьому підході є переваги. Виробник може розраховувати на велику аудиторію в залежності від потреб компаній - потенційних покупців.

У свою чергу, компанії-покупці можуть поступово придбавати нові модулі і підвищувати функціональність і гнучкість свого електронного ресурсу [49]. Варто зауважити, що купити відразу повний пакет програм за кілька тисяч доларів може дозволити собі далеко не кожна фірма.

2. Спеціалізовані системи. Орієнтовані на певний вид діяльності і відображення її на електронному ресурсі. Служать для управління Інтернет-спільнотами, реалізації рішень для електронної комерції, ЗМІ і т. д. Базової функціональності цих систем досить для обслуговування найскладніших процесів певного типу, чого немає в універсальних продуктах. З іншого боку, їх впровадження набагато дешевше, ніж замовна розробка.

3. Замовні системи. Розробка системи ведеться з урахуванням цілей, завдань і побажань замовника.

Існує також класифікація систем управління контентом за призначенням.

1. Системи управління веб-контентом. Призначені для управління веб-контентом на сайті (додавання, видалення, редагування контенту), більш складні розробки забезпечують функції адміністрування існуючих веб-ресурсів (зміна структури веб-

сайту, розмежування прав доступу), а також можуть бути інструментом для розробки нових веб-ресурсів. За допомогою цих систем власник може самостійно управляти вмістом свого ресурсу, не вдаючись до послуг розробників. Дані продукти створюються, як правило, дрібними веб-студіями і окремими веб-розробниками. З цієї причини їх не можна розглядати як серйозних конкурентів.

2. Системи електронного документообігу підприємства, що використовують веб-інтерфейс. До даної категорії систем управління контентом відносяться системи документообігу масштабу підприємства, в яких є функції роботи з веб-документами через веб-інтерфейс. Системи документообігу часто використовуються для оптимізації бізнес-процесів середніх і великих підприємств. Вони інтегровані з корпоративною інформаційною системою підприємства і дозволяють створювати портальні інтранет-рішення, системи взаємодії з постачальниками і збутовою мережею.

Системи, побудовані за модульною схемою, дозволяють нарощувати свою функціональність шляхом введення додаткових модулів.

2.3.2 Аналіз використання систем управління контентом

Системи управління контентом дозволили знайти компроміс між низькою гнучкістю статичних сайтів та високою вартістю динамічних сайтів і в результаті підвищити керованість і знизити витрати на адміністрування ресурсу.

Система управління контентом - це автоматизований редакторський комплекс, що служить для управління змістом і структурою Інтернет-ресурсу в онлайн-режимі. На практиці це означає, що будь-який, навіть незнайомий з інформаційними технологіями користувач може абсолютно самостійно управляти змістом сайту, не вдаючись при цьому до послуг професіоналів. За рахунок обмежень, що накладаються на логічну структуру контенту, на дизайн і функціональні можливості створюваних динамічних сайтів, при використанні таких систем радикально знижується трудомісткість розробки і підтримки.

Ринок систем управління контентом порівняно молодий, але вже встиг зайняти свою нішу в сфері розробки сайтів [47]. Розробники контент-систем пропонують своїм клієнтам продукцію різного рівня складності в залежності від обсягів фірми, роду діяльності, кількості інформації, обслуговуючого персоналу, максимальної пропускної здатності і т. д. Відрізняється і структура систем.

Варто зауважити, що якою б складною не була система, розробники орієнтуються на деякі загальні принципи, в основі яких лежить уявлення про життєвий цикл контенту.

Підтримка розробки контенту по суті справи є одним з ключових компонентів таких програмних продуктів. Саме з розробки починається життєвий цикл будь-якого

матеріалу, опублікованого на сайті: ведуться роботи по його створенню, редагуванню та утвердження контенту, а роль системи полягає в автоматизації цих процесів, підтримці спільної роботи авторів, редакторів, програмістів і менеджерів.

Досягнення поставленої мети здійснюється завдяки поділу контенту і дизайну. Всі компоненти сайту, включаючи шаблони і наповнення, зберігаються в певних місцях сховища даних. Система ж автоматично звертається в потрібні місця сховища, дозволяючи багатьом користувачам, навіть які не є технічними фахівцями, працювати над підготовкою контенту до публікації, включаючи перевірку його достовірності.

Управління діловими процесами включає етапи розробки структури ресурсу, попереднього перегляду і публікації підготовленого контенту. Визначається зовнішній вигляд, готуються шаблони, розподіляються ролі користувачів і класифікується необхідна бізнес-інформація (товари, ціни, обсяги та т. і.). Важливими компонентами тут є служби, які підтримують своєчасність надходження необхідного контенту [48].

При цьому ведеться аудит подій і зберігаються версії документів, причому при необхідності користувачі можуть звертатися до попередніх версій.

Коли сайт повністю підготовлений до публікації, в роботу вступають засоби динамічного формування веб-сторінок в залежності від прямої і непрямой інформації про конкретного споживача. Пряма інформація збирається з особистих карток, непряма - переважно на основі поведінки користувача.

Є три основні форми динамічного поширення контенту - персоналізація, локалізація і глобалізація. Безумовно, контент може бути статично або динамічно згенерований і надано відвідувачам безпосередньо, без застосування будь-якої поведінкової логіки.

Персоналізація в тій чи іншій мірі використовується тепер на більшості веб-сайтів. Тут вступають в хід особисті вітання, рекламні оголошення і особливим чином скомпоновані фрагменти контенту, все це покликане сприяти залученню відвідувачів на сервер.

Існує два методи персоналізації: на основі правил і за допомогою фільтрів.

Персоналізація на базі правил - це надання контенту певним користувачам або їх групам із застосуванням умовної бізнес-логіки. Для розробки правил зазвичай служить введення користувачем в реєстраційну картку інформації про себе і своїх перевагах.

При персоналізації за допомогою фільтрів застосовуються складні алгоритми категоризації і надання контенту на основі аналізу поведінки користувача - до якої інформації він звертається, які сайти відвідує і т. д.

2.4 Переваги та недоліки вибору технологій PHP-MYSQL

Нижче наведено аналіз переваг і недоліків побудови сайту або блогу за допомогою технологій PHP-MySQL

Головною **перевагою** використання мови PHP для розробки сайтів є її практичність. PHP надає програмісту засоби для швидкого і ефективного вирішення поставлених перед ним завдань. Практичний характер PHP обумовлений п'ятьма важливими характеристиками: традиційністю, простотою, ефективністю, безпекою та гнучкістю.

Існує ще одна «характеристика», яка робить мову PHP особливо привабливою [30]: він поширюється безкоштовно! Причому, з відкритими початковими кодами (Open Source).

Традиційність. Мова PHP здається знайомою програмістам, що працюють в різних областях. Багато конструкцій мови запозичені з C++ та Perl. Код PHP дуже схожий на той, який зустрічається в типових програмах на C++, Perl або й Pascal. Це помітно знижує початкові затрати при вивченні PHP. PHP - мова, що поєднує переваги Perl та C++ і спеціально зорієнтованою на роботу в Інтернеті, мова з універсальним (за деякими застереженнями) і зрозумілим синтаксисом.

Простота. Сценарій на мові PHP може складатися з 10 000 рядків або з одного рядка - все залежить від специфіки вашого завдання.

Мова PHP, інструкції якої можуть бути вбудованими безпосередньо в HTML-код сторінок, який також буде коректно оброблятися PHP-інтерпретатором. Можна використовувати PHP для написання CGI-сценаріїв і позбутися від багатьох незручних проблем виведення тексту. Велика розмаїтість функцій PHP позбавлять вас від написання багаторядкових функцій користувача.

Ефективність. Дуже важлива перевага PHP полягає в його «движку». «Движок» PHP не є ні компілятором, ні інтерпретатором. Він дозволяє обробляти сценарії на PHP з достатньо високою швидкістю. За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше аналогічних їм програм, написаних на Perl. Однак, щоб не робили розробники PHP, відкомпільовані виконувані файли будуть працювати значно швидше - в десятки, а іноді і в сотні разів. Таким чином, продуктивність PHP цілком достатня для створення серйозних веб-сайтів та веб-додатків.

Безпека. PHP надає в розпорядження розробників і адміністраторів гнучкі і ефективні засоби безпеки, які умовно поділяються на дві категорії: засоби системного рівня і засоби рівня додатка.

Гнучкість. Оскільки PHP є вбудовуваною (embedded) мовою, він відрізняється винятковою гнучкістю по відношенню до потреб розробника. Хоча PHP зазвичай рекомендується використовувати в поєднанні з HTML, він з таким же успіхом інтегрується і з JavaScript, WML, XML та іншими мовами. Крім того, добре структуровані додатки на PHP легко розширюються в міру необхідності.

Немає проблем і з залежністю від браузерів, оскільки перед відправкою клієнту сценарії PHP повністю компілюються на стороні сервера. По суті, сценарії PHP можуть передаватися будь-яким пристроям з браузерами, включаючи стільникові телефони, електронні записники, пейджери і портативні комп'ютери, не кажучи вже про традиційні ПК.

Безкоштовне розповсюдження. Стратегія Open Source і розповсюдження початкових текстів програм зробили безсумнівно благотворний вплив на багато проектів, в першу чергу, на програмне забезпечення для ОС Linux, хоча і успіх проекту Apache сильно підкріпив позиції прихильників Open Source.

Сказане стосується і до історії створення PHP, оскільки підтримка користувачів зі всього світу виявилася дуже важливим чинником у розвитку проекту PHP [31]. Ухвалення стратегії Open Source і безкоштовне розповсюдження початкових текстів PHP надало неоціненну послугу користувачам.

До того ж, велике співтовариство користувачів PHP є свого роду «колективною службою підтримки», і на популярних електронних конференціях можна знайти відповіді навіть на найскладніші питання.

MySQL - це одна з найпопулярніших і найпоширеніших СКБД (система керування базами даних) в Інтернеті. Вона не призначена для роботи з великими обсягами інформації, але її застосування ідеально підходить для Інтернет-сайтів, як невеликих, так і досить великих.

MySQL відрізняється хорошою швидкістю роботи, надійністю, гнучкістю. Робота з нею, як правило, не викликає великих труднощів. Підтримка сервера MySQL автоматично включається в поставку PHP.

Важливим фактором є безкоштовність СКБД MySQL. MySQL поширюється на умовах загальної ліцензії GNU (GPL, GNU Public License).

Додаток на PHP, що використовує для зберігання інформації базу даних MySQL завжди працює швидше за додатки, побудованих з використанням файлів або багатьох інших СКБД.

Вартість. PHP - одна з мов, позначених буквою "P" в аббревіатурі LAMP, яка позначає широко використовуваний стек програм. В стек LAMP входять такі поширені

програмні продукти, як Linux, Apache, MySQL та PHP / Perl / Python, які експлуатуються на багатьох веб-сайтах і підтримують багато веб-додатків. Значна частина компонентів стека LAMP надається безкоштовно, і система PHP не є винятком.

Ліцензія на СКБД MySQL надається на інших умовах, які не завжди допускають відсутність плати, але в будь-якому випадку залишається можливість отримати версію Community Server безкоштовно. Розробники СКБД MySQL пропонують кілька рівнів контрактів підтримки на свій сервер бази даних. Більш докладні відомості можна отримати за адресою www.mysql.com.

Зручність у використанні. Порівняння з багатьма іншими мовами програмування показує, що мова PHP дозволяє швидше розробляти потужні веб-додатки. У мові PHP вже заздалегідь визначені багато з найбільш корисних спеціалізованих функцій, наприклад, для роботи з базами даних.

А що стосується СКБД MySQL, то для програмістів, які не мають досвіду роботи з реляційними базами даних або знайомих лише з таким середовищем, як Microsoft Access, інтерфейс командного рядка цієї СКБД, а також відсутність готових структурних рішень можуть на перший погляд здатися досить складними. В СКБД MySQL передбачено кілька інструментів з графічним інтерфейсом користувача, що дозволяють спростити роботу з базами даних. Однак, які б то не було графічні інструментальні засоби, вони не відмінюють вивчення теорії і не підкажуть, в чому полягають методи якісного проектування.

В якості **недоліків** наведемо короткий перелік функціональних можливостей, яких не вистачає в СКБД MySQL.

Транзакції - дозволяють об'єднати кілька SQL-запитів в одну одиницю роботи і в разі збою будь-якого із запитів, що входять в цю одиницю, виконати відкат, щоб повернути дані в початковий стан. Пояснимо на прикладі.

Необхідно зняти гроші з одного рахунку і покласти на інший. Для цього потрібно виконати два SQL-запити: перший - зняти гроші з одного рахунку, другий - зарахувати їх на інший рахунок. Якщо не застосовувати транзакцію, то в разі збою при виконанні другого запиту гроші будуть зняті з рахунку, але не будуть зараховані на інший рахунок. Застосування транзакцій дозволяє зробити відкат до того стану, в якому гроші ще не знімалися з рахунку.

Зауважимо, що за допомогою команди LOCK TABLES в MySQL можна емулювати транзакцію. Ця команда блокує таблицю на час виконання запитів, і тим самим забезпечується цілісність даних, але відкат все одно не можна зробити.

Тригери - служать для автоматизації контролю за станом і роботою бази даних.

Тригер зберігається в базі і спрацьовує, коли відбувається певна подія. Візьмемо той же приклад з переказом грошей: в разі збою при виконанні другого запиту спрацює тригер, який виконає відкат або відправить повідомлення адміністратору бази даних.

Збережені процедури - це кілька SQL-команд, які зберігаються в базі даних під певним ім'ям і в сукупності виконують якусь функцію. За допомогою збережених процедур можна розширити синтаксис SQL так, що він буде схожий на звичайну мову програмування (наприклад, Oracle PL / SQL).

У нашому прикладі з переказом грошей два SQL-запити можна було б зберегти під одним ім'ям, а потім викликати цю процедуру, передавши їй в якості параметрів два номери рахунку і суму грошей. Тоді обидва запити виконувалися б в одній транзакції [39].

Вкладені запити - дозволяють підставляти значення в умови відбору динамічно, за результатами виконання іншого запиту. Наприклад, щоб дізнатися, який автомобіль перевіз вантажів більше, ніж в середньому по автопарку, потрібно зробити такий SQL-запит:

```
SELECT auto FROM autopark WHERE massa>! Більше чого ? Я поняття
не маю, яке середнє значення!
```

Для цього середнє значення в поле massa потрібно обчислити:

```
SELECT AVG (massa) FROM autopark
```

Якщо підтримуються вкладені запити, то ці два запити можна вкласти один в одного:

```
SELECT auto FROM autopark WHERE massa> (SELECT AVG (massa)
FROM autopark)
```

Але у випадку з MySQL середнє значення доводиться знаходити окремо і підставляти в інший запит безпосередньо в CGI-сценарії, що, безсумнівно, позначається на продуктивності.

Команда UNION - просто кажучи, вона об'єднує результати декількох запитів в один, з можливістю виключити дублікати рядків.

Каскадне оновлення даних - дозволяє видаляти і оновлювати пов'язані дані. Наприклад, при видаленні з бази даних запису про клієнта з пов'язаних таблиць автоматично видаляються всі записи про замовлення цього клієнта.

3 РОЗРОБКА САЙТІВ І БЛОГІВ НА ОСНОВІ PHP-MYSQL

3.1 Розробка сайтів з використанням технологій PHP-MYSQL

3.1.1 Складові частини сайту, розроблюваного з використанням PHP-MYSQL

В відповідності до технічного завдання розробці підлягає сайт з використанням браузера [46], веб-сервера Apache, інтерпретатора PHP та системи керування базами даних MySQL. На рис. 3.1 наведена загальна схема функціонування такого сайту.

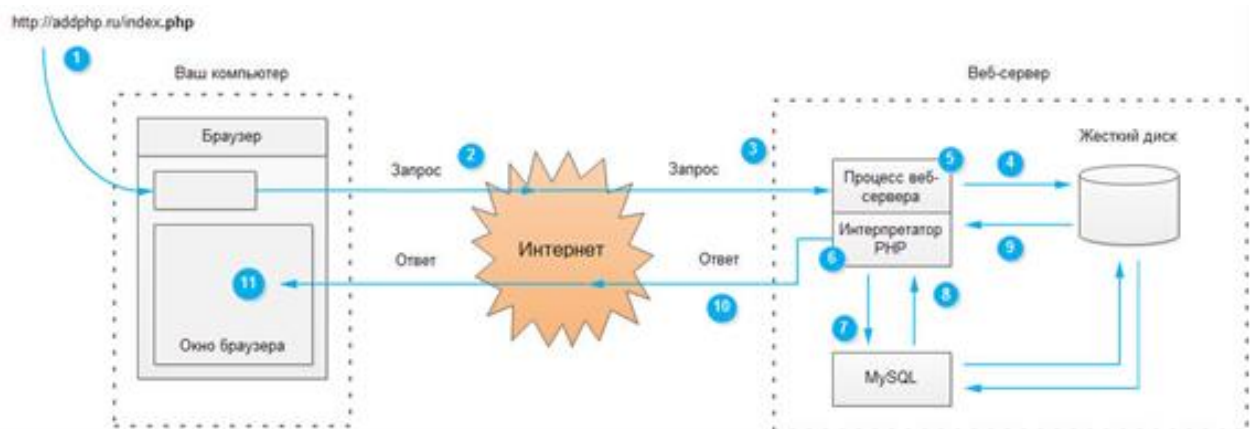


Рисунок 3.1 - Взаємодія компонентів сайту, розроблюваного з використанням PHP та MySQL

Основні складові дії функціонування сайту, позначені на рис. 3.1 відповідними номерами, виконуються в такій послідовності:

1. Користувач вводить адресу веб-сторінки, яка йому потрібна, в адресному рядку браузера.
2. Браузер розбиває адресу на складові і відправляє ім'я сторінки веб-серверу.
3. Процес веб-сервера на хості приймає запит на отримання сторінки index.php.
4. Веб-сервер зчитує файл index.php з жорсткого диска хоста.
5. Веб-сервер визначає, що це сценарій PHP, а не простий HTML-файл, і тому передає його на обробку іншому процесу - інтерпретатору PHP.
6. Інтерпретатор PHP виконує PHP-код, який він виявив у тексті, отриманому від процесу веб-сервера. Цей код включає в себе звернення до бази даних MySQL.
7. Інтерпретатор PHP запитує у процесу бази даних MySQL обробку звернень до бази даних.
8. Процес бази даних MySQL повертає результати запиту до бази даних.

9. Інтерпретатор PHP завершує виконання PHP-коду, додаючи дані, отримані з бази даних, і повертає результат процесу веб-сервера.
10. Веб-сервер повертає результат браузеру у вигляді HTML-тексту.
11. Веб-браузер формує зовнішній вигляд веб-сторінки на екрані комп'ютера користувача відповідно до отриманого HTML-тексту.

Може здатися, що кроків забагато, проте, все це виконується всякий раз, коли запитується веб-сторінка, що містить PHP-код. Фактично, описаний процес може неодноразово повторюватися для однієї веб-сторінки, якщо вона містить багато зображень, медіа-елементів, зовнішніх визначень CSS, файлів зі скриптами на мові JavaScript, оскільки кожне з них браузер отримує за допомогою окремого запиту до веб-сервера [23].

При розробці динамічних веб-сторінок доводиться працювати з великою кількістю змінних і компонентів сервера, що грають важливу роль в створенні привабливого, зручного для навігації і супроводу веб-сайту.

3.1.2 Проектування, визначення та наповнення бази даних сайту

Для розроблюваного сайту створюємо нову базу даних «mysite». В створену базу даних включаємо таблицю з назвою «pages». У цій таблиці будуть зберігатися статичні сторінки майбутнього сайту і інформація про них. Таблиця повинна містити наступні поля:

page_id - ідентифікатор сторінки (SMALLINT, primary key, auto_increment);

page_alias - псевдонім сторінки для рядка адреси ЧПУ (VARCHAR, 255);

page_title - назва сторінки у вікні браузера (VARCHAR, 255);

page_meta_d - опис сторінки для атрибуту description тегу <meta> (VARCHAR, 255);

page_meta_k – набір ключових слів для атрибуту keywords тегу <meta> (VARCHAR, 255);

page_h1 - заголовок сторінки (VARCHAR, 255);

page_s_desc - короткий опис матеріалу, наприклад якщо матеріали сайту будуть у вигляді блогу (TEXT);

page_content - основний текст сторінки, який буде виводитися в центральну колонку сайту (TEXT);

page_publish - містить «Y» - якщо сторінка опублікована, або «N» - якщо вона прихована (CHAR, за замовчуванням «Y»).

Відразу після створення таблиці вставляємо в неї значення для головної сторінки

сайту. В поле «page_alias» для головної сторінки вставляємо значення «home». Атрибутам тегу <meta> задаємо значення відповідно тематиці всього сайту. Таким же чином можна створювати інші сторінки, наприклад «Про компанію» з псевдонімом сторінки «about» і своїми значеннями атрибутів тегу <meta>, або «Контакти» з псевдонімом сторінки «contacts» і т. д.

3.1.3 Створення файлу core.php - ядра сайту

У кореневій папці сайту, яка повинна бути порожня на даному етапі, створюємо папку «cfg», в ній за допомогою .htaccess директивою «deny from all» закриваємо доступ до нього звичайним користувачам сайту, окрім адміністраторів. Створюємо файл ядра сайту core.php наступного змісту :

```
<?php
defined('INDEX') OR die('Прямий доступ до сторінки заборонений!');
// MYSQL
class MyDB {
    var $dblogin = "root"; // Ваш логін до бази даних
    var $dbpass = ""; // Ваш пароль до бази даних
    var $db = "mysite"; // Назва бази даних
    var $dbhost="localhost";
    var $link;
    var $query;
    var $err;
    var $result;
    var $data;
    var $fetch;
    function connect() {
        $this->link = mysql_connect($this->dbhost, $this->dblogin, $this->dbpass);
        mysql_select_db($this->db);
        mysql_query('SET NAMES utf8');
    }
    function close() {
        mysql_close($this->link);
    }
    function run($query) {
```

```

        $this->query = $query;
        $this->result = mysql_query($this->query, $this->link);
        $this->err = mysql_error();
    }
    function row() {
        $this->data = mysql_fetch_assoc($this->result);
    }
    function fetch() {
        while ($this->data = mysql_fetch_assoc($this->result)) {
            $this->fetch = $this->data;
            return $this->fetch;
        }
    }
    function stop() {
        unset($this->data);
        unset($this->result);
        unset($this->fetch);
        unset($this->err);
        unset($this->query);
    }
}
?>

```

Цей файл поки містить тільки простий клас підключення до бази даних, але в подальшому в нього будуть додаватись різні корисні функції, які будуть доступні з будь-якого місця в коді сайту. Файл ядра конфігурації створений.

3.1.4 Створення файлу `index.php` - головного контролера сайту

Тепер в кореневій папці сайту створюємо `index.php`, який буде основним скриптом сайту, свого роду головним контролером всього розроблюваного сайту. Вміст файлу `index.php` наступний :

```

<?php
    session_start();
    define("INDEX", "");

```



```

// Підключення створеного вище файлу ядра конфігурації core.php
require_once($_SERVER[DOCUMENT_ROOT]."/cfg/core.php");
// Підключення до БД
$db = new MyDB();
$db->connect();
// Головний контролер
switch ($_GET[$option]) {
    case "page":
        include($_SERVER[DOCUMENT_ROOT]."/com/page.php");
        break;
    default:
        include($_SERVER[DOCUMENT_ROOT]."/com/home.php");
        break;
}
include ($_SERVER[DOCUMENT_ROOT]."/template.php");
$db->close();
?>

```

Змінна `$_GET[$option]` буде вказувати головному контролеру який компонент сайту завантажувати при запиті. Зараз в сайті передбачено лише два компоненти: «сторінка» та «головна сторінка». Логіка роботи головного контролера така: з рядка URL отримується назва потрібного компонента, і в залежності від його значення підключається файл самого компонента (міститься в папці `/com`). Файл компонента виконує всі необхідні роботи, отримує інформацію з бази дані і записує її в змінні, для передачі в шаблон дизайну. В самому кінці підключається файл дизайну сайту, в який і передаються всі змінні і дані, отримані компонентами.

3.1.5 Створення компонентів виведення сторінок

У корені сайту створюємо папку «com» - в ній будуть зберігатися файли компонентів. Компонент сайту - це файл, в якому відбувається обробка даних для різних розділів сайту. Наприклад компонент звичайної сторінки отримує з бази даних назву, опис і текст матеріалу, і записує їх в змінні `$title`, `$meta_d`, `$meta_k`, `$content` і ін. Ці дані потім передаються в шаблон дизайну (під кожен компонент можна створювати свій шаблон дизайну) і виводяться користувачеві у вигляді HTML-сторінки. Наприклад, компонент каталогу, який можна створити в майбутньому, виконував би майже те ж саме, але з

даними про товари - а там своя специфіка, інші поля в таблиці, і т. д. Із-за цього для кожного функціонального розділу сайту варто створювати окремий компонент. У схемі MVC (Model-View-Controller) компонент виконує роль моделі.

3.1.5.1 Створення компонента виведення звичайної сторінки

В папці «com» створюємо файл «page.php». Вміст файлу наступний:

```
<?php
defined('INDEX') OR die('Прямий доступ до сторінки заборонено!');
/* Компонент сторінки */
$alias = $_GET[$page_alias];
$query = "SELECT * FROM pages WHERE page_alias=".$alias." AND page_publish='Y'
        LIMIT 1";
$db->run($query);
$db->row();
// Змінні компонента
$id = $db->data[page_id];
$alias = $db->data[page_alias];
$title = $db->data[page_title];
$h1 = $db->data[page_h1];
$meta_d = $db->data[page_meta_d];
$meta_k = $db->data[page_meta_k];
$s_desc = $db->data[page_s_desc];
$component = $db->data[page_content];
// Якщо сторінка не існує
if (!$id) {
    header("HTTP/1.1 404 Not Found");
    $component = "Помилка 404! Ця сторінка не існує";
}
$db->stop();
?>
```

3.1.5.2 Створення компонента виведення головної сторінки

Головна сторінка в базі даних зберігається під псевдонімом «home», і поки за своєю структурою не відрізняється від звичайних сторінок сайту - це просто стаття. Проте краще створити для неї окремий компонент.

Вміст компонента «home.php» в папці «com» майже збігається з вмістом компонента звичайної сторінки, за винятком рядка запиту до бази даних і назви компонента. Рядок запиту для головної сторінки повинен мати вигляд:

```
$query = "SELECT * FROM wx_pages WHERE page_alias = 'home' LIMIT 1";
```

3.1.6 Створення шаблону дизайну всього сайту

У корені сайту створюємо файл template.php. По суті це звичайний макет веб-дизайну в форматі HTML + CSS, тільки з PHP змінними в потрібних місцях. Між тегами <title> вставляємо <? = \$Title?>, в центральній колонці сайту вставляємо <? = \$Component?> і так по всьому шаблону розставляємо потрібні змінні, які оголошені в компонентах.

У кореневій папці також повинні бути папки «css» і «images» для елементів дизайну. У файлі /css/style.css стилі можна налаштувати на свій розсуд.

3.1.6.1 Чисті посилання і файл .htaccess

Для створення чистих посилань в файлі .htaccess використовуємо mod_rewrite з прямими вказівками правил для кожного компонента окремо. Вміст .htaccess на даного проекту повинен бути таким:

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteCond% {REQUEST_FILENAME}! -d
RewriteCond% {REQUEST_FILENAME}! -f
# ЗАБОРОНЕНІ ФАЙЛИ
RewriteRule .htaccess - [F]
RewriteRule template.php - [F]
# ПРАВИЛА mod_rewrite
RewriteRule page / ([a-z0-9_-]+) ([\ /] {0,1}) \. htm $ index.php? option = page &alias = $1[L]
```

3.2 Розробка блогів з використанням технологій PHP-MySQL

3.2.1 Створення бази даних блогу

Спочатку потрібно створити для розроблюваного блогу необхідну базу даних MySQL, наприклад, з ім'ям `sake_blog`. Після цього потрібно створимо єдину таблицю для статей блогу, виконавши наступні SQL-запити до створеної бази даних:

```

/* По-перше, створимо таблицю зі статтями */
CREATE TABLE articles (
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR (50),
    body TEXT,
    created DATETIME DEFAULT NULL,
    modified DATETIME DEFAULT NULL
);
/* Потім додаємо кілька статей для тестування */
INSERT INTO articles (title, body, created)
    VALUES ( 'The title', 'This is the article body.', NOW ());
INSERT INTO articles (title, body, created)
    VALUES ( 'A title once again', 'And the article body follows.', NOW ());
INSERT INTO articles (title, body, created)
    VALUES ( 'Title strikes back', 'This is really exciting! Not.', NOW ());

```

3.2.2 Створення контролера статей

Тепер створюємо контролер для статей. Контролер - це програма на мові PHP, яка реалізує всі дії (actions) зі статтями блогу. Розмістимо цей контролер в файлі `ArticlesController.php` і помістимо цей файл в папку `src/Controller`. Дії (actions) реалізують окремі функції або інтерфейси в програмі контролера. Наприклад, коли користувач запитує адресу `www.example.com/articles/index` (те ж саме, що і `www.example.com/articles/`), він очікує побачити список статей. Код цієї дії в контролері `ArticlesController.php` повинен виглядати наступним чином:

```

// src/Controller/ArticlesController.php
namespace App\Controller;

```

```

class ArticlesController extends ApplicationController {
    public function index () {
        $Articles = $this->Articles-> find( 'all');
        $this-> set(compact ( 'articles'));
    }
}

```

Єдина інструкція `set()` в дії передає дані від контролера в елемент відображення (view), створення якого описується нижче.

Нижче буде детально описаний процес програмування дій для відображення статей та додавання в блог нових статей. Решта дій блогу, такі як редагування статей, які вже є в блозі, видалення застарілих статей з блогу, включення коментарів до вже наявних статей, програмуються подібним же чином.

3.2.6 Відображення списку статей

Тепер, коли є дані, одержувані з моделі і логіка, визначена в контролері, створюємо елемент відображення для дії `index`, створеної вище.

Елементи відображення - це просто фрагменти, які розміщуються всередині шаблону для відображення даних створюваного додатку. Для більшості додатків, елементи відображення представляють собою HTML-код з включеннями інструкцій PHP, але вони можуть бути оформлені і як XML, CSV, і навіть виконавчі інструкції.

Візуальний шаблон - це код, який обертає собою всі елементи відображення. У додатку може бути визначено багато візуальних шаблонів, між якими можна переключатися.

Файли шаблонів зберігаються в папці `src/Template/Articles`. Код елемента відображення списку статей буде таким:

```

<!-- Файл: src/Template/Articles/index.ctp -->
<h1> Статті блогу </ h1>
<table>
    <tr>
        <th> Id </th>
        <th> Тема </th>
        <th> Дата створення </th>
    </tr>

```

```

<!-- Тут ми проходимся в циклі по об'єкту запиту $articles -->
<?php foreach ($articles as $article):?>
<tr>
    <td> <? = $ Article-> id?> </td>
    <td>
        <? = $this-> Html-> link ($ article-> title, [ 'action' => 'view', $ article-> id])?>
    </td>
    <td>
        <? = $ Article-> created-> format (DATE_RFC850)?>
    </td>
</tr>
<? Php endforeach; ?>
</table>

```

Коли прописуються URL-адреси, бажано використовувати формат масиву. Код дії відображення списку статей в контролері буде наступним:

```

// src/Controller/ArticlesController.php
namespace App\Controller;
class ArticlesController extends ApplicationController {
    public function index () {
        $this-> set( 'articles', $this-> Articles-> find ( 'all'));
    }
    public function view ($ id = null)
    {
        $Article = $this-> Articles-> get ($ id);
        $this-> set (compact ( 'article'));
    }
}

```

3.2.3 Додавання нових статей

Щоб можна було створювати нові статті і додавати їх в блог включаємо в контролер ArticlesController дію add():

```

// src/Controller/ArticlesController.php

```

```

namespace App\Controller;
use App\Controller\AppController;
class ArticlesController extends AppController {
    public function initialize() {
        parent :: initialize();
        $this->loadComponent ( 'Flash'); // Включення Flash-компонента
    }
    public function index() {
        $this-> set ( 'articles', $this-> Articles-> find ( 'all'));
    }
    public function view($id) {
        $Article = $this-> Articles-> get ($id);
        $this-> set (compact ( 'article'));
    }
    public function add () {
        $Article = $this-> Articles-> newEntity ();
        if ($this-> request-> is ( 'post')) {
            $Article = $this->Articles->patchEntity ($article, $this->request->getData());
            if ($this-> Articles-> save ($ article)) {
                $this-> Flash-> success (__ ('Ваша стаття була збережена.'));
                return $this-> redirect ([ 'action' => 'index']);
            }
            $this-> Flash-> error (__ ('Неможливо додати вашу статтю.'));
        }
        $this-> set ( 'article', $ article);
    }
}

```

Коли користувач відправляє дані форми методом POST, ця інформація стає доступною у властивості `$this->request->data`. Можна використовувати функцію `pr()` або `debug()` для виведення вмісту цієї властивості.

Елемент відображення для дії `add()` буде наступним:

```

<!-- Файл: src/Template/Articles/add.ctp -->
<h1> Add Article </ h1>

```

```

<?php
    echo $this->Form->create($ article);
    echo $this->Form->input( 'title');
    echo $this->Form->input( 'body', [ 'rows' => '3']);
    echo $this->Form->button(__ ( 'Зберегти статтю '));
    echo $this -> Form -> end();
?>

```

Тут використовуються методи-помічники для створення візуальних форм. Ось, наприклад, що згенерує метод-помічник `$this->Form->create()`:

```
<Form method = "post" action = "/articles/add">
```

Якщо `create()` викликається без додаткових аргументів, то передбачається, що створюється форма, яка відправляє дані методом POST до дії `add()` поточного контролера (або ж до дії `edit()`, якщо з даними передається параметр `id`). Метод `$this->Form->input()` використовується для створення полів введення форми з відповідними іменами. Метод `$this->Form->end()` закриває форму.

На рис. 3.2 представлений приклад відображення однієї зі статей, добавлених в розроблюваний блог.

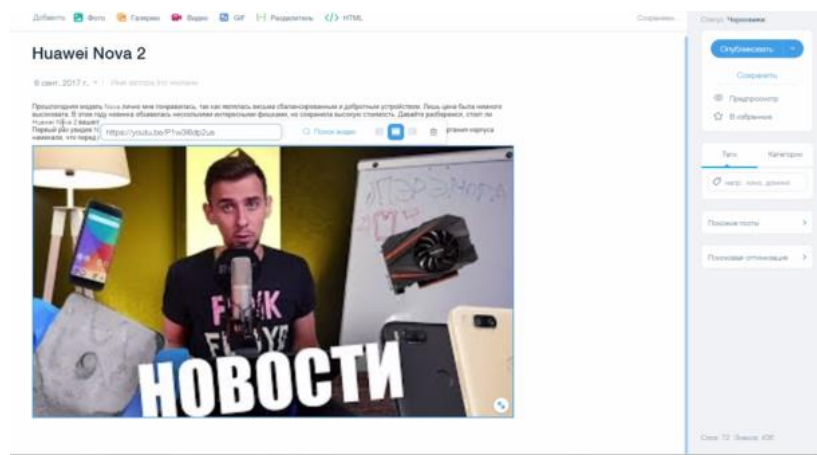


Рисунок 3. 2 – Представлення статей блогу на екрані комп'ютера

3.3 Створення сайту-блогу для ТСЦ «Шангир»

3.3.1. Мета і завдання створення сайту-блогу для ТСЦ «Шангир»

Підставою для створення власного веб-сайту для ТСЦ «Шангир» послужили наступні цілі:

- підтримка старих і створення нових зв'язків на ринку;
- забезпечення цілодобової підтримки клієнтів і партнерів;
- створення сприятливої думки про ТСЦ «Шангир».

Розроблюваний веб-сайт повинен вирішувати такі завдання:

- надавати інформацію про ТСЦ «Шангир», його діяльність, режим роботи, нові надходження, новини та ін.;
- забезпечити можливість пошуку по змісту сайту;
- забезпечити зв'язок користувача сайту з компанією.

3.3.2 Підготовка системи управління контентом Drupal для створення веб-сайту

Для роботи системи Drupal потрібні веб-сервер Apache з підтримкою мови програмування PHP версії 5.5 або вище і СКБД MySQL або PostgreSQL.

Для установки Drupal треба завантажити з офіційного веб-сайту (www.drupal.org) останню стабільну версію. В даному випадку була загружена система Drupal версії 7.37. Скачаний архів слід розпакувати в папку майбутнього веб-сайту. Наступною дією є створення бази даних для Drupal'у.

Для цього треба зайти в панель управління базами даних MySQL – phpMyAdmin (рис. 3.3).

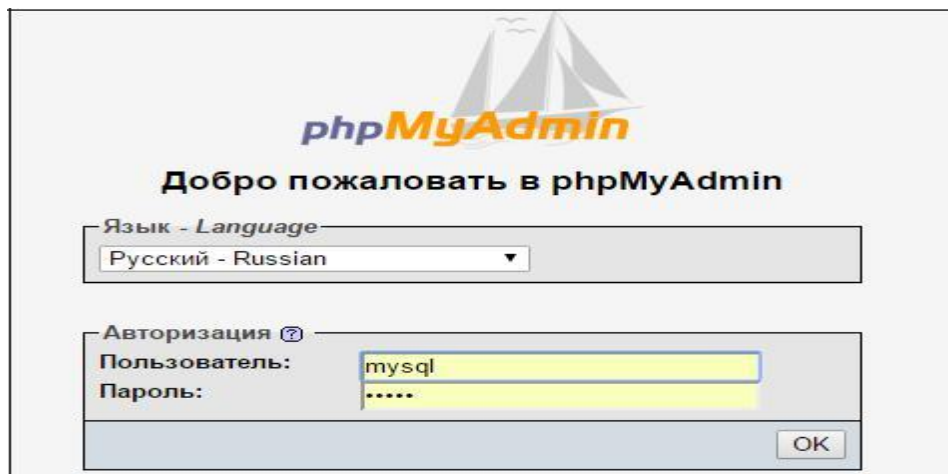


Рисунок 3.3 - Форма авторизації в phpMyAdmin

Після авторизації, в формі «Створити базу даних» слід ввести назву бази даних і вибрати кодування символів, яке буде використовуватися для порівняння рядків - «utf8_general_ci» (рис. 3.4).

Рисунок 3.4 - Форма створення нової бази даних.

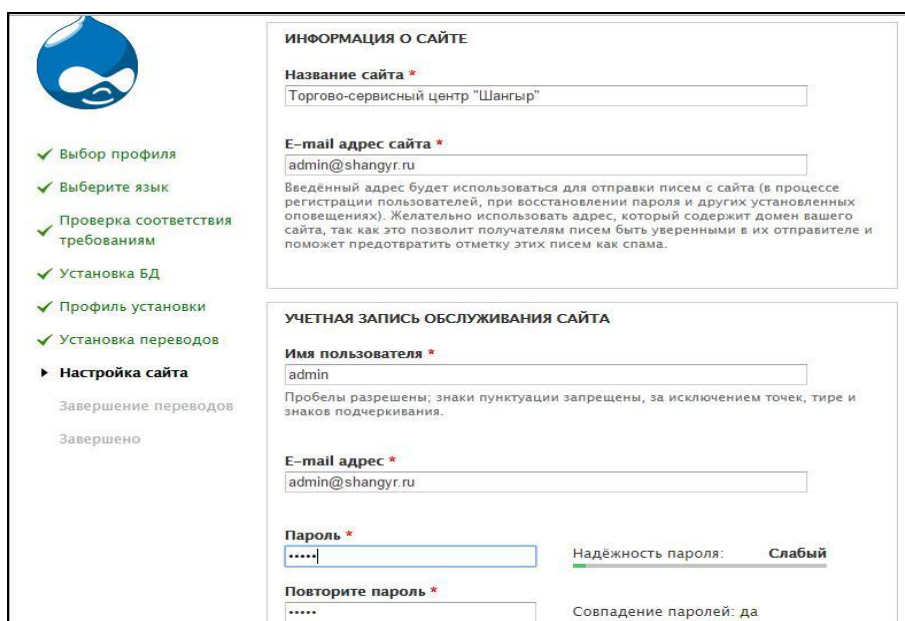
Далі треба перейти на веб-сторінку розроблюваного веб-сайту на локальному сервері. Для цього в браузері треба ввести локальну URL-адресу (в даній роботі використовується адреса «Shangyr.ru»), на якій відобразиться сторінка покрокової установки Drupal'у. У перших двох кроках вибирається стартовий набір функціоналу (мінімальний або стандартний) і обирається мова інтерфейсу сайту.

У третьому кроці вводяться налаштування підключення до раніше створеної бази даних. Для цього буде потрібно ім'я бази даних, логін і паролі для підключення до неї. Якщо сервер з базою даних знаходиться на іншій машині, то в розділі "Додаткові параметри" слід вказати налаштування з'єднання з віддаленою базою даних (рис. 3.5).

Рисунок 3.5 - Налаштування підключення до бази даних Drupal

Після налаштування підключення до бази даних, Drupal зробить наступну роботу: створить таблиці в базі даних, заповнить таблиці необхідними даними, оновить файлову структуру відповідно до обраного стартового функціоналу, створить файл «settings.php» за адресою «... / sites / default / ».

Останнім кроком в установці Dgupal, треба ввести назву веб-сайту, який в подальшому можна змінити через панель адміністратора, ввести логін і пароль головного адміністратора сайту і налаштувати тимчасові зони (рис. 3.6).



The screenshot shows the 'Настройка сайта' (Site Configuration) section of the Dgupal administration interface. On the left, a sidebar contains a list of configuration steps with green checkmarks: 'Выбор профиля', 'Выберите язык', 'Проверка соответствия требованиям', 'Установка БД', 'Профиль установки', and 'Установка переводов'. The 'Настройка сайта' step is highlighted with a blue arrow. Below this, there are two status indicators: 'Завершение переводов' and 'Завершено'. The main content area is titled 'ИНФОРМАЦИЯ О САЙТЕ' and contains the following fields:

- Имя сайта ***: Торгово-сервисный центр "Шангыр"
- E-mail адрес сайта ***: admin@shangyr.ru

Below these fields is a note: 'Введённый адрес будет использоваться для отправки писем с сайта (в процессе регистрации пользователей, при восстановлении пароля и других установленных оповещениях). Желательно использовать адрес, который содержит домен вашего сайта, так как это позволит получателям писем быть уверенными в их отправителе и поможет предотвратить отметку этих писем как спама.' Below this is the 'УЧЕТНАЯ ЗАПИСЬ ОБСЛУЖИВАНИЯ САЙТА' section:

- Имя пользователя ***: admin
- E-mail адрес ***: admin@shangyr.ru
- Пароль ***: [masked]
- Надёжность пароля:** Слабый
- Повторите пароль ***: [masked]
- Совпадение паролей:** да

Рисунок 3.6 - Налаштування сайту

Після введення всіх налаштувань натискаємо «Зберегти і продовжити», після чого буде запропоновано перейти на початкову сторінку сайту (рис. 3.7).

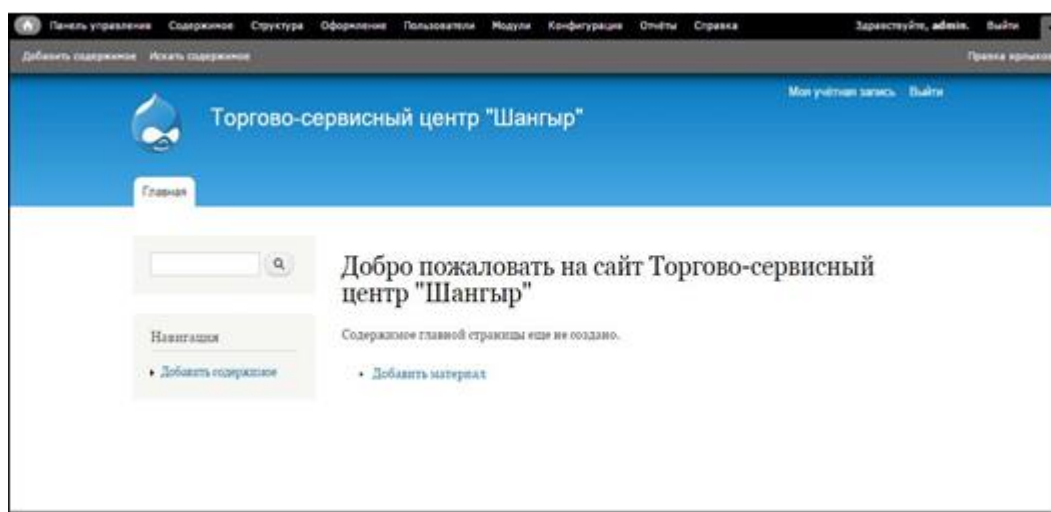


Рисунок 3.7 - Початкова сторінка сайту ТСЦ «Шангыр»

3.3.3 Дизайн веб-сайту

Під дизайном сайту розуміється не просто шаблон, а повне оформлення сторінок сайту в єдиному стилі. Найважливіше значення тут має головна сторінка. Вона відображає обличчя компанії, саме її бачать практично всі відвідувачі сайту, саме вона формує перше враження відвідувача про компанію.

Говорячи про візуальне подання веб-сайту, не можна не згадати про відмінність між фіксованими і динамічними сайтами. Сучасні монітори мають різні роздільні здатності. Монітор із роздільною здатністю по ширині в пікселях може бути 800, 1024, 1280, 1600 і навіть більше. Більш того, користувач може переглядати сайт в мінімізованому вікні.

Фіксований дизайн сайту передбачає одну єдину ширину сайту для всіх роздільних здатностей екрану і для всіх ширин вікон, тобто якщо ширина сайту 1000 пікселів, то на екрані 800 пікселів буде з'являтися горизонтальна прокрутка, а на екрані 1280 пікселів - порожні поля зліва і справа (або тільки з одного боку). Така побудова сайту дозволяє чітко управляти композицією сайту і однозначно знати, де буде кожен елемент навігації на кожному екрані. Цей тип сайту дозволяє використовувати більш насичені графічні елементи.

Динамічний дизайн, на відміну від фіксованого «підлаштовується» під ширину екрану. Це призводить до того, що сайт без горизонтальної прокрутки побачить більшість користувачів, але на різних екранах композиція сайту буде непередбачуваною.

Для досягнення поставлених цілей, було вирішено використовувати дизайн ділового бізнес-стилю. Даний дизайн є перевіреним і робочим інструментом. Такий дизайн гранично зрозумілий відвідувачам, тому що їм звично вже бачити такий стиль оформлення сайтів.

Даний ділової бізнес-стиль, який ще називають американським бізнес-стилем, передбачає наявність логотипу і контактів в верхній частині сайту, який називають «шапкою» сайту. Під «шапкою» розташовується меню навігації (головне меню) сайту, далі основна колонка, в якій може бути від однієї до трьох колонок, які мають у своєму розпорядженні в собі вміст поточної сторінки, додаткові меню і рекламні банери та інша інформація. Потім під основний колонкою поміщається «підвал» сайту, в якому зберігаються додаткові контактні дані, посилання на допоміжні ресурси, інформація про авторські права.

На підставі цих критеріїв була встановлена тема оформлення «Corporate Clean» з офіційного репозиторію Drupal на сайті drupal.org.

3.3.4. Створення сторінок веб-сайту

Для вирішення поставлених перед веб-сайтом задач, були створені наступні веб-сторінки сайту:

- Головна сторінка
- Про компанію
- Новини
- Вакансії
- Партнери
- Послуги
- Ремонт оргтехніки
- Заправка картриджів
- Ремонт комп'ютерної техніки
- Створення сайтів
- Статус замовлення
- Контакти

Головна сторінка (рис. 3.8) містить в собі презентацію з плавними переходами, в якій містяться фотографії та інші фото-ілюстративні матеріали. На слайдах презентації міститься короткий опис діяльності ТСЦ «Шангир» і послуги, що надаються.

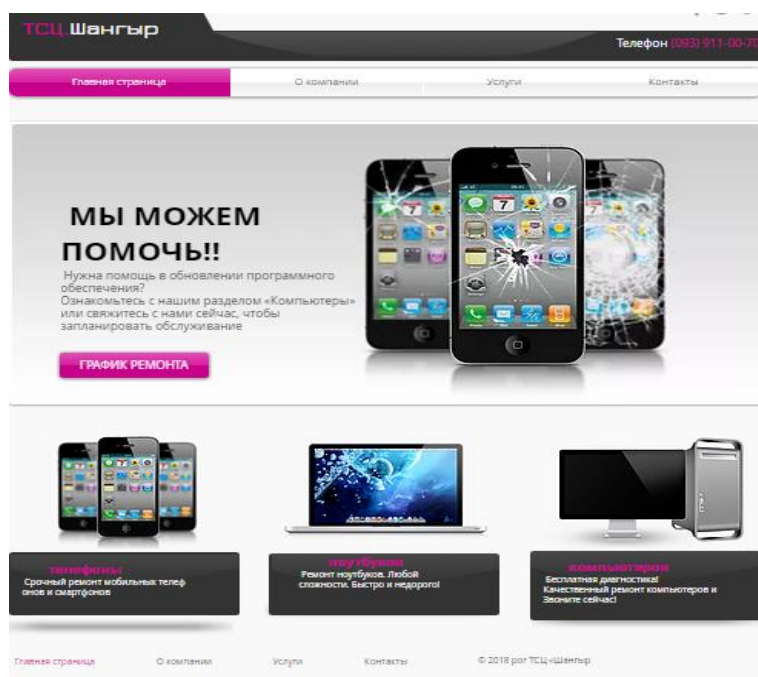


Рисунок 3.8 - Головна сторінка сайту

На сторінці «**Про компанію**» (рис. 3.9) сайту розміщено докладний опис діяльності

сервісного центру.

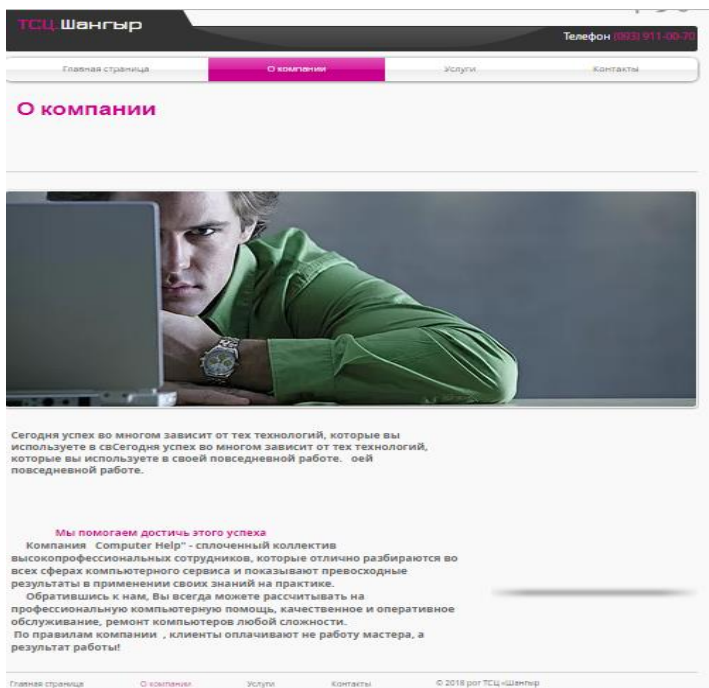


Рисунок 3.9 - Сторінка «Про компанію»

На сторінці «Новини» (рис. 3.10) розміщуються новини сервісного центру, відсортовані за спаданням дати створення новини.



Рисунок 3.10 - Сторінка «Новини»

На сторінці сайту «Вакансії» (рис. 3.11) показуються поточні відкриті вакансії в ТСЦ «Шангир».

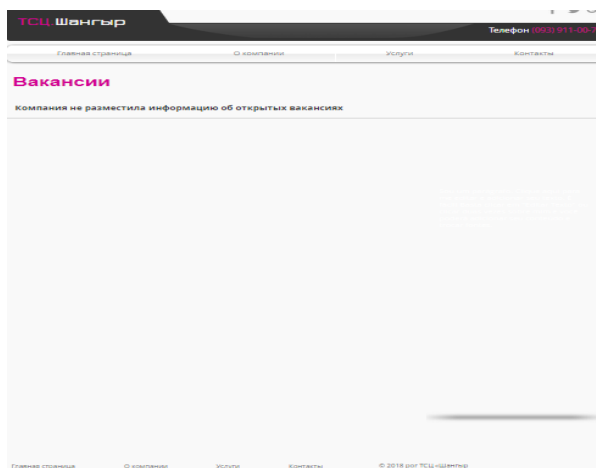


Рисунок 3.11 - Сторінка «Вакансії»

На сторінці «Партнери» (рис. 3.12) міститься інформація про партнерів ТСЦ «Шангир» і посилання на їх офіційні сайти.



Рисунок 3.12 - Сторінка «Партнери»

Сторінка «Послуги» (рис. 3.13) містить інформацію про послуги, які надаються ТСЦ «Шангир».

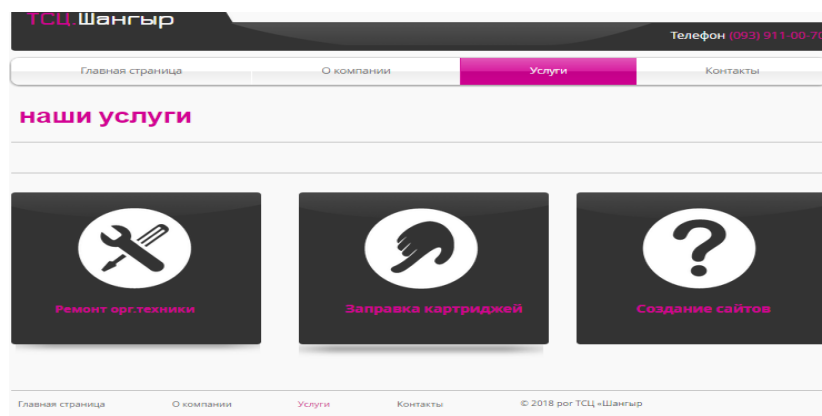


Рисунок 3.13 - Сторінка «Послуги»

Змістом сторінки «**Контакти**» (рис. 3.14) є контактна інформація (номер телефону, електронна пошта), адреса ТСЦ, схема проїзду і кнопка переміщення на форму зворотного зв'язку.

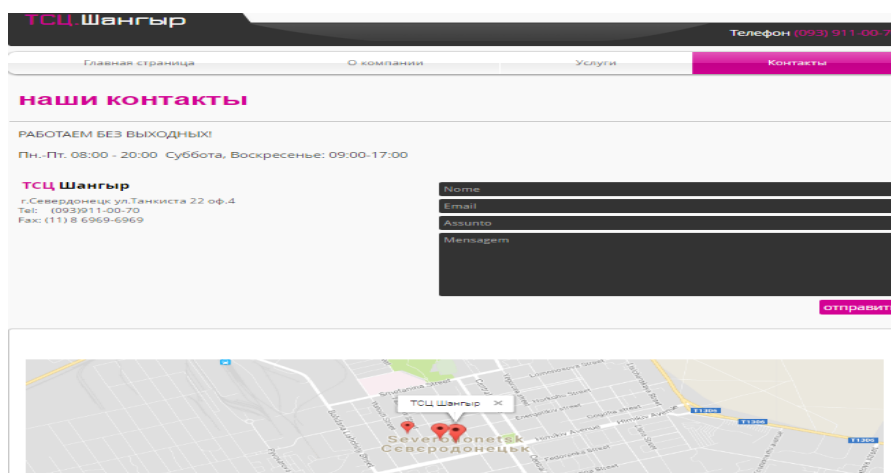


Рисунок 3.14 - Сторінка «Контакти»

3.3.5 Використані модулі CMS Drupal

Модулі - це розширення Drupal'у, які доповнюють його додатковим функціоналом. В розроблюваному сайті були використані наступні модулі CMS Drupal.

JQuery Update - модуль дозволяє оновлювати JQuery бібліотеку в Drupal. JQuery – JavaScript-бібліотека, використання якої робить розробку JavaScript коду набагато швидшою. JQuery дозволяє легко звертатися і маніпулювати з будь-яким елементом DOM (об'єктної моделі документа), полегшує роботу з подіями, легко здійснювати різні візуальні ефекти. Має величезну кількість JavaScript-плагінів, призначених для створення елементів призначених для користувача інтерфейсів.

Entity API - Модуль з API-інтерфейсом ядра Drupal, який забезпечує одноманітний доступ до об'єктів і їх властивостей.

Pathauto - модуль призначений для автоматичної генерації URL-адрес сутностей Drupal'у (матеріалів, таксономії, користувачів). Шаблон для генерації URL-адрес налаштовується адміністратором.

Token - модуль дозволяє використовувати в тексті змінні, в яких зберігається інформація про об'єкти Drupal'у.

Slider - Модуль дозволяє відобразити в шаблоні сайту слайд-шоу з плавними

ефектами переходу. Слайди додаються адміністратором зі спеціальної форми. Даний модуль залежить від модулів JQuery Update, Entity API, Pathauto і Token.

Contact - дозволяє розмістити на сайті форму зворотного зв'язку. Для роботи потрібно налаштувати параметри електронної пошти.

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) - повністю автоматизований публічний тест Тьюринга для розрізнення комп'ютерів і людей. Модуль потрібен для реєстрації користувача, відправлення форми зворотного зв'язку і додавання коментарів на сайті.

CKEditor - текстовий процесор, який використовується при створенні, редагуванні контенту сайту.

3.3.6 Використані сторонні модулі

uLogin – модуль, розроблений компанією uLogin та призначений для інтеграції з сервісом uLogin для авторизації на веб-сайті за допомогою соціальних мереж і популярних порталів (Google, Яндекс, Mail.ru, VK, Facebook та ін.). Даний модуль дозволяє користувачам не реєструватися зайвий раз на сайтах, а використовувати для цього свої облікові записи популярних веб-порталів (рис. 3.15).

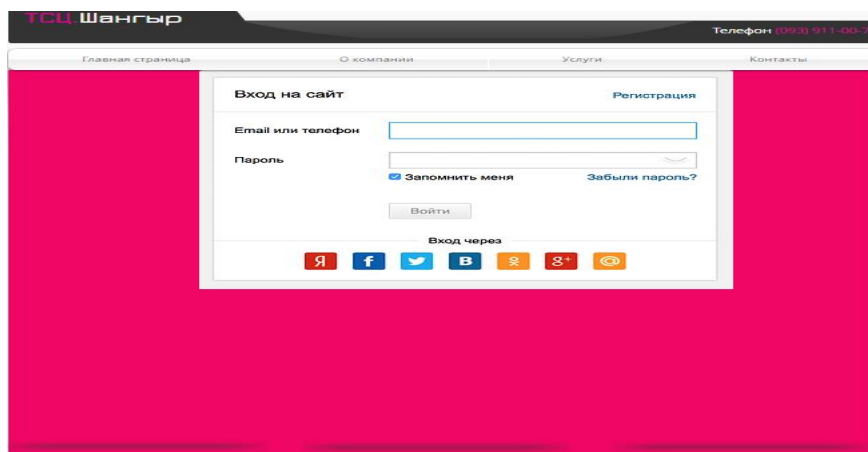


Рисунок 3.15 - Форма авторизації для входу на сайт ТСЦ «Шангыр», розроблена за допомогою модуля uLogin

JivoSite - модуль, розроблений компанією JivoSite, надає відвідувачам сайту можливість спілкуватися зі співробітниками ТСЦ «Шангыр» в реальному часі. Відвідувачам надається форма зворотного зв'язку, якщо на даний момент часу на зв'язку немає співробітників сервісного центру.

Для зв'язку з відвідувачами співробітнику необхідно запустити спеціальне

програмне забезпечення, що надається JivoSite. Програма (рис. 3.16) має зручний інтерфейс, забезпечує історію листування з відвідувачами, має інтерактивну підказку, присутня можливість голосового спілкування з відвідувачем сайту.

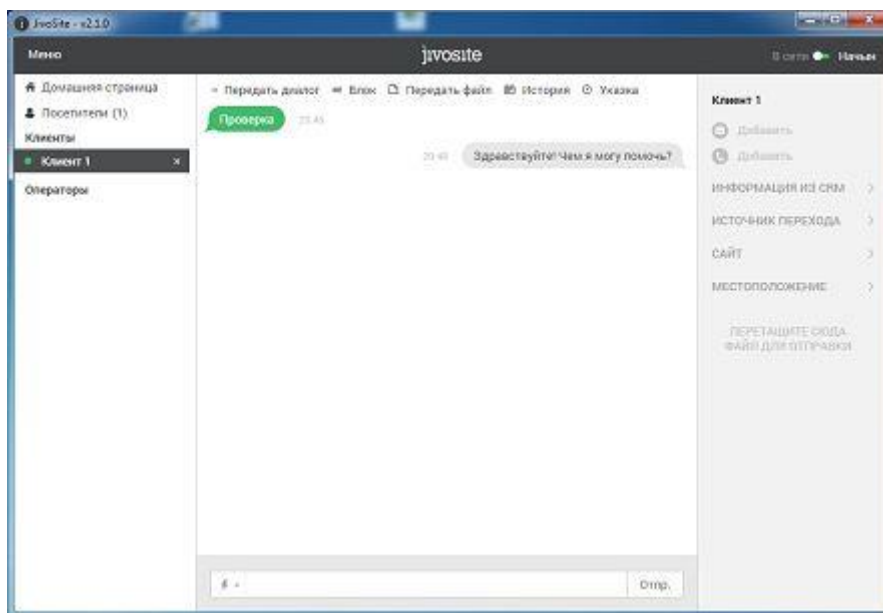


Рисунок 3.16 - Интерфейс програми JivoSite

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ

4.1 Вступ

Охорона праці – це система законодавчих актів, соціально-економічних, організаційних, технічних і лікувально-профілактичних заходів і засобів, що забезпечують безпеку, збереження здоров'я і працездатність людини в процесі праці.

Охорона праці виявляє і вивчає можливі причини виробничих нещасливих випадків, професійних захворювань, аварій, вибухів, пожеж і розробляє систему заходів і вимог з метою усунення цих причин, і створення безпечних і сприятливих для людини умов праці [50].

З питаннями охорони праці нерозривно пов'язане і вирішення питань охорони природи. Складність задач, які стоять перед охороною праці, вимагає використання досягнень і висновків багатьох наукових дисциплін, прямо або побічно пов'язаних із задачами створення здорових і безпечних умов праці.

Оскільки головним об'єктом охорони праці є людина в процесі праці, то при розробці вимог виробничої санітарії використовуються результати досліджень ряду медичних і біологічних дисциплін.

Особливо тісний зв'язок існує між охороною праці, науковою організацією праці, ергономікою, інженерною психологією і технічною естетикою.

Успіх у вирішенні проблем охорони праці здебільшого залежить від якості підготовки фахівців у цій області, від їхнього уміння приймати правильні рішення в складних і мінливих умовах сучасного виробництва [51].

4.2 Правила охорони праці під час експлуатації електронно-обчислювальних машин

Правила охорони праці під час експлуатації електронно-обчислювальних машин поширюються на всі підприємства, установи, організації, юридичні особи, незалежно від форми власності, відомчої належності, видів діяльності та на фізичних осіб, які здійснюють розробку, виробництво та застосування електронно-обчислювальних машин і персональних комп'ютерів (далі – ЕОМ), у тому числі й на тих, які мають робочі місця, обладнані ЕОМ, або виконують обслуговування, ремонт та налагодження ЕОМ [52].

Загальні вимоги до виробничих приміщень:

Облаштування робочих місць, обладнаних відеотерміналами, повинно забезпечувати:

– належні умови освітлення приміщення і робочого місця, відсутність відблисків;

- оптимальні параметри мікроклімату;
- наявність шуму та вібрації;
- м'яке рентгенівське випромінювання;
- електромагнітне випромінювання;
- ультрафіолетове та інфрачервоне випромінювання;
- електростатичне поле між екраном і оператором;

Для всіх споруд і приміщень, в яких експлуатуються відеотермінали та ЕОМ, повинна бути визначена категорія з вибухопожежної і пожежної безпеки відповідно до НАПБ Б.03.002-2007 «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою».

Виробничі приміщення, в яких розташовані ЕОМ, не повинні межувати з приміщеннями, де рівні шуму та вібрації перевищують норму (механічні цехи, майстерні тощо). Робочі місця з відеотерміналами або персональними ЕОМ у приміщеннях з джерелами шкідливих виробничих факторів повинні розміщуватися в ізольованих кабінах з обладнаним повітрообміном. Площу приміщень, в яких розташовують відеотермінали, визначають згідно з чинними нормативними документами з розрахунку на одне робоче місце, обладнане відеотерміналом: площа – не менше $6,0 \text{ м}^2$, обсяг – не менше $20,0 \text{ м}^3$, з урахуванням максимальної кількості осіб, які одночасно працюють у зміні [53].

Стіни, стеля, підлога приміщень, де розміщені ЕОМ, повинні виготовлятися з матеріалів, дозволених для оздоблення приміщень органами державного санітарно-епідеміологічного нагляду.

Обслуговування, ремонт та налагодження ЕОМ, вузлів та блоків ЕОМ слід виконувати в окремому приміщенні (майстерні), які можуть передбачити можливість вологого очищення поверхонь комунікацій та опалювальних приладів.

Підлога всієї зони обслуговування, ремонту та налагодження ЕОМ, вузлів та блоків ЕОМ має бути вкрита діелектричними килимками, термін використання яких після їх випробування на електричну міцність не закінчився, або викладена ізолювальними підстилками (шириною не менше ніж $0,75\text{-}0,8 \text{ м}$) для ніг.

Приміщення комп'ютерних класів (залів), в яких проводиться навчання на ЕОМ повинні мати суміжне приміщення (лаборантську) площею не менше 18 м^2 з двома входами: в учбове приміщення та в коридор (на сходову клітку).

Заземлені конструкції, що знаходяться в приміщеннях (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном тощо), мають бути надійно захищені діелектричними щитками або сітками від випадкового дотику.

У приміщеннях з ЕОМ слід щоденно проводити вологе прибирання. У приміщеннях з

ЕОМ повинні бути медичні аптечки першої допомоги.

Приміщення, в яких розміщуються ЕОМ типу ЕС, СМ та інші великі ЕОМ загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежегасіння відповідно до вимог Переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежегасіння та пожежної сигналізації, СНиП 2.04.09-84, ДБН В.2.5-13-98 Пожежна автоматика будинків та споруд, Правил пожежної безпеки в Україні та вимог нормативно-технічної та експлуатаційної документації заводу-виробника. Підходи до засобів пожежегасіння повинні бути вільними.

Санітарно-гігієнічні вимоги:

Умови праці осіб, які працюють з ЕОМ, повинні відповідати I або II класу згідно з Гігієнічною класифікацією праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу № 4137-86, затвердженою МОЗ України 12.08.86.

Вимоги до освітлення:

Приміщення з ЕОМ повинні мати природне і штучне освітлення відповідно до ДБН В.2.5-28:2015 «Природне і штучне освітлення». Природне світло повинно проникати через бічні світло прорізи, зорієнтовані, як правило, на північ чи північний схід, і забезпечувати коефіцієнт природної освітленості (КПО) не нижче 1,5%. Розрахунки КПО проводяться відповідно до ДБН В.2.5-28:2015. Вікна приміщень з відео терміналами повинні мати регульовальні пристрої для відкривання, а також жалюзі, штори, зовнішні козирки тощо.

Штучне освітлення приміщення з робочими місцями, обладнаними відеотерміналами ЕОМ загального та персонального користування, має бути обладнане системою загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, де переважають роботи з документами, допускається вживати систему комбінованого освітлення (додатково до загального освітлення встановлюються світильники місцевого освітлення). Відношення яскравості екрану комп'ютера до яскравості оточуючих його поверхонь не повинно перевищувати у робочій зоні 3:1.

При розташуванні відео терміналів ЕОМ за периметром приміщення лінії світильників штучного освітлення повинні розміщуватися локально над робочими місцями.

Для забезпечення нормованих значень освітлення в приміщеннях з відеотерміналами ЕОМ загального та персонального користування необхідно очищати віконне скло та світильники не рідше ніж 2 рази на рік, та своєчасно, проводити заміну ламп, що перегоріли.

Таблиця 4.1

Норми освітленості в кабінетах і класах з ПК

Характеристика роботи	Робоча поверхня	Плосщина	Освітленість, лк	Примітка
Робота переважно з екранами дисплеїв ПК (50 % та більше робочого часу)	Екран	В	200	не вище
	Клаватура	Г	400	не нижче
	Стіл	Г	400	не нижче
Робота переважно з документами (з екранами дисплеїв ПК менше 50 % робочого часу)	Екран	В	200	не вище
	Клаватура	Г	400	не нижче
	Стіл	Г	500	не нижче
	Дошка	В	500	не нижче
Проходи основні	Підлога	Г	100	

Примітка: В - вертикальна площина, Г - горизонтальна площина. Загальне освітлення має бути виконане у вигляді суцільних або переривчатих ліній світильників, що розміщуються збоку від робочих місць (переважно зліва) паралельно лінії зору працівників. Допускається застосовувати світильники таких класів світлорозподілу:

- світильники прямого світла — П;
- переважно прямого світла — Н;
- переважно відбитого світла — В.

Вимоги до рівнів шуму та вібрації.

У приміщеннях з ЕОМ рівні звукового тиску, рівні звуку та еквівалентні рівні звуку на робочих місцях повинні відповідати вимогам ГОСТ 12.1.003 ССБТ «Шум. Общие требования безопасности», ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку», затверджених Міністерством охорони здоров'я СРСР, ГР N

2411-81 «Гігієнічні рекомендації по встановленню рівнів шуму на робочих місцях з урахуванням напруженості та тяжкості праці», затверджених Міністерством охорони здоров'я України.

Рівні вібрації під час виконання робіт з ЕОМ у виробничих приміщеннях не повинні перевищувати допустимих значень, визначених в ДСН 3.3.6.039-99 «Санітарні норми виробничої загальної та локальної вібрації», затверджених Міністерством охорони здоров'я СРСР, та ДСанПіН 3.3.2-007-98.

Вимоги до вентиляції, опалення та кондиціонування, мікроклімату.

Приміщення з ЕОМ повинні бути обладнані системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією відповідно до ДБНВ.2.5-67:2013 «Опалення, вентиляція та кондиціонування» [55]. Параметри мікроклімату, іонного складу повітря, вміст шкідливих речовин на робочих місцях, оснащених відеотерміналами, повинні відповідати вимогам пункту 2.4 ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень», затверджених Міністерством охорони здоров'я СРСР, ГОСТ 12.1.005-88 «ССБТ Общие санитарно-гигиенические требования к воздуху рабочей зоны» (табл. 4.2), СН 2152-80 «Санітарно-гігієнічні норми допустимих рівнів іонізації повітря виробничих та громадських приміщень» (таблиця 4.3).

Таблиця 4.2

Нормовані параметри мікроклімату для приміщень з ВДТ та ПЕОМ

П ора року	Категорія робіт згідно ГОСТ 12.1- 005-88	Темпер атура повітря, град.С,	Відносн а вологість повітря, %	Шкідливість руху повітря, м/с
		оптима льна	Оптима льна	Оптимальна
Х олодна	легка -1 а	22 – 24	40 – 60	0,1
	легка -1 б	21 – 23	40 – 60	0,1
Те пла	легка -1 а	23 – 25	40 – 60	0,1
	легка -1 б	22 – 24	40 – 60	0,2

Таблиця 4.3

Рівні іонізації повітря приміщень при роботі на ВДТ та ПЕОМ (відповідно до СН 2152-80)

Рівні	Кількість іонів в 1 см куб. повітря	
	+	-
	N	n
Мінімально необхідне	400	600
Оптимальні	1500 – 3000	3000 – 5000
Максимально допустимі	50000	50000

Вимоги щодо рівня неіонізуючих електромагнітних випромінювань, електростатичних та магнітних полів.

Гранично допустима напруженість електростатичного поля на робочих місцях не повинна перевищувати рівнів, наведених в ГОСТ 12.1.045 «ССБТ. Електромагнітні поля. Допустимі рівні на робочих місцях і вимоги до проведення контролю», НПАОП 0.00.-1.28-10. Правила охорони праці під час експлуатації електронно-обчислювальних машин. Електробезпека.

Потужність експозиційної дози рентгенівського випромінювання на відстані 0,05 м від екрана та корпусу відео терміналу при будь-яких положеннях регулювальних пристроїв відповідно до Норм радіаційної безпеки України (НРБУ-97), затверджених постановою державного санітарного лікаря Міністерства охорони здоров'я України від 18.08.97 № 58, не повинна перевищувати $7,74 \times 10^{-12}$ А/кГ, що відповідає еквівалентній дозі 0,1 мбер/год. (100 мкР/год.).

Вимоги електробезпеки.

Під час проектування систем електропостачання, монтажу силового електрообладнання та електричного освітлення будівель та приміщень для ЕОМ необхідно дотримуватись вимог ПВЕ, ПТЕ, ПБЕ, ДБНВ.2.5-28:2015 «Природне і штучне освітлення», затверджених Держбудом, ГОСТ 12.1.006, ГОСТ 12.1.030 «ССБТ. Електробезопасность. Защитное заземление, зануление», НПАОП 40.1-1.01-97 «Правила безопасной эксплуатации электроустановок », ГОСТ 12.1.045, ДБН В.2.5-23-2003 Держкомархитектуры «Проектирование электрообладнання об'єктів цивільного призначення », Правил пожежної безпеки в Україні, цих Правил, а також розділів СНиП, що стосуються штучного освітлення і електротехнічних пристроїв, та вимог нормативно-технічної і експлуатаційної документації заводу-виробника ЕОМ.

У приміщенні, де одночасно експлуатується або обслуговується більше п'яти персональних ЕОМ, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

ЕОМ, периферійні пристрої ЕОМ та устаткування для обслуговування ремонту та

налагодження ЕОМ повинні підключатися до електромережі тільки з допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. Неприпустимим є підключення ЕОМ, периферійних пристроїв ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв. Електромережу штепсельних розеток для живлення персональних ЕОМ, периферійних пристроїв ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ при розташуванні їх уздовж стін приміщення прокладають по підлозі поряд зі стінами приміщення, як правило, в металевих трубах і гнучких металевих рукавах з відводами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання.

Вимоги до обладнання.

Відеотермінали, ЕОМ, ПЕОМ, спеціальні периферійні пристрої ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ повинні відповідати вимогам чинних в Україні стандартів, нормативних актів з охорони праці та цих Правил. Відеотермінали, ЕОМ, ПЕОМ, спеціальні периферійні пристрої ЕОМ закордонного виробництва додатково повинні відповідати вимогам національних стандартів держав–виробників і мати відповідну позначку на корпусі, в паспорті або іншій експлуатаційній документації.

Вимоги до організації робочого місця користувача ЕОМ.

Організація робочого місця користувача відеотерміналу та ЕОМ повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ГОСТ 12.2.032 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования»; характеру та особливостям трудової діяльності [56].

Площа, виділена для одного робочого місця з відеотерміналом або персональною ЕОМ, повинна складати не менше 6 м^2 , а обсяг – не менше 20 м^3 . Робочі місця з відеотерміналами відносно світлових прорізів повинні розміщуватися так, щоб природне світло падало збоку, переважно зліва.

При розміщенні робочих місць з відеотерміналами та персональними ЕОМ необхідно дотримуватись таких вимог:

- робочі місця з відеотерміналами та персональними ЕОМ розміщуються на відстані не менше 1 м від стін зі світловими прорізами;
- відстань між бічними поверхнями відеотерміналів має бути не меншою за 1,2 м;
- відстань між тильною поверхнею одного відео терміналу та екраном іншого не повинна бути меншою 2,5 м;
- прохід між рядами робочих місць має бути не меншим 1 м.

Конструкція робочого місця користувача відеотермінала (при роботі сидячи) має забезпечувати підтримання оптимальної робочої пози з такими ергономічними характеристиками: ступні ніг – на підлозі або на підставці для ніг; стегна – в горизонтальній площині; передпліччя – вертикально; лікті – під кутом $70-90^\circ$ до вертикальної площини; зап'ястя зігнуті під кутом не більше 20° відносно горизонтальної площини, нахил голови $-15-20^\circ$ відносно вертикальної площини.

Висота робочої поверхні столу для відеотермінала має бути в межах 680–800 мм, а ширина – забезпечувати можливість виконання операцій в зоні досяжності моторного поля. Рекомендовані розміри столу: висота – 725 мм, ширина – 600–1400 мм, глибина – 800–1000 мм.

Для зниження статичного напруження м'язів рук необхідно застосовувати стаціонарні або знімні підлокітники довжиною не менше 250 мм, шириною – 50 – 70 мм, що регулюються по висоті над сидінням у межах 230 ± 30 мм та по відстані між підлокітниками в межах 350–500 мм.

Екран відеотермінала та клавіатура мають розташовуватися на оптимальній відстані від очей користувача, але не ближче 600 мм, з урахуванням розміру алфавітно-цифрових знаків та символів. Відстань від екрана до ока працівника повинна складати:

- при розмірі екрану по діагоналі 35/38 см (14" /15") – 600 – 700 мм
- при розмірі екрану по діагоналі 43 см (17") – 700 – 800 мм
- при розмірі екрану по діагоналі 48 см (19") – 800 – 900 мм
- при розмірі екрану по діагоналі 53 см (21") – 900 – 1000 мм

Клавіатуру слід розміщувати на поверхні столу або на спеціальній, регульованій за висотою, робочій поверхні окремо від столу на відстані 100–300 мм від краю, ближчого до працівника. Кут нахилу клавіатури має бути в межах 5–15'.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість екрану відеотермінала, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля: по висоті 900 – 1300 мм, по глибині 400 – 500 мм.

При потребі високої концентрації уваги під час виконання робіт з високим рівнем напруженості суміжні робочі місця з відеотерміналами та персональними ЕОМ необхідно відділяти одне від одного перегородками висотою 1,5 – 2 м.

Вимоги безпеки під час експлуатації ЕОМ.

Користувачі ЕОМ повинні слідкувати за тим, щоб відеотермінали, ЕОМ, периферійні пристрої ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ були справними і випробуваними відповідно до чинних нормативних документів. Щоденно перед початком роботи необхідно проводити очищення

екрана відеотерміналу від пилу та інших забруднень. Після закінчення роботи відеотермінал та персональна ЕОМ повинні бути відключені від електричної мережі. У разі виникнення аварійної ситуації необхідно негайно відключити відеотермінал та ЕОМ від електричної мережі. При використанні з ЕОМ та відеотерміналами лазерних принтерів потрібно дотримуватись вимог Санітарних норм та правил устрою та експлуатації лазерів № 5804–91, затверджених Міністерством охорони здоров'я в 1991 р.

Розрахунок штучного освітлення.

Рациональне освітлення виробничих ділянок є одним з найважливіших факторів попередження травматизму і професійних захворювань. Правильно організоване висвітлення створює сприятливі умови праці, підвищує працездатність і продуктивність праці. Освітленість на робочому місці повинна бути такою, щоб працюючий міг без напруги зору виконувати свою роботу при припустимому з народногосподарської точки зору витратою засобів, матеріалів і електроенергії [54, 57].

Приведемо розрахунок штучного освітлення для приміщення, розміри якого: довжина 5м, ширина 3м, висота 3м.

Розрахунок освітленості виконаємо методом коефіцієнта використання. Цей метод використовується для розрахунку загального рівномірного висвітлення горизонтальних поверхонь виробничих приміщень при відсутності затемнень.

Розрахунок висвітлення методом коефіцієнта використання виконується по формулі:

$$\Phi = \frac{E \times S \times k \times z}{N \times \eta}, \quad (4.1)$$

де Φ – необхідний світловий потік ламп у кожному світильнику, лм; E – нормативна мінімальна освітленість, лк, з довідкової літератури [53]; k – коефіцієнт запасу, вибирається з довідкової літератури [55, 56]; S – освітлювана площа, кв м; z – коефіцієнт мінімальної освітленості, величина якого знаходиться в межах від 1,1 до 1,5 (при оптимальних відносинах відстані між світильниками до розрахункової висоти для ламп розжарювання і ДРЛ $z=1,15$ і для люмінесцентних ламп $z = 1,1$); N – число світильників у приміщенні; η – коефіцієнт використання світлового потоку.

Приймаємо: $E=400$ лк; $k=1,5$; $z=1,1$

Освітлювана площа приміщення визначається по формулі:

$$S = A \times B, \quad (4.2)$$

де S – освітлювана площа, кв м; A – довжина приміщення, м; B – ширина приміщення, м.

Отримуємо $S = 5 \times 3 = 15 \text{ м}^2$.

Розміщення світильників у приміщенні при системі загального висвітлення залежить від розрахованої висоти їхнього підвісу h , що звичайно задається розмірами приміщень.

Найбільш вигідне співвідношення відстані між світильниками до розрахункової висоти підвісу:

$$\lambda = \frac{L}{h}, \quad (4.3)$$

приймається по таблиці в довідковій літературі [53], у залежності від типової кривої сили світла світильника. Для люмінесцентних ламп при косинусоїдальній типової кривої вибираємо $a = 1,4$.

Знаходимо розрахункову висоту підвісу по наступній формулі:

$$h = H - h_{\%} - h_p, \quad (4.5)$$

де H – висота приміщення, м; $h_{\%}$ – висота звису світильника (від перекриття), м; h_p – висота робочої поверхні над підлогою, м.

Приймаємо: $H=3,9$ м, $h_{\%}=0,7$ м, $h_p=0,8$ м. Тоді $h=3,9-0,7-0,8=1,4$ м.

Відстань між світильниками визначаємо з формули (4.5):

$$L = \lambda \times h, \quad L = 1,4 \times 0,4 = 2,5 \text{ м} \quad (4.6)$$

Визначаємо кількість світильників для установки в приміщенні:

$$N = \frac{S}{L^2}, \quad N = \frac{15}{0,96^2} = 1,6 \approx 2, \text{ шт.} \quad (4.7)$$

Для визначення коефіцієнта використання η знаходимо індекс приміщення i :

$$i = \frac{A \times B}{h \cdot (A + B)}, \quad (4.8)$$

де A і B – відповідно довжина і ширина приміщення, м; h – розрахункова висота підвісу, м. Отримуємо

$$i = \frac{5 \times 3}{2,4 \times (5 + 3)} = 0,78.$$

Отримане значення i округляємо до найближчого табличного значення і приймаємо $i = 0,78$.

Оцінюємо коефіцієнти відображення поверхонь приміщення: стелі – ρ_i , стін – ρ_n , робочої поверхні – ρ_p . Приймаємо: $\rho_i = 70\%$, $\rho_n = 50\%$, $\rho_p = 30\%$.

За отриманим значенням i і ρ визначаємо величину коефіцієнта використання світлового потоку для обраного світильника [56].

Вибираємо світильник типу ПВЛМ–Д, для якого $\eta=73\%$. По формулі (4.1) визначаємо необхідний світловий потік ламп у кожному світильнику:

$$\Phi = \frac{400 \times 15 \times 1.5 \times 1.1}{2 \times 0.73} = 6781 \text{ лм.}$$

З довідкової літератури вибираємо необхідну лампу. Тип обраної лампи – ЛД-65 – 3570 лм.. У світильнику будуть установлені дві таких лампи.

4.3 Пожежна безпека

Пожежна безпека – це стан об’єкту, при якому виключається можливість пожежі, а у випадку її виникнення виключається дія на людей небезпечних факторів пожежі та забезпечується захист матеріальних цінностей.

Відповідно до Закону України «Про пожежну безпеку» [58] пожежна безпека повинна забезпечуватися шляхом проведення організаційних, технічних та інших заходів, спрямованих на попередження пожеж, забезпечення безпеки людей, зниження можливих майнових втрат і зменшення негативних екологічних наслідків у разі їх виникнення, створення умов для швидкого виклику пожежних підрозділів та успішного гасіння пожеж.

Система активного пожежного захисту – це комплекс організаційних заходів і технічних засобів по боротьбі з пожежами і запобіганню дії на людей небезпечних чинників пожежі, а також обмеження матеріальних збитків від неї.

Основний напрямок діяльності пожежної охорони полягає у профілактиці пожеж і в обмеженні їх розмірів. Пожежі або вибухи на підприємствах завдають великих матеріальних збитків і майже завжди супроводжуються нещасними випадками з людьми (опіки, поранення) тощо.

Контроль за дотриманням затверджених відповідно до діючого законодавства норм і правил пожежної безпеки на об’єктах народного господарства здійснюють органи Державного пожежного нагляду (ДНП) управління пожежної охорони Міністерства надзвичайних ситуацій України.

Відповідальність за прийняття протипожежних заходів в організаціях покладається персонально на їх керівників без права передовіряти цю відповідальність іншим, підлеглим їм особам, вони здійснюють загальне керівництво роботою в галузі пожежної безпеки підприємств і організацій.

На (підприємстві) встановлюються первинні засоби пожежегасіння відповідно до вимог Правил пожежної безпеки України. У приміщеннях з комп'ютерами рекомендується використовувати системи автоматичної пожежної сигналізації для виявлення пожежі та оповіщення працівників. Для приміщення, по якому проводиться розрахунок, потрібно влаштування місць утримання засобів пожежегасіння вогнегасників порошкових типу ОПУ–5 з вільним доступом у випадку необхідності їх використання.

4.3 Влияние компьютеров на окружающую среду

По мнению исследователей из ООН, настало время для принятия скоординированных международных шагов для уменьшения ущерба окружающей среде, наносимого компьютерным оборудованием. По их данным, при создании одного среднестатистического персонального компьютера общий вес различных химикатов и ископаемого топлива в 10 раз превышает вес окончательного продукта. Причем многие из этих химикатов токсичны, а применение ископаемого топлива усугубляет процесс глобального потепления. Эти отходы затем либо выбрасываются на огромные свалки, либо перерабатываются, зачастую в плохо подходящих условиях в развивающихся странах, что создает существенную угрозу здоровью. Исследователи считают, что необходимо дать больше стимулов как производителям компьютеров, так и пользователям, чтобы они усовершенствовали и повторно использовали свое оборудование, а не выбрасывали его.

По мере того, как компьютеры становятся все меньше и потребляют меньше энергии, их влияние на окружающую среду должно уменьшаться. Однако ученые пришли к выводу, что происходит обратный процесс. Они подсчитали, что для производства компьютера с монитором общим весом в 24 кг необходимо, по меньшей мере, 240 кг ископаемого топлива для обеспечения энергии и 22 кг химикатов. Кроме того, к этому добавляются 1,5 тонны воды. Именно поэтому ученые призывают всех пользователей не менять старые компьютеры на новые без необходимости и стараться просто обновлять его устаревшие элементы.

Рост применения компьютерной техники, ее быстрое моральное старение остро ставит вопрос об утилизации элементов ЭВМ после окончания срока ее эксплуатации. При утилизации старых компьютеров происходит их разработка на фракции: металлы, пластмассы, стекло, провода, штекеры.

В настоящее время разработаны следующие методы переработки компьютерного лома и защиты литосферы от него:

-сортировка печатных плат по доминирующим материалам: дробление и

измельчение, гранулирование, в отдельных случаях сепарация, обжиг полученной массы для удаления сгорающих компонент;

-расплавление полученной массы, рафинирование;

-прецизионное извлечение отдельных металлов: создание экологических схем переработки компьютерного лома;

-создание экологически чистых компьютеров.

-Важной задачей является также переработка медных проводов и кабелей, так как более одной трети меди идет на производство проводов.

Переработку промышленных отходов производят на специальных полигонах, создаваемых в соответствии с требованиями СНиП 2.01.28-85 и предназначенных для централизованного сбора обезвреживания и захоронения токсичных отходов промышленных предприятий, НИИ и учреждений.

4.4 Висновки по розділу

Детально ознайомившись з дипломною роботою, ми одержали глибокі, фундаментальні знання в сфері охорони праці.

Основні висновки з техніки безпеки та охорони праці:

- Для запобігання конфліктів необхідно дотримуватись певних правил поведінки, етики та моралі;

- Для мінімізації впливу електромагнітного випромінювання передбачена заміна застарілої лампової техніки на сучасну;

- Для мінімізації негативного впливу напруженості та інтенсивності праці передбачена оптимальна організація робочого місця та процесу праці;

- Для забезпечення необхідної освітленості передбачено використання енергозберігаючих ламп та додаткових освітлювальних приладів;

- Для забезпечення оптимальних параметрів повітряного середовища передбачено використання систем опалення та кондиціонування.

- Для забезпечення пожежної безпеки передбачено проведення пожежної профілактики, яка включає в себе комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, запобігання пожежі, обмеження її поширення, а також створення умов успішного гасіння.

ВИСНОВКИ

В ході виконання даної дипломної роботи був створений веб-сайт, готовий для використання торгово-сервісним центром «Шангир».

Розроблений сайт орієнтований на залучення нових і обслуговування старих клієнтів сервісного центру. Відвідувачі сайту можуть ознайомитися з діяльністю і здійснюваними послугами, мають можливість отримати відповідь на їх питання, задавши його співробітникові сервісного центру в реальному часі.

При створенні веб-сайту були проаналізовані сучасні веб-технології, що дозволяють створювати динамічні веб-сторінки. Відповідним рішенням для поставлених завдань була система управління контентом Drupal, написана на мові веб-програмування PHP. Створений сайт задовольняє всім вимогам, поставленим ТСЦ «Шангир» до веб-сайту. При створенні були використані готові модулі аутентифікації, пошуку по сайту і інтерактивного спілкування. Дані модулі були доопрацьовані з урахуванням специфіки веб-сайту і успішно впроваджені в його структуру.

Надалі веб-сайт можна вдосконалювати до Інтернет-магазину, завдяки модульній структурі системи Drupal. Так само є можливість доопрацювання структури і змісту сайту з метою підвищення інформативності, привабливості та зручності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Adrian W. West. Practical PHP and MySQL Web Site Databases: A Simplified Approach. - Apress: New York, 2013. – 437 p.
2. Алексеев А.П. Введение в Web-дизайн: учебное пособие. - М.: СОЛОН-ПРЕСС, 2008
3. CMSList. Обзор CMS. Сайт про системы управления сайтом.
<http://www.cmslist.ru>
4. Матеріал з Вікіпедії — вільній енциклопедії про системи управління сайтом.
5. Алексеев А.П. Введение в Web-дизайн: учебное пособие. - М.: СОЛОН-ПРЕСС, 2008
6. Contentmanagementsystem <http://www.brutto.ru/informacija/uznat-bolshe/content-management-system>
7. CMS обзор: CMS, движок сайта, система управления сайтом, mambo, phpnuke, netcat, phpbb, invisionpowerboard, vbulletin.
8. Дронов В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов – СПб.: БХВ-Петербург, 2011
9. «Drupal cms MX 2004 для "чайников".» Уорнер, Джанни, Гарднер, Сюзанна.
10. Лаура Томсони, Люк Веллинг. РАЗРАБОТКА WEB-ПРИЛОЖЕНИЙ НА PHP и MySQL
11. PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов. Дронов В. А. — СПб.: БХВ-Петербург, 2005. — 448 с :
12. PHP and MySQL Web Development (4th Edition), Luke Welling, Laura Thomson 848 стр., сил.; ISBN 978-5-8459-1574-0, 978-0-672-32916-6.
13. PHP и MySQL для начинающих, видеокурс Евгений Попов.
14. Стивен Хольцнер. PHP в примерах. / Стивен Хольцнер. М.: 000 «Бином-Пресс», Пер. с англ., 2007
15. Робин Никсон - Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5, 3-е изд-е, - СПб.: Питер, 2015
16. Севостьянов И. Поисковая оптимизация: практическое руководство по продвижению сайта в Интернете.- СПб.: Питер, 2010
17. Сеницын, Сергей Владимирович. Операционные системы : учебник / С. В. Сеницын, А. В. Батаев, Н. Ю. Налютин. - 3-е изд., стер. - М.: Академия, 2013
18. Ташков П. Веб-мастеринг Веб-мастеринг на 100 процентов HTML, CSS, JavaScript, PHP, AJAX, раскрутка– СПб.: Питер, 2010
19. Гейтс Билл. Бизнес со скоростью мысли. Изд. 2-е, исправленное- М.: ЭКСМО-Пресс, 2001. – 480 с.
20. Лаура Томсон, Люк Веллинг. Разработка Web-приложений на PHP и MySQL: Пер. с англ. — 2-е изд., испр. — СПб: ООО «ДиаСофтЮП», 2003. —672 с.
21. Кэтрин А. и Уэйн П. Adobe Photoshop CS3 «Вершины мастерства» - СПб.: Символ-Плюс – 2011
22. <https://ru.wikipedia.org>
23. Хлебников, Андрей Александрович. Информационные технологии: учебник / А. А. Хлебников. - М. :КноРус, 2014
24. Бучек Г. ASP.NET. Учебный курс. – СПб.: Питер, 2002. – 512с.
25. Гери Д., Хорстманн К. JavaServer Faces – М.: Вильямс, 2008. – 576с.
26. Мальтцер Кевин, Михальски Brent Разработка CGI-приложений на Perl. – М.: Издательский дом «Вильямс», 2001. – 400с.
27. Котеров Д.В. PHP 4. – СПб.: БХВ-Петербург, 2004. – 576с.
28. Кузнецов М., Симдянов И. MySQL 5. – СПб.: БХВ-Петербург, 2010. –

- 1024с.
29. Jason Price, Oracle Database 11g SQL. – McGraw-Hill Osborne Media, 2007. – 656с.
 30. Суэринг С. PHP и MySQL. Библия программиста / С. Суэринг, Д. Парк, Т. Конверс. – Москва: Вильямс, 2010. – 912 с.
 31. Веллинг Л. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон. – Москва: Вильямс, 2010. – 848 с.
 32. Шлосснейгл Д. Профессиональное программирование на PHP / Джордж Шлосснейгл. – Москва: Вильямс, 2006. – 624 с.
 33. Кухарчик А. С. PHP: обучение на примерах / А.С.Кухарчик. – Минск: Новое знание, 2004. – 240 с
 34. Аткинсон Л. PHP 5. Библиотека профессионала / Л. Аткинсон, З. Сураски. – Москва: Вильямс, 2006. – 944 с
 35. Мазуркевич А. М. PHP: Настольная книга программиста / А. М.Мазуркевич, Д. С. Еловой. – Минск: Новое знание, 2004. – 480 с.
 36. Дронов В. А. PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов / Владимир Александрович Дронов. – Санкт-Петербург: БХВ-Петербург., 2009. – 544 с.
 37. Котеров Д.В., Костарев А.Ф. PHP 5. – СПб.: БХВ-Петербург, 2005. – 1120 с.: ил.
 38. Качанов А., Ткаченко В. Букварь по PHP и MySQL. Электронное издание. Версия 1.1 (от 16.05.2000)
 39. Попов Е.Г. Теоретический курс по PHP и MySQL – Москва, Триумф, 2008. – 513с.
 40. Шапошников И.Е. PHP 5.3 - Москва, Питер, 2011 г. - 192 с.
 41. Все о PHP, MySQL и не только! – Режим доступа: <http://php.su/>. – Назва з екрана.
 42. Шмидт Г.А. Практическое введение в программирование на JavaScript – Оренбург, 2011. 458с.
 43. Далаа С.М. Компьютерные сети, интернет и HTML. Учебное пособие. Кызыл: РИО ТывГУ, 2005. – 63 с.
 44. Новикова К. В., Старатович А. С., Медведева Э. А. Интернет-маркетинг и электронная коммерция: учеб.- метод. пособие; Перм. гос. нац. исслед. ун-т. – Пермь, 2013. – 78 с.
 45. Колисниченко Д.Н. Drupal 7. Руководство пользователя. –Киев: Диалектика, 2011. – 208 с.
 46. Обучающий видеокурс Евгения Попова «PHP и MySQL для начинающих» (<<http://www.smartwebmaster.ru/link/popov/php+mysql.html>>)
 47. Nirav Mehta. Choosing an Open Source CMS. Beginner's Guide. — Packt (англ.)русск., April 2009. — 340 p.
 48. Martin Brampton. PHP5 CMS Framework Development. — PACT publishing, June 2008. — 328 p.
 49. Савельева Н. Системы управления контентом (рус.) // Открытые системы. — 2004. — № 4.
 50. Основи охорони праці в підрозділах МНС України : навч. посіб. / В. А. Батлук, Б. О. Білінський, В. В. Ковалишин, О. Л. Мірус ; М-во освіти і науки, молоді та спорту
 51. Охорона праці (лісопаркове господарство) : навч. посіб. / І. А. Березовецька, І. О. Трунова, А. П. Березовецький, І. П. Пістун. – Л. : Ліга-Прес, 2012. – 496 с. : іл. – Бібліогр.: с. 479-489 (153 назви). – ISBN 978-966-397-133-5

52. Семич, А.В. Охрана труда при работе на персональных электронно-вычислительных машинах и другой офисной технике. Практическое пособие / А.В. Семич – Мн.: ЦОТЖ, 2001.
53. Правила охорони праці під час експлуатації електронно-обчислювальних машин . Затверджено МінЮстом України 17 червня 1999 р. №382/3675 . Про охорону праці: Закон України - редакція від 01.01.2004.
54. Державний реєстр міжгалузевих і галузевих нормативних актів про охорону праці (реєстр ДНАОП). – К.: 1998. — 240 с .
55. Закон України "Про охорону праці" Законодавство України про охорону праці , т.1.- К. - 1995. - 558 с. Крамер Эрик HTML: Наглядный курс Web-дизайна. М.: Издательский дом ,2001
56. Рекомендації щодо побудови системи управління охороною праці на виробництві. Затв. наказом МНС України № 398 від 27.06.2006 р.
57. Рекомендации "Управление охраной труда. Основные положения" / Г. Г. Гогиташвили, А. П. Купчин и др. — М.: Изд-во стандартов, 1983. — 15 с.
58. Правила пожежної безпеки в Україні. – К.: Основа, 2002. – 352с
59. ДСанПіН 3.3.6.096-2002 Державні санітарні норми і правила при роботі з джерелами електромагнітних полів.
60. ДСТУ 2293-99 «Охорона праці. Терміни та визначення основних понять»

ДОДАТОК А.

Початковий код шаблону «Corporate Clean»

```

<!-- #header -->
<div id="header">
<!-- #header-inside -->
<div id="header-inside" class="container_12 clearfix"> <!-- #header-inside-left -->
<div id="header-inside-left" class="grid_8">
<?php if ($logo): ?>
  <a href="<?php print check_url($front_page); ?>" title="<?php print t('Home');
?>">" /></a>
  <?php endif; ?>
  <?php if ($site_name || $site_slogan): ?> <div class="clearfix">
    <?php if ($site_name): ?>
      <span id="site-name"><a href="<?php print
check_url($front_page); ?>" title="<?php print t('Home'); ?>"><?php
print $site_name; ?></a></span><br/>
    <?php endif; ?>
    <?php if ($site_slogan): ?>
      <span id="slogan"><?php print $site_slogan; ?></span> <?php endif; ?>
    </div>
  <?php endif; ?>
</div><!-- EOF: #header-inside-left -->
<!-- #header-inside-right -->
<div id="header-inside-right" class="grid_4"> <?php print render($page['search_area']);
?> </div><!-- EOF: #header-inside-right -->
</div><!-- EOF: #header-inside -->
</div><!-- EOF: #header -->
<!-- #header-menu -->
<div id="header-menu">
<!-- #header-menu-inside -->
<div id="header-menu-inside" class="container_12 clearfix">
<div class="grid_12">
<div id="navigation" class="clearfix"> <?php if ($page['navigation']): ?>
<?php print drupal_render($page['navigation']); ?> <?php else :

```

```

if (module_exists('i18n_menu')) {
  $main_menu_tree =
  i18n_menu_translated_tree(variable_get('menu_main_links_source',
  'main-menu'));
} else {
  $main_menu_tree =
  menu_tree(variable_get('menu_main_links_source', 'main-menu'));
}
print drupal_render($main_menu_tree); endif; ?>
</div>
</div>
</div><!-- EOF: #header-menu-inside -->
</div><!-- EOF: #header-menu -->
<!-- #content -->
<div id="content">
<!-- #content-inside -->
<div id="content-inside" class="container_12 clearfix">
<?php if ($page['sidebar_first']) :?> <!-- #sidebar-first -->
<div id="sidebar-first" class="grid_4">
<?php print render($page['sidebar_first']); ?> </div><!-- EOF: #sidebar-first -->
<?php endif; ?>
<?php if ($page['sidebar_first'] && $page['sidebar_second']) {
?>
<div class="grid_4">
<?php } elseif ($page['sidebar_first'] || $page['sidebar_second']) { ?>
<div id="main" class="grid_8">
<?php } else { ?>
<div id="main" class="grid_12">
<?php } ?>
<?php if
(theme_get_setting('breadcrumb_display','corporateclean')): print
$breadcrumb; endif; ?>
<?php if ($page['highlighted']): ?><div
id="highlighted"><?php print render($page['highlighted']);
?></div><?php endif; ?>

```

```

<?php if ($messages): ?>
<div id="console" class="clearfix">
<?php print $messages; ?>
</div>
<?php endif; ?>
<?php if ($page['help']): ?>
<div id="help">
<?php print render($page['help']); ?> </div>
<?php endif; ?>
<?php if ($action_links): ?>
<ul class="action-links">
<?php print render($action_links); ?> </ul>
<?php endif; ?>
<?php print render($title_prefix); ?> <?php if ($title): ?>
<h1><?php print $title ?></h1>
<?php endif; ?>
<?php print render($title_suffix); ?>
<?php if ($tabs): ?><?php print render($tabs); ?><?php
endif; ?>
<?php print render($page['content']); ?>
<?php print $feed_icons; ?>
</div><!-- EOF: #main -->
<?php if ($page['sidebar_second']) :?> <!-- #sidebar-second -->
<div id="sidebar-second" class="grid_4">
<?php print render($page['sidebar_second']); ?> </div><!-- EOF: #sidebar-second -->
<?php endif; ?>
</div><!-- EOF: #content-inside -->
</div><!-- EOF: #content -->
<?php if (($page['footer_first'])and($page['footer_second'])and($page['footer_t hird')):
?>
<!-- #footer -->
<div id="footer">
<!-- #footer-inside -->
<div id="footer-inside" class="container_12 clearfix">
<div class="footer-area grid_4">

```

```

<?php print render($page['footer_first']); ?> </div><!-- EOF: .footer-area -->
<div class="footer-area grid_4">
<?php print render($page['footer_second']); ?> </div><!-- EOF: .footer-area -->
<div class="footer-area grid_4">
<?php print render($page['footer_third']); ?> </div><!-- EOF: .footer-area -->
</div><!-- EOF: #footer-inside -->
</div><!-- EOF: #footer -->
<?php endif; ?>
<!-- #footer-bottom -->
<div id="footer-bottom">
<!-- #footer-bottom-inside -->
<div id="footer-bottom-inside" class="container_12 clearfix"> <!-- #footer-bottom-left --
>
<div id="footer-bottom-left" class="grid_8">
<?php print theme('links__system_secondary_menu', array('links' => $secondary_menu,
'attributes' => array('class' => array('secondary-menu', 'links', 'clearfix')))); ?>

<?php print render($page['footer']); ?>
</div>
<!-- #footer-bottom-right -->
<div id="footer-bottom-right" class="grid_4">
<?php print render($page['footer_bottom_right']); ?> </div><!-- EOF: #footer-bottom-
right -->
</div><!-- EOF: #footer-bottom-inside -->
<?php if (theme_get_setting('credits_display','corporateclean')):
?>
<!-- #credits -->
<div id="credits" class="container_12 clearfix"> <div class="grid_12">
<p>Ported to Drupal by <a
href="http://www.drupalizing.com">Drupalizing</a>, a Project of <a
href="http://www.morethanthemes.com">More than (just) Themes</a>.
Designed by <a href="http://www.kaolti.com/">Zsolt Kacso</a></p>
</div>
</div>
<!-- EOF: #credits -->

```

```
<?php endif; ?>
```

```
</div><!-- EOF: #footer -->
```


ДОДАТОК Б.

Код «JivoSite», який вставляється в тіло документа html

```
<!-- BEGIN JIVOSITE CODE {literal} --> <script type='text/javascript'>
(function(){ var widget_id = 'dJjlyIn7eo';
var s = document.createElement('script'); s.type =
'text/javascript'; s.async = true; s.src =
'//code.jivosite.com/script/widget/'+widget_id; var ss =
document.getElementsByTagName('script')[0];
ss.parentNode.insertBefore(s, ss);})();</script>
<!-- {/literal} END JIVOSITE CODE -->
```

ДОДАТОК В.

Перелік графічних матеріалів магістерської роботи



	Цель и задачи исследования
Методи побудови сайтів ведення блогів на базі технологій PHP-MySQL	<p>Целью данной магистерской аттестационной работы является исследование и практическое использование методов построения сайтов ведения блогов на базе технологий PHP-MySQL.</p> <p>Для достижения поставленной цели необходимо решить следующие задачи:</p> <ul style="list-style-type: none"> - ознакомиться с использованием в Web-разработке современных Интернет-технологий; - исследовать программное обеспечение для проектирования, разработки и эксплуатации сайтов и блогов; - проанализировать стратегии разработки и дизайна сайтов и блогов; - проанализировать методы и способы представления на Web-страницах различных видов информации; - в соответствии с результатами, полученными в результате проведенных исследований, разработать сайт-блог для торгово-сервисного центра (ТСЦ) «Шангир». <p><input type="checkbox"/> Объект исследования - Методы построения сайтов ведения блогов.</p> <p><input type="checkbox"/> Предмет исследования - Методы построения сайтов ведения блогов с использованием технологий PHP-MySQL.</p> <p style="text-align: right;">2</p>

Технологии PHP-MYSQL для создания сайтов и блогов

Для всех уже не секрет, что многие CMS и современные динамические сайты используют связку PHP+MySQL. Это оправдано, так как использование этих технологий позволяет создавать динамические Web-страницы наряду со статическими HTML-страницами.

CMS — это система, которая позволяет в удобном виде создавать и управлять (редактировать, удалять) текстовыми материалами и мультимедиа - документами (содержимое или контент) на сайте.

Таблица 1 - Характеристики популярных систем управления контентом

Характеристики	1С Bitrix	WordPress	Joomla!	Drupal
Бесплатная лицензия	-	+	+	+
Языки программирования	ASP.NET, PHP	PHP	PHP	PHP
База данных	MySQL, MS SQL, Oracle	MySQL	MySQL	MySQL, PostgreSQL
Наиболее подходит как	Много-функц-ая	Блог-сайт	Сайт-визитка	Много-функц-ая
Производительность	Высокая	Высокая	Средняя	Высокая
Расширяемость	+	-	+	+

3

Этапы планирования сайта

Типичный процесс разработки сайта, состоит из упорядоченного процесса действий, который в свою очередь при соответствующем планировании, приведет к итоговому запуску сайта.

Если рассматривать визуально этапы планирования Web-сайта, то это выглядит следующим образом:



4

Анализ технологий создание сайтов и блогов

Таблица 2 - Характеристики языков программирования

Характеристики	ASP.net	Java	Python	Perl	Ruby	PHP
Бесплатная лицензия	-	+	+	+	+	+
Стоимость дополнительного ПО (ОС, БД)	> 40 т.р.	-	-	-	-	-
Кроссплатформенность	-	+	+	+	+	+
Достаточное количество справочных материалов	+	+	+	-	-	+
Легкость в изучении	-	-	-	-	+	+
Много готовых систем управления контентом	-	-	-	-	-	+

5

Обзор и сравнительный анализ базовых технологий создания сайтов и блогов

Таблица 3 - Сравнение современных технологий разработки Web-сайтов.

	PHP	JSP	ASP.NET
многоплатформенность	+	+	-
производительность	+/-	+/-	+
простота использования	+	+/-	+/-
Набор доступных программных библиотек	+	+	+
Распределение дизайна и логики	+/-	+/-	+

Результаты проведенного анализа позволяют сделать вывод о наличии преимуществ эксплуатационных характеристик PHP-технологии. Хотя PHP не очень старый язык, она пользуется большой популярностью среди Web-программистов и сегодня считается едва ли не самым популярным языком для создания Web-приложений (скриптов)

6

Анализ средств реализации баз данных в веб-технологиях

Таблица 4 - Сравнительная характеристика СУБД

	SQL	Oracle	MySQL
Надежность	+	+	+
Быстродействие	-	+	+
Простота	-	-	+
Удобство	+/-	+	+
Использования			

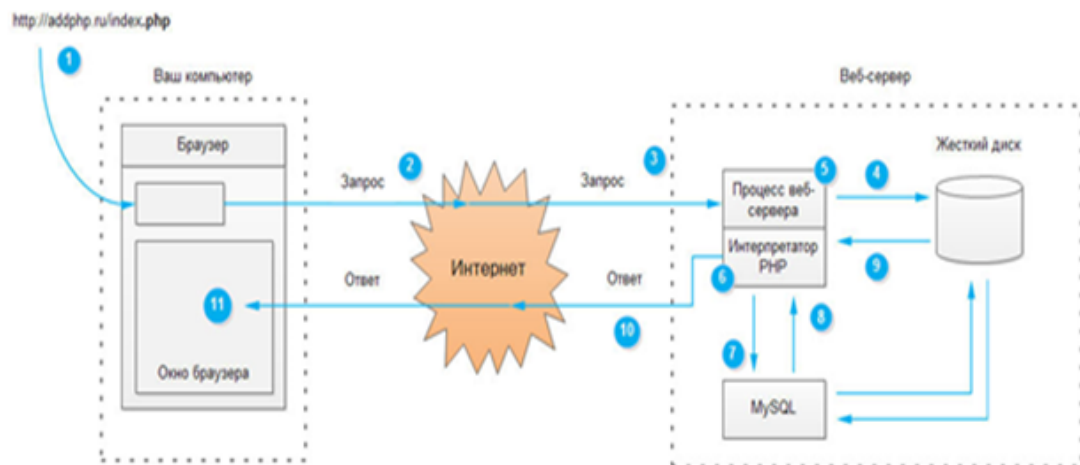
Проанализировав базовые характеристики рассматриваемых СУБД, можно сделать вывод об удобстве использования средств MySQL для создания динамических веб-страниц, поскольку сервер MySQL обеспечивается поддержкой со стороны различных языков программирования, в том числе и PHP.

7

Разработка сайтов и блогов на основе PHP-MYSQL

Составные части сайта, разрабатываемого с использованием PHP-MySQL

В соответствии с техническим заданием разработке подлежит сайт с использованием браузера, веб-сервера Apache, интерпретатора PHP и системы управления базами данных MySQL. На рис. приведена общая схема функционирования такого сайта.



8

Разработка сайтов и блогов на основе PHP-MYSQL

Подготовка системы управления содержимым Drupal для создания Web-сайта

Для работы системы Drupal потребуется веб-сервер Apache с поддержкой языка программирования PHP версии 5.5 или выше, и СУБД MySQL или PostgreSQL.

Методы побuдoвu сайтu вeдeннuя блoгuв нa бaзe тeхнoлoгuй PHP-MYSQL

Форма авторизации в phpMyAdmin

Форма создания новой базы данных.

9

Разработка сайтов и блогов на основе PHP-MYSQL

Настройки подключения к базе данных

Настройка сайта

Методы побuдoвu сайтu вeдeннuя блoгuв нa бaзe тeхнoлoгuй PHP-MYSQL

10

Создание сайта-блога для ТСЦ «Шангир»

Цель и задачи создания сайта-блога

Основанием для создания своего сайта-блога ТСЦ «Шангир» послужили следующие цели:

- поддержка старых и создание новых связей на рынке;
- обеспечение круглосуточной поддержки клиентов и партнеров;
- создание благоприятного мнения о ТСЦ «Шангир».

Разрабатываемый сайт-блог должен решать следующие задачи:

- предоставлять информацию о ТСЦ «Шангир», его деятельности, режиме работы, новых поступлениях, новостях и др.;
- обеспечить возможность поиска по содержимому сайта;
- обеспечить связь посетителя сайта с компанией.

11

Создание сайта-блога для ТСЦ «Шангир»

Создание структуры web-сайта

Главная страница

- |
- | — О компании — | — Новости
- | | — Вакансии
- |
- | | — Партнеры
- |
- | — Услуги — | — Ремонт орг.техники
- | | — Заправка картриджей
- | | — Ремонт компьютерной техники
- | | — Создание сайтов
- |
- | — Статус заказа
- |
- | — Контакты

Иерархическая структура сайта




Главная страница


12

Создание сайта-блога для ТСЦ «Шангыр»


Методы побудови сайтів ведення блогів на базі технологій PHP-MYSQL




О компании



Новости



Форма авторизации на сайте



Интерфейс программы, используемой сотрудником

13

Выводы и Рекомендации

Методы побудови сайтів ведення блогів на базі технологій PHP-MYSQL

В ходе выполнения данной дипломной работы был создан Web-сайт, готовый для использования торгово-сервисным центром «Шангыр». Полученный сайт ориентирован на создание новых и поддержания старых клиентов сервисного центра. Посетители сайта могут ознакомиться с деятельностью и предоставляемыми услугами, имеют возможность получить ответ на интересующий их вопрос, задав его сотруднику сервисного центра в реальном времени.

При создании web-сайта были проанализированы современные web-технологии, позволяющие создавать динамические Web-страницы.

Подходящим решением для поставленных задач была система управления контентом Drupal, написанная на языке Web-программирования PHP.

14

Выводы и Рекомендации

Рекомендации

Практические выгоды, которые Вы получаете, изучив технологий PHP - MySQL:

- Возможность реализовать свой собственный Web-проект без лишних расходов на разработку;
- Возможность заниматься фрилансом по Web-программированию, и работать не выходя из дома;
- PHP - MySQL - это старт ко всем остальным средствам Web-разработки (JavaScript, Ajax, Flash), при взаимодействии с которыми, Вы сможете сделать Ваши сайты еще более удобными.

И самое главное, - полученные знания останетса с Вами навсегда!

15

Спасибо за внимание!