Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії
Кафедра програмної інженерії

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи

магістра

на тему: «Розробка спеціалізованої програмної CMS системи типу WordPress із використанням СУБД MySQL для трейдингових компаній»

Виконав: студент (ка) VI курсу, групи СПм-62

спеціальності (напряму підготовки) 121

Інженерія програмного забезпечення

(шифр і назва спеціальності (напряму підготовки))

(підпис) Мохнацька О.А. (прізвище та ініціали)

Керівник (підпис) Кінах Я.І. (прізвище та ініціали)

Нормоконтроль (підпис) Бойко І.В. (прізвище та ініціали)

Рецензент (підпис) Дмитроца Л.П. (прізвище та ініціали)

Тернопіль – 2019

АНОТАЦІЯ

Темою магістерської роботи є «Розробка спеціалізованої програмної CMS системи типу WordPress із використанням СУБД MySQL для трейдингових компаній» Мохнацької Олени Андріївни. – Тернопільський національний технічний університет імені Івана Пулюя, Факультет комп'ютерно-інформаційних систем і програмної інженерії, Кафедра програмної інженерії, група СПм–62 // Тернопіль, 2019.С. – 88, рис. – 30, табл. – 4, слайдів. – 12, додат. – 4, бібліогр. – 40.

Метою дипломної роботи є подальший розвиток спеціалізованої програмної системи керування мережевим контентом на основі використання СУБД My-SQL для трейдингових підприємств. Методи та програмні засоби, використані при виконанні розробки системи: мова програмування PHP, середовище розробки PhpStorm, СУБД MySQL, методологію гнучкої (Agile) розробки програмного забезпечення.Практична цінність роботи полягає в удосконаленні спеціалізованої програмної системи керування мережевим контентом.

Ключові слова: WORDPRESS, CMS, Web-сайт, СУБД.

ABSTRACT

The theme of this master's thesis is "Development of a specialized CMS software system such as WordPress using MySQL DBMS for trading companies" by Mohnatska Olena. – Ternopil Ivan Puliuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, SPm-62 group // Ternopil, 2019.C. – 88, fig. – 30, tab. – 4 slides. – 12, add. – 4, bibliography. – 40.

The aim of the thesis is to further develop a specialized software system for content management based on the use of My-SQL DBMS for trading companies.
Methods and software used in system development: PHP programming language, PhpStorm development environment, MySQL DBMS, Agile software development methodology.The practical value of the work lies in the improvement of a specialized software system for managing network content.

Keywords: WORDPRESS, CMS, Website, DBMS., CMS, WEB-SITE.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення.

ПС – програмна система, комплекс програмного забезпечення.

ПК – персональний комп’ютер, робоча машина для розробки та виконання програм.

PHP – скрипкова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера.

OpenServer – це портативний локальний WAMP / WNMP сервер, який має багатофункціональну керуючу програму і великий вибір підключених компонентів.

ООП – (Об’єктно-орієнтоване програмування) парадигма програмування, в якій основою є класи та об’єкти, які між собою взаємодіють.

UML – (Unified Modeling Language) уніфікована мова графічного представлення та об’єктного моделювання в області розробки програмного забезпечення парадигми об’єктно-орієнтованого програмування.

HTML (Hyper Text Markup Language) – мова розмітки гіпертекстових документів.

ДНАОП – державний нормативний акт з охорони праці.

ОП – охорона праці.

ССКК – спеціалізована система керування контентом.

CMS – система керування вмістом.

ОПС – об’єкту поточної сторінки.

ІС – інформаційна система.

SQL – мова структурованих запитів

ЖЦ – життєвий цикл.

ЗМІСТ

ВСТУП

Серед програмних рішень, що використовуються для керування контентом слід зазначити такі CMS системи: Drupal, Joomla, Wordpress, Plone та ін. Для дистанційного навчання часто використовуються системи Moodle та aTutor, а також MediaWiki як засіб створення освітніх вікі-проектів. Перевагою цих систем, що зумовило їх популярність, є відкритий програмний код, наявність великої кількості додактів та доступна ліцензія на використання. Серед популярних систем слід зазначити такі: Magento, 1-С Бітрікс; Microsoft SharePoint, Adobe Business Catalyst – комплексні SaaS-системи; Blackboard – система дистанційнооого навчання.

Незважаючи на велике розмаїття програмних систем, що застосовуються для керування контентом, актуальним завданням залишається дослідження, розробка та вдосконалення засобів універсального керування інформаційними об’єктами веб-ресурсів, що дозволить спростити створення нових ресурсів різного призначення, а також забезпечить ефективні механізми їх супроводження та налаштування. Зокрема актуальним є розв’язання наступних задач: створення спеціалізованих інформаційних об’єктів із користувацьким набором полів, керування відображенням таких об’єктів, сортування, фільтрація та навігація по сховищу створених об’єктів, інтеграція сховища інформаційних об’єктів із загальним сховищем контенту, керування ієрархією об’єктів, класифікація та групування об’єктів, що працюють з високою продуктивності.

Метою дипломної роботи є подальший розвиток спеціалізованої програмної системи керування мережевим контентом на основі використання СУБД MySQL для трейдингових підприємств.

Загалом задача подання спеціалізованих інформаційних об’єктів в контексті CMS є дотичною до напрямку об’єктно-реляційного відображення ORM (Object-relational mapping) [12], об’єктно-реляційних баз даних [33] та інших напрямів, пов’язаних із моделюванням даних та знань. Тому подальші дослідження у цій галузі мають перспективу вдосконалити процес керування контентом сучасних інформаційних Веб-систем та підвищити продуктивність створення та підтримки Веб-ресурсів різного призначення.

Дипломна робота описує процес та основні етапи дослідницької роботи та створення програмної системи типу WordPress для трейдингу із використанням СУБД MySql.

Об’єкт дослідження – системи керування контентом та бази даних для побудови спеціалізованої програмної системи.

Предмет дослідження – високопродуктивні спеціалізованні алгоритми і програмні засоби систем керування для Web-контенту.

Наукова новизна полягає у:

– отримала подальший розвиток спеціалізована програмна система керування контентом, що дозволяє раціонально використовувати мережевий ресурс;

– підвищено рівень продуктивності запропонованих мережевих алгоритмів, що скорочує час виконання поставленої задачі на 11%;

– вперше запропоновано високопродуктивну методологію розробки спеціалізованих програмних систем керування контентом на основі використання СУБД, що значно підвищує рівень мобільності програми.

Методи досліджень. Теоретичні дослідження ґрунтуються на застосуванні системного аналізу, об’єктно-орієнтованого проектування, теорії алгоритмів, інженерії знань, теорії множин, теорії графів, алгебраїчної теорії і методології функціонального моделювання.

1 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

1.1 Аналіз вимог до предметної області

1.1.1 Аналіз предметної області

В рамках даного проекту виконується розробка програмної системи типу WordPress для трейдингових компанії з використанням СУБД MySQL.

Предметною областю спеціалізованої системи, частиною якої виступає проектована система керуванням контентом, є об’єкт розміщений для реалізації, відгуки та коментарі до нього.

Проектування, створення, впровадження та супровід ССКК неможливі без використання сучасних методів та інформаційних технологій формування, керування та супроводу контенту [2-9].

В основному існує два основних типи веб-сайтів – статичний та динамічний.

Статичний сайт – це те, що зазвичай пишеться звичайним HTML-кодом, а код у цій сторінці – це те, що відображається користувачеві.

Динамічним сайтом написаний за допомогою сценаріїв на стороні сервера, таких як PHP, ASP, JSP або Coldfusion. На такому веб-сайті вміст викликається задопомогою скриптів з інших файлів або з бази даних, залежно від дійкористувача.

Гнучкість – головна перевага статичного сайту – кожна сторінка за бажанням може бути різною, щоб відповідати різному макеті контенту, і дизайнер може розміщувати будь-які спеціальні ефекти, які клієнт хоче. Це дозволяє шаблонізувати, наприклад, автору необхідно буде іншої теми для для веб-сторінок.

Основна проблема будь-якого статичного сайту з’являється, під час оновлення контенту. Без необхідних знань HTML та дизайну, що використовуються на сайті, необхідно постійно звертатися до адміністратора.

Друга основна проблема – масштабованість. При розростанні проекту необхідно буде постійно створювати нові сторінки, що може зайняти чималий час, зусилля та витрати.

Основні переваги динамічних сайтів полягають у тому, що, підключаючи їх до баз даних, можна легко впорядковувати інформацію організованим та структурованим способом.

Система керування вмістом (CMS) – це програмне забезпечення, яке керує створенням і модифікацією цифрового контенту [18]. CMS зазвичай використовуються для корпоративного управління контентом та управління веб-контентом. Тут представлена деяка кількість безкоштовних CMS, доступних для особистого та корпоративного використання: WordPress, Joomla, Drupal, Oracle WebCenter, OpenText.

Особливості можуть відрізнятися між різними пропозиціями CMS, але основними функціями часто вважаються індексація, пошук та пошук, управління форматом, контроль редагування та публікація.Інтуїтивні функції індексації, пошуку та пошуку індексують усі дані для легкого доступу через функції пошуку та дозволяють користувачам здійснювати пошук за такими атрибутами, як дати публікації, ключові слова чи автор.Управління форматом дозволяє перетворити скановані паперові документи та застарілі електронні документи в HTML або PDF документи. Функції редагування дозволяють оновлювати та редагувати вміст після початкової публікації. Ревізія контролю також відстежує будь-які зміни, внесені до файлів окремими особами.

Функціонал публікації дозволяє користувачам використовувати шаблон або набір шаблонів, затверджених організацією, а також майстри та інші інструменти для створення або зміни вмісту.

MySQL – це найпопулярніша база даних з відкритим кодом у світі, що забезпечує економічну доставку надійних, високопродуктивних та масштабованих веб-додатків та вбудованих баз даних. Це інтегрована

безпечна для транзакцій база даних, сумісна з ACID, з повною віддачею, відкатом, відновленням аварій. MySQL забезпечує простоту використання, масштабованість та високу продуктивність, а також повний набір драйверів баз даних та візуальних інструментів, щоб допомогти розробникам створити та керувати своїми критичними для бізнесу програмами. MySQL розробляється, розповсюджується та підтримується Oracle, а останню інформацію про програмне забезпечення MySQL можна знайти на веб-сайті MySQL. База даних MySQL надає такі функції: висока продуктивність і масштабованість навантажень даних та користувачів.Кластери реплікації самолікування для покращення масштабованості, продуктивності та доступності. SQL використовується для виконання складних і простих запитів, швидких операцій з ключовим значенням.Незалежність від платформи надає гнучкість для розробки та розгортання в декількох операційних системах. Велика сумісність даних MySQL як для оперативного сховища даних, так і для Hadoop та Cassandra.Для досягнення високого рівня масштабованості, безпеки, надійності та безперервного часу MySQL Enterprise Edition включає найбільш повний набір розширених функцій, інструментів управління та технічної підтримки, включаючи MySQL Enterprise Monitor, MySQL Enterprise Backup, а також масштабованість, безпеку, аудит та високі можливості доступності.

1.1.2 Постановка задачі дослідження

Розвиток діяльності в глобальній мережі Інтернет зумовили велику актуальність ресурсів загального призначення, і, зокрема інтернет-сервісів.

На основі комплексного аналізу предметної області, виділено основні завдання для виконання: проаналізувати існуючі системи керуванням вмістом для створення спеціалізованої системи для трейдингових компаній і пошук відповідної архітектури проектування для розробки системи.

Продуктивність, модульність, розширюваність і простота в управлінні – основні вимоги до проекту. Сама CMS система повинна забезпечувати лише базовий функціонал такий як: керування сторінками, структурою сайту і редагування інформації на ньому, який по можливості розширювався. Основна вимога це гнучка конфігурація сайту за допомогою функціональних модулів. Вони повинні розширювати функціонал сайту. Дуже важливо було зробити адміністрування сайту максимально простим і зрозумілим.

Для здійснення всіх поставлених задач перед спеціалізованою CMS системою, вона повинна мати:

– каталог товарів повинен допомогти покупцеві легко знайти та вибрати товар.

– Розширюваний каталог. Можливість зростання асортименту – від одиниць до тисяч товарів.

– Формування замовлення на покупку. Зручний кошик – можливість легко поміняти кількість товарів, видалення і дозамовлення товарів.

– Можливість як швидкого замовлення без реєстрації, так і замовлення з реєстрацією.

– Зручний вибір способів оплати й доставлення. Можливість онлайн-оплати.

– Утримання покупців. Особистий кабінет користувача з історією замовлень і іншими даними.

– Інтеграція з системами аналітики для аналізу конверсій і ефективності реклами.

– Редагування прав доступу.

– Можливість внесення змін в шаблон і доопрацювання під свої потреби.

1.1.3 Аналіз досліджень і публікацій спеціалізованих CMS

Система управління вмістом, часто скорочена як CMS – це програмне забезпечення, яке допомагає користувачам і компаніям створювати, керувати та змінювати вміст на веб-сайті без необхідності спеціалізованих технічних знань. Зазвичай CMS системи мають певний набір вимог (див. рис. 1.1) [10].

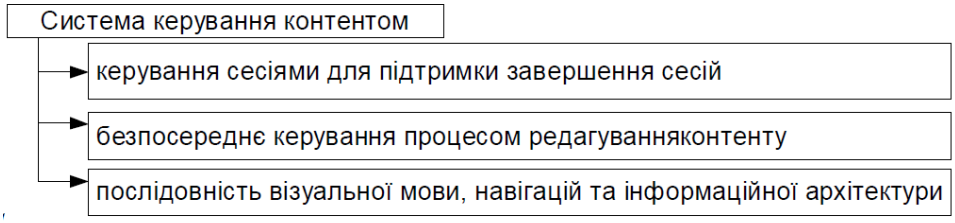


Рисунок 1.1 – Вимоги систем керуваннння контентом

Одні з цих систем є спеціалізовані програмні системи керування контентом (ССКК). Основне призначення цієї системи – це організація процесів формування, підтримки та поширення загальних інформаційних ресурсів у процесах їх використання. Такі CMS дають змогу робити різні текстові і графічні маніпуляції з наповненням інформаційного ресурсу, надаючи користувачеві інструменти зберігання і публікації інформації.

Для генерації вмісту всередині веб-сайтів створюються спеціалізовані системи керування веб-контентом (див. рис. 1.2), які мають ті самі задачі, що й інші різновиди веб-додатків: тобто кешування контенту, його безпека й динамічне збирання. Значущість контенту визначає його привабливість для споживача. Інтеграція застосувань робить ресурси корисними, а інтеграція контенту – привабливими. Оскільки користувачі все частіше схиляються до виконання на сайті щораз більшої кількості застосувань, то застосування ці, зокрема ECM, вимушені виконуватися на самому порталі.

Панель адміністратора дає можливість створювати нові публікації, категорії, теги, сторінки, посилання та користувацькі типи публікацій. Там також змінюються шаблони, додаються віджети, активуються або

оновлюються плагіни та змінюються параметри читання, запису й загальні параметри.Тобто це місце, де створюється вміст і відбувається керування веб-сайтом.

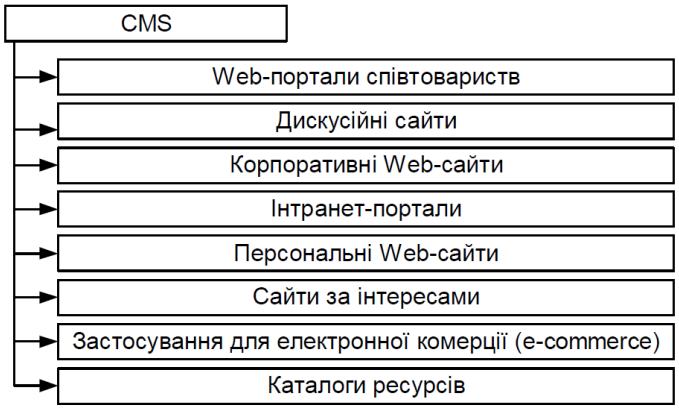


Рисунок 1.2 – Призначення CMS

Як видно з рис. 1.3 подано класифікацію спеціалізованих програмних систем керування контентом з використання СУБД MySql.

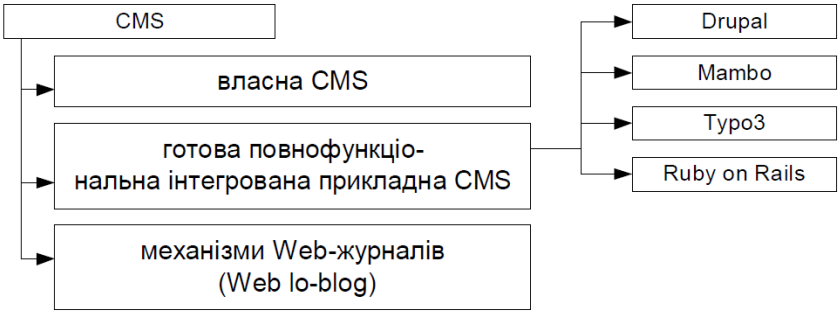


Рисунок 1.3 – Класифікація спеціалізованих систем керування контентом

Система управління веб-сайтами – це інструмент (див. рис.1.4), що дозволяє моделювати розроблені розроблені сайти та керувати їх інформаційними повідомленнями [2, 5]. Ці системи управління вмістом веб-сайтів не потребують спеціальної технічної навігації під час програмування або html-версії. Відображається система управління веб-сайтом, яка дозволяє

змінити контроль над доступом до сайту та зменшити дані. Система управління вмістом виявляє максимальну провокацію розміщення веб-сайтів, зберігаючи гнучку простору на сайті [10].

CMIS	Web-сервер	База даних	Мова
Ruby on Rails	Apache, FastCGI	MySQL, PostgreSQL, SQLite,Oracle, SQL Serer, DB2, Firebird	Ruby
Drupal	Apache IIS	MySQL, PostgreSQL	PHP
Mambo	Apache IIS	Apache IIS	PHP
Typo3	Apache IIS	Apache IIS	PHP

Рисунок 1.4 – Вимоги до ПЗ для основних CMS

Базова спеціалізована система керування забезпечує наступний: функціонал: оновлення контенту на веб-ресурсі; пошуки інформації в Інтернеті; збирання даних про користувачів та потенційних клієнтів; формування і редагування контенту; аналіз відвідування веб-сайту [10]. Основні модулі спеціалізованої CMS системи зображено на рис.1.5:

Назва	Характеристика модуля CMS
Пункти меню	додавання, редагування, управління пунктами меню сайта будь-якого рівня
Статті	додавання, редагування, планування та публікування "статей" (сторінок сайта) у WORD-подібному інтерфейсі
Новини	додавання, редагування та публікування новин
Фотогалерея	можливість робити фотогалереї з підгалереями, автоматичне масштабування фото
Дошка оголошень	додавання оголошень з фотографіями, описом та контактними даними
Налаштування	зберігаються всі налаштування сайта та системи управління сайтом
Користувачі	управління правами зареєстрованих користувачів
Каталог фірм	додавання, редагування, публікування фірм у підгрупах будь-якої вкладеності
Опитування	додавання, редагування опитувань на сайті, результати у вигляді графіків, кількість питань: 2–10

Рисунок 1.5 – Основні модулі системи керування контентом

Застосування спеціалізованої системи керування мережевим контентом не потребує установку додаткового ПЗ на робоче місці. Легкість роботи й зручний інтерфейс із спеціалізованою системою полегшує керування контентом і зменшує наступні витрати на підтримання веб-ресурсу. Для адміністрування та редагування використовується звичайний браузер.

Найпопулярніші CMS системи, згідно з аналітичними даними є WordPress, Joomla, Drupal (див. рис. 1.6) [5].

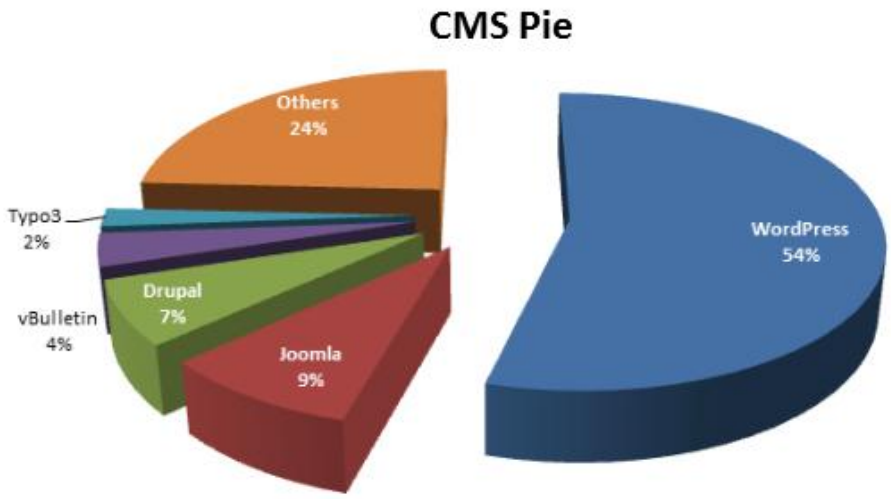


Рисунок 1.6 – Рейтинг CMS на основі вимог

Користуючись переліком завдань до виконання в рамках даної роботи, виконано пошук основних акторів системи та варіанти використання для них.

Ця спеціалізована система планується використовувати для створення web-сайтів. Згідно вимог – це веб-додаток, який знаходиться у харному середовищі для отримання онлайн-послуг на основі певної спеціалізованої системи. Система зосереджена на компанії, які займаються відповідними продажами. Тобто в роботі з системою буде присутні такі актори – Адміністратор, Модератор, Автор.

Розроблення загальної архітектури ССКК сприяє узагальненню методики опрацювання інформаційних ресурсів у ССКК через етапи формування, керування та супроводу контенту загального призначення для скорочення тривалості побудови типових систем електронної діяльності[15].

Впровадження ССКК пришвидшує виробництво власного загального контенту, аналіз зовнішнього контенту з інших джерел, аналіз динаміки життєвого циклу контенту, аналіз статистики функціонування ССКК, аналіз статистики діяльності користувачів інформаційних ресурсів у ССКК,

збільшення цільової аудиторії інформаційних ресурсів та розширення функціональних можливостей цих ССКК [18].

Адміністрація системи здійснюється через інтерфейс адміністратора, який обмежений та реалізований за допомогою логіна та пароля. Він коригує структуру системи та ресурсу, додає, редагує або видаляє права доступу користувачів, змінює правила розповсюдження вмісту.

Призначенням ССКК є формування, керування та супровід загального контенту на засадах опрацювання інформаційних ресурсів. ССКК призначена для створення загальних функціональних вимог та стандартизованих специфікацій щодо розроблення ССКК з оптимізацією етапів процесу опрацювання інформаційних ресурсів у аналогічних системах [12].

ССКК підтримує шість інтерфейсів: з обмеженим доступом для відвідувачів та користувачів, без обмежень для адміністратора та модератора, з вільним доступом для автора.

Додатково реалізований такий сервіс, як вибір контенту за визначений період часу з початку контентного наповнення за допомогою календаря.

Зручна рубрика дозволяє вибирати контент за певною категорією. Використовуйте пошук для пошуку в базі даних за ключовими словами.

Створення та редагування контенту здійснюється за допомогою інтерфейсу автора, доступ до якого обмежений і реалізований за допомогою логіну та пароля.

1.2 Проектування системи

1.2.1 Вибір процесу розробки системи та архітектури

Керування процесом розробки програмного забезпечення (ПЗ) відрізняється особливою складністю та слабкою передбачуваністю кінцевого результату. Наявність різних методологій і підходів до розробки

передбачають використання різних способів до моделювання даного процесу.

Для розробки даної проекту було обрано інкрементну модель. В ітеративній моделі ітераційний процес починається з простої реалізації невеликого набору вимог до програмного забезпечення та ітеративно покращує розвиваючі версії до тих пір, поки не з’явиться повна система реалізовані та готова до розгортання.

Ітераційна модель життєвого циклу не намагається почати із повної специфікації вимог.Натомість розробка починається з конкретизації та впровадження лише частини програмного забезпечення, яке потім переглядається з метою визначення подальших вимог. Потім цей процес повторюють, утворюючи нову версія програмного забезпечення в кінці кожної ітерації моделі.

Ітераційний процес починається з простої реалізації підмножини програмних вимог і ітеративно покращує розвиток версії до моменту впровадження повної системи. На кожній ітерації вносяться модифікації дизайну та додаються нові функціональні можливості. Основна ідея цього методу полягає в розробці системи за допомогою повторних ітераційних циклів і меншими порціями в приріст часу.

На рисунку 1.7 зображено зображення ітеративної та інкрементальної моделі:

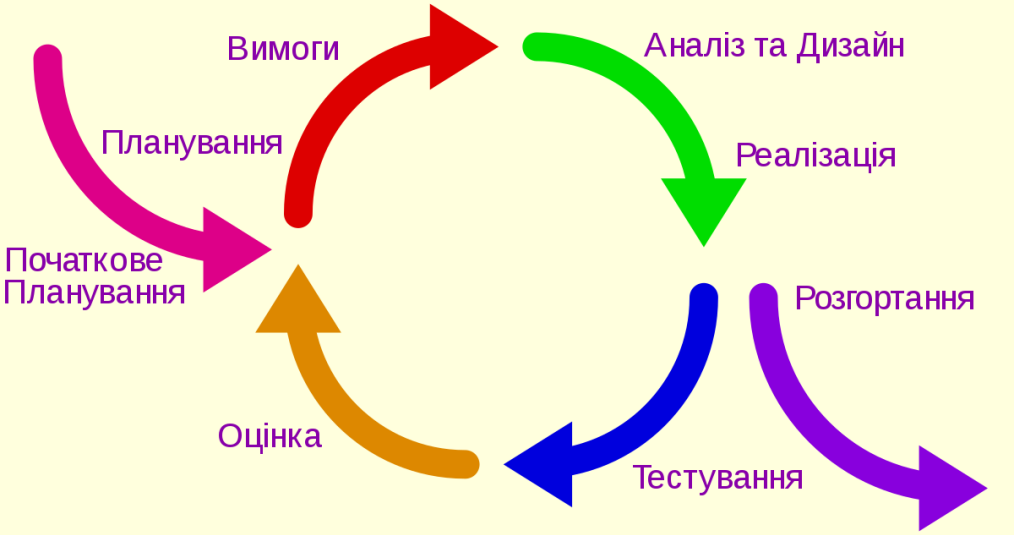


Рисунок 1.7 – Ітеративна та інкрементна модель

Під час розробки програмного забезпечення одночасно може тривати одна ітерація циклу розробки програмного забезпечення. Це процес може бути описаний як еволюційне вдосконалення або поступовий збір. У поступовій моделі всі вимоги поділяється на різні складання. Під час кожної ітерації модуль розробки проходить через вимоги, проектування, впровадження та тестування фази. Кожен наступний випуск модуля додає функцію до попереднього випуску. Процес продовжується, поки повна система не буде готова відповідно до вимог. Запорукою успішного використання ітеративного життєвого циклу розробки програмного забезпечення є сувора перевірка вимог, а також перевірка та тестування кожної версії програмного забезпечення проти цих вимоги в межах кожного циклу моделі. Коли програмне забезпечення розвивається через послідовні цикли, тести повинні бути повторені та розширені для перевірки кожної версії програмного забезпечення.

Для отримання загального контенту з визначеними параметрами користувач повинен пройти певну кількість кроків. Процес керування контентом користувачем або модератором реалізується у 5 етапів:

1. Авторизація в спеціалізованій програмній системі.

2. Вибір із діалогового вікна шаблону контенту.
3. Вибір та налаштовування різних параметрів.
4. Вибір за протоколом шаблону необхідного функціоналу.
5. Генерація контенту із шаблону за визначеними критеріями із етапу 3.

Основою підсистеми керування контентом є ядро – підсистема, яка зв’язує воєдино всі частини застосування та керує завантаженнями, конфігураціями модулів, підключенням загальних залежностей і наданням точок інтеграції інформаційних ресурсів. Значною задачею є забезпечення інформаційних потреб проблемно-орієнтованих елементів системи, підтримання доступу до даних різної категорії користувачів, мінімізація та контроль надлишку даних, дотримання правил цілісності та несуперечності даних, здатність до розвитку та зміни внутрішньої організації інформаційного ресурсу, дотримання вимог якості та ефективності даних. ССКК забезпечує модифікацію інформаційних ресурсів через способи подання, формати та внутрішню організацію контенту; вимоги користувачів, появу нових вимог та категорій користувачів; порядок розподілу контенту та способів доступу користувачів, середовище зберігання контенту, фізичні одиниці зберігання, технічні засоби [20].

Підсистема керування контентом підтримує інтерактивний зв'язок між користувачем та спеціалізованою системою керування контентом через інформаційний ресурс. Підсистема має формувати інформаційний ресурс під потреби користувача та відповідати на його запити. Інформаційний ресурс складається з чітко визначеної множини компонентів, які є напів впорядкованими, тобто деякі з компонентів перебувають у чітко визначеному порядку, а місце їх розташування у графі є не обов’язково визначеним. Кількість компонентів інформаційного ресурсу та відповідний об’єм наборів компонентів є точно визначеними або оціненими. Побудова графу навігації по інформаційному ресурсу відбувається на основі заданого відношення слідування на множині відношень компонентів інформаційного ресурсу.

Навігація виконується без переривання, перехід на новий вузол навігаційного графа обов’язково є логічним. Вузли подаються довільними типами відношень, які не мають істотніших обмежень. Мінімальна кількість вузлів проекту визначає граф, який надаватиме найзмістовнішу інформацію щодо інформаційного ресурсу. В навігаційному графі кожна дуга графу зображає елементарний зв’язок компонентів інформаційного ресурсу та обов’язково має орієнтацію. Шаблони контенту використовують для побудови сторінок з можливостями подавати дані в різних форматах, ділити сторінки на частини, які повторюються, кешувати їх [21].

За допомогою модуля редагування, інформація в базі даних змінюється. А за модулем подання під час кожного запиту сторінки заново створюються (див. рис. 1.9).

На рисунку 1.9 зображено схематичну взаємодію між компонентами:

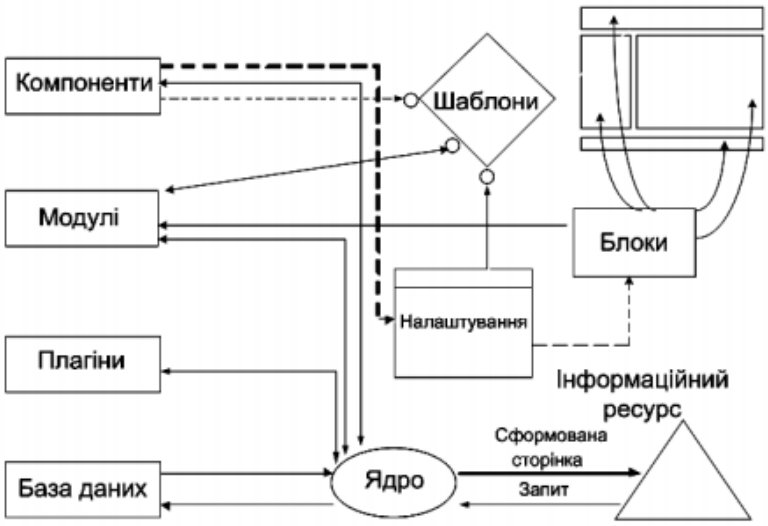


Рис.1.9 – Архітектура компонентів спеціалізованої системи на базі CMS

URL-адреса в модулі аналізу запитів визначає запитувану частину вмісту. Найкраща структура для подання вмісту – ієрархічна, тому доцільно впорядкувати об’єкти вмісту в базі даних по дереву – додати поле, що вказує

ідентифікатор основного елемента. Неправильна обробка даних призведе до повідомлення про помилку та послідовності подальших дій. Процес управління вмістом сервісу та ССКК реалізується на концепції розробки CMS [16].

1.2.2 Модель і структура сторінки спеціалізованої програмної системи керування контентом

Основним завданням спеціалізованої програмної системи є управління веб-вмістом. Керування веб-вмістом – це методи та інструменти, які використовується для організації та забезпечення процесу щодо спільного створення, управління і редагування вмісту сайту. Сучасні розробники CMS систем часто залишають основне завдання системи без уваги та реалізують її виключно на механічному рівні, не створюючи достатньої автоматизації для процесу. Управління веб-контентом повинно забезпечувати ЖЦ інформації, яка надається користувачам сайту.

Логіка під час переміщенню по вмісту сайту, формалізація зв’язків між тематичними сторінками та побудова зручних навігаційних схем – такі завдання потребують більш глибокого перегляду вмісту веб-сайту, ніж простої HTML-розмітки із зображеннями на сервері та прокладанням шляху від управління вмістом до управління логікою у веб-середовищі. Теперішнє управління вмістом потребує розуміти контент як об'єкт структурування, моделювання та формалізації для організації його ефективного зберігання, публікації та презентації для кінцевих користувачів у зручній формі із зручними інкапсульованими навігаційними засобами в спеціалізовані класи програмних систем. Зважаючи на значну роль управління мережевем контентом, пріоритет розробників спеціалізованих CMS повинен бути зосереджений на розробці найефективніших моделей для структуризації, використання та структурування веб-вмісту.

Управління контентом охоплює не лише фізичні процеси розміщення веб-сторінок в глобальній мережі. Але залежно від типу CMS системи, яка

має ієрархічні та структурні зв’язки між елементами вмісту можуть бути по-різному реалізовані.

Як правило, в базі даних CMS системи є спеціальна таблиця, яка представляє цю сутність структури веб-сторінки, характеризуючи її певним набором полів.

Основна інформаційна суть програмної систем управління мережевим контентом – сторінка або частина контенту.У Wordpress загальний елемент вмісту позначається терміном post, у Drupal – Node, у Joomla – матеріалом.

Веб-сайт сайту в загальному вигляді містить (див. рис. 1.10) інформацію, вміст, елементи навігації (меню, шлях навігації, додаткові посилання), елементи дизайну, заголовок сайту, підвал сайту.

Вміст веб-сторінки – це як інформація, щодо якої сторінку було додано на веб-сайт, так і інформація, щодо якої користувач переходить на цю сторінку. На додаток до ефективного вмісту на сторінці може бути безліч довідкової інформації (елементи дизайну, елементи навігації, реклама тощо). Найефективніший вміст сторінки – це логічний фрагмент інформації в загальній інформаційній структурі веб-сайту. Саме ці фрагменти та їх взаємозв'язки насамперед потребують ефективного моделювання та подання до бази даних веб-системи.

Рис. 1.10 – Загальна структура веб-сторінки

Цей набір атрибутів є основним для більшості сторінок сайту, серед яких такі: домашня сторінка, сторінка розділу, сторінка інформації, стаття, новини, продукт, що має додаткові атрибути.

Таким чином, ефективний вміст веб-сторінки буде сприйматися як суть системи, яка має бути формалізована в базі даних як елемент контенту або сторінки. Тому під сторінками терміна в межах інформаційно-логічної моделі системи ми сприймемо частину інформації, що складається з вищезазначених атрибутів і може бути представлена на веб-сайті як веб-документ із власною URL-адресою.

Крім окремого веб-документа, на веб-сайті також може бути опублікований змістовий елемент або сторінка у таких формах: посилання всередині вмісту іншої сторінки; у цьому випадку атрибут URL використовується для публікації сторінки та, можливо, анотації як значення атрибута заголовка тегу <a>; посилання як пункт меню; аналогічно, у цьому випадку атрибут URL та примітка використовуються для публікації сторінки як атрибута заголовка тега <a>; анотований блок посилань – це форма

подання посилання на сторінку, яка містить заголовок посилання (назва пов'язаної сторінки), анотацію на сторінці та ілюстрацію. Цей дизайн зручний і інформативний, тому він поширений на багатьох веб-ресурсах. Якщо ви розмістите позначене посилання у стрічці новин, блок посилань може також містити дату сторінки. На основі заданої структури сторінки як елемента змісту можна описати основну структуру таблиці сторінок у базі даних сайту (див. рис. 1.11).

Код сторінки, на її основі формується унікальна URL-адреса сторінки.

Заголовок
Короткий заголовок
Анотація
Основний контент (html)
Ілюстрація основна
Ілюстрація для анотації
Дати

Рисунок 1.11 – Основна структура сторінки

Таблиця класів є ключовою сутністю в базі даних спеціалізованої системи WordPress CMS. По мірі розгляду завдань, що розв'язуються системою CMS, ця таблиця буде оновлена новими атрибутами.

Відповідним типом даних для коду сторінки є тип рядка. Таким чином, кожна сторінка матиме унікальний ідентифікатор тексту, наприклад, про, контакт, продукти тощо. Такі ідентифікатори формуватимуть зручні та зрозумілі URL-адреси для сторінок, наприклад «site.com/?page=about».

Багатомовність сайту реалізується різними способами – від спеціальних утворень в базі даних, що містять переклад, до окремих гілок в ієрархії вмісту, де нова мовна версія сайту – лише черговий розділ сайту.

Такі підходи не завжди ефективні з точки зору зручності використання та простоти в реалізації.

Відповідно до контентної моделі, де сторінка є основною сутністю у базі даних, багатомовність зручно реалізовувати на основі таблиці сторінок. Таким чином, для кожного з полів, що містять текстову інформацію, створюється аналогічне поле для іншої мовної версії сайту, наприклад: "textUa, textRu, textEn" – поля, що містять базовий вміст html для різних мовних версій. Такий підхід дозволить вам використовувати ту саму логіку бізнесу при роботі зі сторінками, незалежно від мовної версії.

Клас системи та клас сторінки будуть основними об’єктами спеціалізованої програмної системи.

Робота з веб-сторінками буде реалізована за допомогою класу «Pages». Інший клас спеціалізованої програмнох системи відповідає за логіку роботи веб-додатку та обробку певних запитів. Діаграма послідовності роботи системи зображено на рис. 1.12.

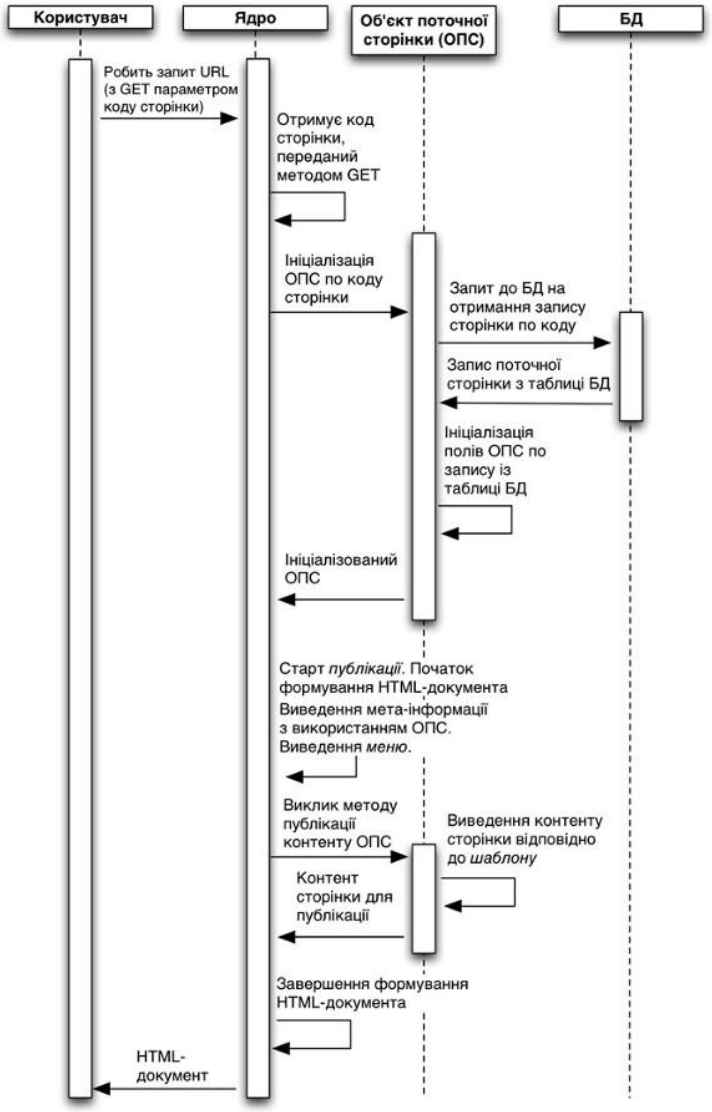


Рисунок 1.12 – Діаграма послідовності CMS

Публікація сторінки передбачає ініціалізацію запиту на веб-сайт, який створює об’єкт поточної сторінки за кодом сторінки за допомогою ядра системи.

Під час ініціалізації об'єкт поточної сторінки отримує доступ до бази даних, щоб отримати відповідний запис із таблиці сторінок. Ядро системи починає процес публікації сторінки. Заголовок документа та елементи навігації готуються відповідно до шаблону. Система доопрацьовує документ і надсилає веб-сторінку користувачеві.

Особливості прямого опублікування документа повинні базуватися на роботі спеціальної підсистеми шаблонізації, яка дозволяє відрізняти ділову логіку серверного коду від подання контенту за допомогою HTML.

1.3 Конструювання програмної системи

1.3.1 Використані технологій для розробки

Для реалізації спеціалізованої системи потрібно звернути увагу на вибір мови, середовища і СКБД. Адже від правильності вибору залежатиме результативність та швидкість розробки.

Існуючі на сьогоднішній день підходи щодо проектування і реалізації систем управління контентом припускають вибір різних стеків технологій для розробки. Так, в залежності від вимог, висунутих розробниками до системи, процес прийняття рішення про вибір відповідних мов програмування спрощується. Розробляється інформаційна система, що включає в себе клієнтську частину, з якої безпосередньо взаємодіє користувач, і серверну частину, що виконує обчислення і здійснює управління ресурсами, не вимоглива до забезпечення відмовостійкості, тому що не передбачається пікового високого трафіку. З іншого боку, швидкість розробки програмного рішення, природність і органічність використання PHP для реалізації веб-сайту є важливими факторами.

Для проектування та інтеграції системи управління контентом з урахуванням наявного програмного рішення, реалізованого з використанням таких мов як HTML, XML, CSS, JavaScript, а також PHP, було прийнято рішення про використання PHP для розробки системи управління контентом. Таке рішення є оптимальним при розгляді стека технологій і завдань, які поставлені перед системою. Середовищем розробки є phpStorm IDE, допоміжні технології CSS, JavaScript, СКБД MySql.

MySQL становить дедалі більшу конкуренцію таким дорогим гігантам як Oracle і MS SQL Server. Важливим фактором є те, що СУБД MySQL розповсюджується абсолютно безкоштовно. В даний час пакет MySQL доступний як програмне забезпечення з відкритим вихідним кодом.MySQL відрізняться хорошою швидкістю роботи, надійністю, гнучкістю. Робота з нею, як правило, не викликає великих труднощів. Підтримка сервера MySQL автоматично включається в поставку PHP [30].

Веб-сервер Apache продовжує залишатися одним з найпопулярніших веб-серверів для розміщення веб-контенту. В комерційній чи у відкритому ресурсі. Веб-сервер Apache - це спільний проект 1995 року, який пропонує відкритий доступ для безкоштовно користування на веб-сервері HTTP.

1.3.2 Опис реалізації спеціалізованої системи

Контент – мультиплекс продуктів й інформаційних ресурсів, які зберігаються у певному середовищі ІС і доступні користувачам цієї системи [11]. Формами мережевого контенту є інформаційний ресурс, інформаційний вміст веб-сайту, який є об’єктом бізнес-процесів контент-сервісу [12]. Стрімкий розвиток інформаційних технологій зробив так, що в сучасному світі мережевий контент став основним поняттям у процесах розвитку, який впроваджений в майже в усіх галузях. Успішний розвиток глобальної мережі та популяризація онлайн-послуг показали, що інформаційна частина є найдинамічнішою і найприбутковішою [6].

Більшість популярних сьогодні спеціалізованих систем керування мережевим контентом не підтримують весь життєвий цикл, формування, керування, реалізація мережево-контентного потоку, також і не вирішують основних задач з оброблення інформаційних об’єктів – задач формування та реалізації контенту.

Існує декілька різних концепцій та моделей життєвого циклу мережевого контенту, розробники яких пропонують та описують кілька

етапів з набором методів, наприклад, керування записами, цифровими активами, спільної роботи й версіями, які реалізовано різними технологіями та процесами. У деяких моделях життєвого циклу мережевого контенту передбачені концепції керування проектами, керування інформацією, інформаційна архітектура, стратегії контенту, керування веб-сайтом, семантичний друк [18].

Типова модель спеціалізованої програмної системи не повністю реалізує методи керування мережевим контентом. Типові моделі керування мережвевим контентом призначені лише для визначення процесів актуальності мережевого контентного потоку, а деякі із них, такі як аналітична, логістична, та для тематичного контентного потоку. Вони не вирішують задачі типізації й реалізації контенту і не всі задачі з керування мережевим контентом, наприклад, подання мультиплексу контенту кінцевій аудиторії згідно з їх запитами, історією або інформаційним портретом, інформаційних структур, автоматичне визначання тематичної інформації, побудова таблиць взаємозв'язку понять, підрахунок рейтингів понять, збирання інформації з різних джерел та її форматування, виявлення основних слів та понять мережевого контенту, автоматична рубрикація контенту, виявлення дублювання мережвого контенту, вибіркове поширення мережвого контенту. А основний недолік типових моделей керування контентом – це відсутність зв'язків між вхідною інформацією, контентом та вихідною інформацією у спеціалізованих програмних системах.

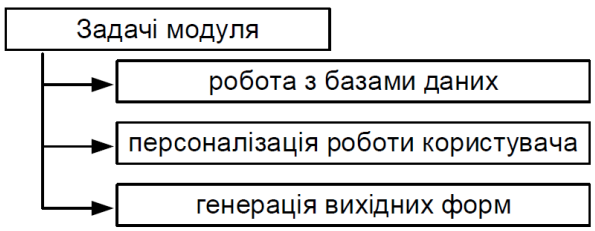


Рисунок 1.13 – Основні задачі модуля

Головним завданням модуля керування контентом (див. рис.1.13) є формування, оновлення та забезпечення доступу до БД, формування швидких і продуктивних БД, шаблонізація роботи користувачів, збереження персональних даних від користувачів і джерел, ведення статистики роботи, забезпечення швидкого пошуку в базі даних, генерація вихідних форм, інформаційна взаємодія з іншими БД. На рис. 1.14 зображено класифікацію основних моделей керування контентом.

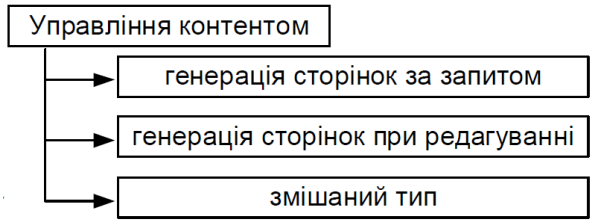


Рисунок 1.14 – Моделі керування контентом

Аналітики розглядають різноманітні етапи ЖЦ для веб-контенту. Майже в усіх запропонованих моделях, основні фази життєвого циклу такі: розробка, поширення та архівування контенту, перегляд. ЖЦ дій, процесів, статусу та ролі керування мережевого контенту відрізняються у моделях залежно від вимог і можливостей, організаційних стратегій, потреб тощо [10, 16].

1.4 Проектування та реалізація таблиць баз даних

Для кращого розуміння задачі створення архітектури БД наводяться основні інформаційні сутності, що мають бути в ній присутні. На даному етапі виділені наступні сутності: користувачі, розділи, підрозділи, новини, журнал, зворотній зв'язок, cms, адміністратори, реклама.

У відповідності до вибраної архітектури, необхідно реалізувати моделі, які будуть виступати у ролі посередника між скріптом і базою даних. База

даних повинна відповідати усіім поставленим вимогам і будуватись у відповідності до моделей. Загальна структура розробленої БД з усіма зв'язками зображена в додатках.

Отже в даному пункті розглянуто основні інформаційні сутності, які мають бути застосовані для зберігання інформації в системі. Слід зазначити, що при реалізації таблиць БД з'являться додаткові, більш деталізовані елементи.

У відповідності до вибраної архітектури, необхідно реалізувати моделі, які будуть виступати у ролі посередника між скріптом і базою даних. База даних повинна відповідати усіім поставленим вимогам і будуватись у відповідності до моделей. Загальна структура розробленої БД зображена на рисунку 1.15:

	Table	Action	Records [1]	Type	Collation	Size	Overhead
☐	admin		1	MyISAM	latin1_swedish_c	2.0 KiB	-
☐	tbl_advertisement		1	MyISAM	latin1_swedish_c	2.1 KiB	-
☐	tbl_cms		6	MyISAM	latin1_swedish_c	5.2 KiB	124 B
☐	tbl_feedback		0	MyISAM	latin1_swedish_c	1.0 KiB	-
☐	tbl_magazine		4	MyISAM	latin1_swedish_c	2.4 KiB	-
☐	tbl_momentphotos		0	MyISAM	latin1_swedish_c	1.0 KiB	-
☐	tbl_moments		39	MyISAM	latin1_swedish_c	3.5 KiB	-
☐	tbl_news		27	MyISAM	latin1_swedish_c	14.8 KiB	-
☐	tbl_ourpillars		1	MyISAM	latin1_swedish_c	2.1 KiB	-
☐	tbl_sections		8	MyISAM	latin1_swedish_c	2.7 KiB	-
☐	tbl_subsections		9	MyISAM	latin1_swedish_c	2.9 KiB	-
☐	tbl_users		2	MyISAM	latin1_swedish_c	2.3 KiB	-
	12 table(s)	Sum	98	MyISAM	latin1_swedish_ci	41.9 KiB	124 B

Рисунок 1.15 – База даних спеціалізованою CMS системи

Ця база складається з таблиць для зберігання інформації для: користувачів, розділів, підрозділів, новинин, журналів, зворотнього зв'язоку, cms, адміністраторів та реклама.

Таблиця Admin відображає сутність в даній роботі – адміністратори(рис.1.16):

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	Id	int(11)			No	None	auto_increment	
☐	username	varchar(25)	latin1_swedish_ci		No	None		
☐	password	varchar(32)	latin1_swedish_ci		No	None		
☐	Created_date	datetime			No	None		
☐	Status	tinyint(4)			No	None		

Рисунок 1.16 – Таблиця Admin

Таблиця Admin містить такі поля:

1. Id – дане поле є первиним ключем таблиці, що визначає адміністратора;
2. Username – символьне поле, що містить ім'я користувача;
3. Password – символьне поле, що містить пароль;
4. Created_date – дата створення обєкту;
5. Status – числове значення, що містить статус цього обєкту;

Таблиця Advertisement відображає сутність в даній роботі – реклама(рис.1.17):

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	Id	int(11)			No	None	auto_increment	
☐	title	varchar(300)	latin1_swedish_ci		No	None		
☐	link	text	latin1_swedish_ci		No	None		
☐	advertisementimage	text	latin1_swedish_ci		No	None		
☐	Created_date	datetime			No	None		
☐	Status	tinyint(4)			No	None		

Рисунок 1.17 – Таблиця Advertisement

Таблиця Advertisement містить такі поля:

1. Id – дане поле є первиним ключем таблиці, що визначає рекламу;
2. title – текстове поле, що визначає загаловок реклами;
3. link – текстове поле, що містить посилання на рекламу;
4. Advertisementimage – текстове поле, що містить адресу зображення реклами;
5. Created_date – дата створення обєкту;
6. Status – числове значення, що містить статус цього обєкту;

Таблиця CMS відображає сутність в даній роботі – основу спеціалізованої системи(рис.1.18):

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	Id	int(11)			No	None	auto_increment	
☐	PageTitle	varchar(20)	latin1_swedish_ci		No	None		
☐	PageName	varchar(20)	latin1_swedish_ci		No	None		
☐	PageMetaKeywords	text	latin1_swedish_ci		No	None		
☐	PageMetaDescription	text	latin1_swedish_ci		No	None		
☐	PageDescription	text	latin1_swedish_ci		No	None		
☐	Position	tinyint(4)			No	None		
☐	Created_date	datetime			No	None		
☐	Status	tinyint(4)			No	None		

Рисунок 1.18 – Таблиця CMS

Таблиця CMS містить такі поля:

1. Id – дане поле є первиним ключем таблиці, що визначає CMS;
2. PageTitle – символьне поле, що містить заголовок сторінки;
3. PageName – символьне поле, що містить назву сторінки;
4. PageMetaKeywords – текстове поле, що містить мета-слова до сторінки;
5. PageMetaDescription – текстове поле, що містить мета-опис сторінки;
6. PageDescription – текстове поле, що містить опис до сторінки;
7. Position – текстове поле, що визначає позицію сторінки;
8. Created_date – дата створення обєкту;
9. Status – числове значення, що містить статус цього обєкту;

Таблиця Feedback відображає сутність в даній роботі – зворотний зв’язок(рис.1.19):

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	Id	int(11)			No	None	auto_increment	
☐	userid	int(11)			No	None		
☐	description	text	latin1_swedish_ci		No	None		
☐	Created_date	datetime			No	None		
☐	Status	tinyint(4)			No	None		

Рисунок 1.19 – Таблиця Feedback

Таблиця CMS містить такі поля:

1. Id – дане поле є первиним ключем таблиці, що визначає зворотний звя’зок;

2. UserId – дане поле є вторинним ключем таблиці, що містить ідентифікатор користувача з таблиці Users;

3. Descroption – текстове поле, що містить текст зворотного зв’язка;

4. Created_date – дата створення обєкту;

5. Status – числове значення, що містить статус цього обєкту;

Таблиця Magazine відображає сутність в даній роботі – журнал(рис.1.20):

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	Id	int(11)			No	None	auto_increment	
☐	magazinetitle	varchar(400)	latin1_swedish_ci		No	None		
☐	magazinefile	text	latin1_swedish_ci		No	None		
☐	magazinecoverimg	text	latin1_swedish_ci		No	None		
☐	descriptions	text	latin1_swedish_ci		No	None		
☐	Created_date	datetime			No	None		
☐	Status	tinyint(4)			No	None		

Рисунок 1.20 – Таблиця Magazine

Таблиця Magazine містить такі поля:

1. Id – дане поле є первиним ключем таблиці, що визначає зворотний звя’зок;

2. Magazinetitle – символьне поле, що містить заголовок журналу;

3. Magazinefile – текстове поле, що містить документ журталу;

4. Magazinecoverimg – тестове поле, що містить адрес на ображення журналу;

5. descroption – текстове поле, що містить текст журналу;

6. Created_date – дата створення обєкту;

7. Status – числове значення, що містить статус цього обєкту;

Таблиця News відображає сутність в даній роботі – новини(рис.1.21):

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	Id	int(11)			No	None	auto_increment	
☐	newstitle	varchar(400)	latin1_swedish_ci		No	None		
☐	newsimage	text	latin1_swedish_ci		No	None		
☐	descriptions	text	latin1_swedish_ci		No	None		
☐	Created_date	datetime			No	None		
☐	Status	tinyint(4)			No	None		

Рисунок 1.21 – Таблиця News

Таблиця News містить такі поля:

1. Id – дане поле є первиним ключем таблиці, що визначає зворотний звя'зок;
2. newstitle – символьне поле, що містить заголовок новин;
3. descroption – текстове поле, що опис повини;
4. Created_date – дата створення обєкту;
5. Status – числове значення, що містить статус цього обєкту;

Таблиця OurPillars відображає сутність в даній роботі – основне про компанію(рис.1.22):

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	Id	int(11)			No	None	auto_increment	
☐	firstname	varchar(30)	latin1_swedish_ci		No	None		
☐	middlename	varchar(30)	latin1_swedish_ci		No	None		
☐	lastname	varchar(30)	latin1_swedish_ci		No	None		
☐	picture	text	latin1_swedish_ci		No	None		
☐	address	text	latin1_swedish_ci		No	None		
☐	descriptions	text	latin1_swedish_ci		No	None		
☐	contactno	varchar(20)	latin1_swedish_ci		No	None		
☐	Created_date	datetime			No	None		
☐	Status	tinyint(4)			No	None		

Рисунок 1.22 – Таблиці OurPillars

Таблиця OurPillars містить такі поля:

1. Id – дане поле є первиним ключем таблиці, що визначає зворотний звя'зок;
2. Descroption – текстове поле, що опис зворотного зв'язка;
3. Created_date – дата створення обєкту;
4. Status – числове значення, що містить статус цього обєкту;

Таблиця Sections відображає сутність в даній роботі – розділи(рис.1.23):

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	Id	int(11)			No	None	auto_increment	
☐	section_name	varchar(100)	latin1_swedish_ci		No	None		
☐	link	varchar(400)	latin1_swedish_ci		No	None		
☐	descriptions	text	latin1_swedish_ci		No	None		
☐	Order_no	int(11)			No	None		
☐	Created_date	datetime			No	None		
☐	Status	tinyint(4)			No	None		

Рисунок 1.23 – Таблиця Sections

Таблиця CMS містить такі поля:

1. Id – дане поле є первиним ключем таблиці, що визначає розділи;
2. Sectioname – символьне поле, що мість назву розділу;
3. Description – текстове поле, що містить опис розділу;
4. Order_no – числове значення, яке вказує на розташування розділу;
5. Created_date – дата створення обєкту;
6. Status – числове значення, що містить статус цього обєкту;

Таблиця Sub-Sections відображає сутність в даній роботі – підрозділи(рис.1.24):

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	Id	int(11)			No	None	auto_increment	
☐	sectionid	int(11)			No	None		
☐	subsection_name	varchar(100)	latin1_swedish_ci		No	None		
☐	link	varchar(400)	latin1_swedish_ci		No	None		
☐	descriptions	text	latin1_swedish_ci		No	None		
☐	Order_no	int(11)			No	None		
☐	Created_date	datetime			No	None		
☐	Status	tinyint(4)			No	None		

Рисунок 1.24 – Таблиця Sub-Sections

Таблиця Sub-Sections містить такі поля:

1. Id – дане поле є первиним ключем таблиці, що визначає підрозділи;
2. SectionId – дане поле є вторинний ключем, що містить ідентифікатор розділу з таблиці Section;
3. Subsectioname – символьне поле, що мість назву підрозділу;

4. Description – текстове поле, що містить опис підрозділу;
5. Order_no – числове значення, яке вказує на розташування підрозділу;
6. Created_date – дата створення обєкту;
7. Status – числове значення, що містить статус цього обєкту;

Таблиця Users відображає сутність в даній роботі – користувачі(рис.1.25):

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	Id	int(11)			No	None	auto_increment	
☐	username	varchar(25)	latin1_swedish_ci		No	None		
☐	password	varchar(32)	latin1_swedish_ci		No	None		
☐	firstname	varchar(30)	latin1_swedish_ci		No	None		
☐	middlename	varchar(30)	latin1_swedish_ci		No	None		
☐	lastname	varchar(30)	latin1_swedish_ci		No	None		
☐	picture	text	latin1_swedish_ci		No	None		
☐	gender	tinyint(4)			No	None		
☐	encodedata	text	latin1_swedish_ci		No	None		
☐	address	text	latin1_swedish_ci		No	None		
☐	country	int(11)			No	None		
☐	state	int(11)			No	None		
☐	city	int(11)			No	None		
☐	emailid	varchar(320)	latin1_swedish_ci		No	None		
☐	contactno	varchar(20)	latin1_swedish_ci		No	None		
☐	Created_date	datetime			No	None		
☐	usertype	tinyint(4)			No	None		
☐	linkcheck	int(11)			No	None		
☐	approve	tinyint(4)			No	None		
☐	Status	tinyint(4)			No	None		

Рисунок 1.25 – Таблиця Users

Декілька полів таблиці Users наведено нище:

1. Id – дане поле є первиним ключем таблиці, що визначає користувача;
2. Username – символьне поле, що містить логін користувача;
3. Password – символьне поле, що містить пароль;
4. Firstname – символьне поле, що містить ім’я користувача;
5. Lastname – символьне поле, що містить прізвище користувача;
6. EmailId – символьне поле, що містить назву пошти користувача;
7. Type – VARCHAR(45) – символьне поле, що містить тип користувача;
8. Created_date – дата створення обєкту;
9. Status – числове значення, що містить статус цього обєкту;

Діаграма відносин між сутністностями (ERD) – це техніка моделювання даних, яка створює графічне представлення сутностей та взаємозв'язків між сутностями в межах інформаційної системи. Діаграма сутностей бази даних спеціалізованої програмної Cms системи типу WordPress зображено на рисунку 1.26.

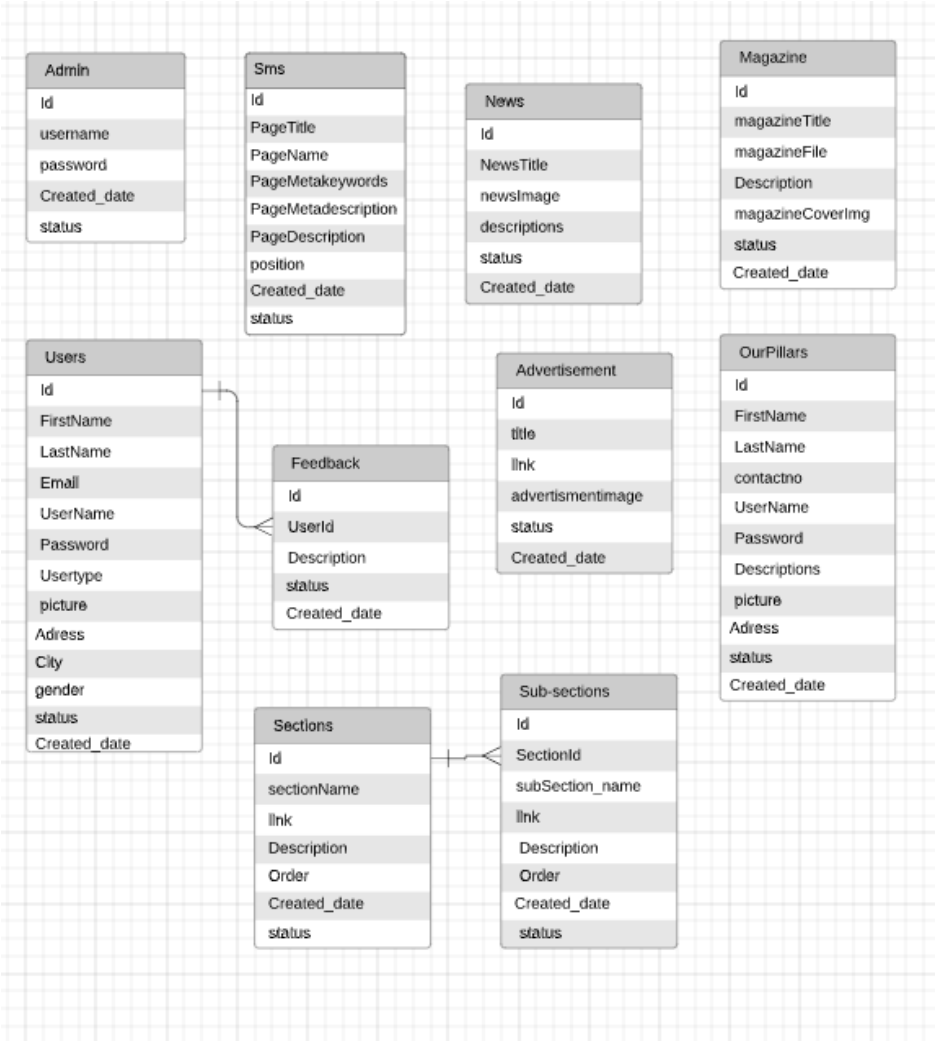


Рисунок 1.26 – ER-діаграма бази даних

1.5 Розробка модуля користувача в спеціалізованій системі

Модуль користувача спеціалізованої програмної системи керуванням контентом містить реалізацію авторизації та аутентифікації, тобто

визначення прав для кожного користувача та відображення відповідного меню і фунціональних можливостей.

Для реалізації було використано найбільш розповсюджені технології веб-розробки: HTML, CSS, JavaScript, PHP.

Користувачі поділяються на такікатегорії: клієнти, які мають доступ лише до можливості перегляду інформації, та адміністратори, хто може спостерігати за всім процесом роботи системи. На рисунку 1.27 зображено команди, які знаходяться в «index.php», задопомого яких виконується збір декількох сторінок в одну сторінку.

```
<!doctype html>
<html lang="en">

<?php

session_start();

if (!(isset($_SESSION['valid_user']) && $_SESSION['valid_user'] != '')) {
    header ("Location: ../modules/mod_login/login.php");
}
echo ('<p>Welcome '.$_SESSION['valid_user'].'  :)</p>');


include_once 'header.php';
include_once 'nav.php';
include_once 'intro.php';
include_once 'content.php';
include_once 'aside.php';
include_once 'footer.php';
?>
```

Рисунок 1.27 – Сторінка Index.php

Основною метою розробки графічного інтерфейсу користувача є створення ефективного засобу взаємодії між системою та людиною [18].

Інтерфейс повинен підтримувати увесь наявний функціонал серверної частини, надавати доступ до перегляду усієї інформації, що зберігається в базі даних, а також можливість редагування та доповнення даної інформації. Окрім цього, зазначені властивості повинні бути реалізовані таким чином, щоб користування системою було інтуїтивним.

Піж час розробки графічного інтерфейсу можливі два варіанти реалізації. Оскільки система, що розробляється, буде побудована за принципами клієнт-серверної архітектури, клієнт представлятиме окрему підсистему, яка буде виконувати свої функції за рахунок взаємодії з прикладним програмним інтерфейсом (API), який надаватиме серверна частина [16].

Перше, що повинен побачити користувач увійшовши в систему, це вікно для введення логіну та паролю. Його розробка зображена в лістингу. Його розробка зображена в лістингу 1.1:

Лістинг 1.1 – Реєстрація користувача

```
    <form action="/enter" method="post" accept-charset="utf-8" class="form-horizontal">  <div class="control-group">
        <label id="username" class="control-label" for="username">Логін</label>    <div class="controls">
          <input type="text" id="username" name="username" size="20" placeholder="Логін" autofocus="autofocus">    </div>
      </div>
      <div class="control-group">       <label class="control-label" for="password">Пароль</label>    <div class="controls">
          <input type="password" id="password" name="password" value="" size="20" placeholder="Пароль">    </div>  </div>
      <div class="control-group">
        <label class="control-label" for="RememberMe">Запам'ятати</label>       <div class="controls RememberMe">
            <div class="iPhoneCheckContainer" style="width: 81px;"><input type="checkbox" name="rememberme" id="rememberMe"><label class="iPhoneCheckLabelOff" style="width: 76px;">
      <span>Нет</span>
    </label><label class="iPhoneCheckLabelOn" style="width: 0px;">
      <span style="margin-left: -42px;">Да</span>
    </label><div class="iPhoneCheckHandle" style="width: 33px;">
      <div class="iPhoneCheckHandleRight">
        <div class="iPhoneCheckHandleCenter"></div>
      </div></div></div></div></div>
        <div class="control-group">
            <div class="controls">
              <button type="submit" name="submit" class="btn btn-primary">Вхід</button></div> </div>
    </form>
```

Отримаємо форму авторизації, зображену на рисунку 1.28.

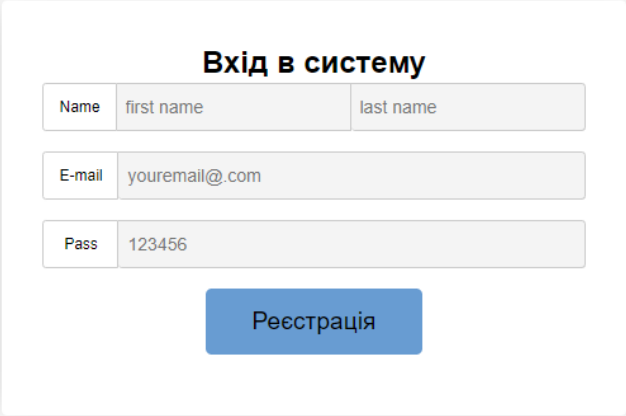


Рисунок 1.28 – Форма авторизації для користувачів системи

Адміністративна частина має більше можливостей. Вона повинна мати можливість не лише монітурити весь стан сайту, роботу модулів, дії користувачів, а й статистику роботи системи за певні періоди часу, а також можливість відстежування помилок роботи системи [21].

Адміністративна панель є основною складовою системи, доступ до якої здійснюється за допомогою переходу по адресі – «/admin». Ця панель використовує складну систему авторизації та прав доступу, щоб забезпечити систему від несанкціонованого доступу. Така система забезпечує повний контроль над кожною одиницею проекту, вона є зрозумілою і наочно простою, що максимально зменшеє початковий поріг входження. Головне вікно адміністративної панелі зображено на рисунку 1.29:

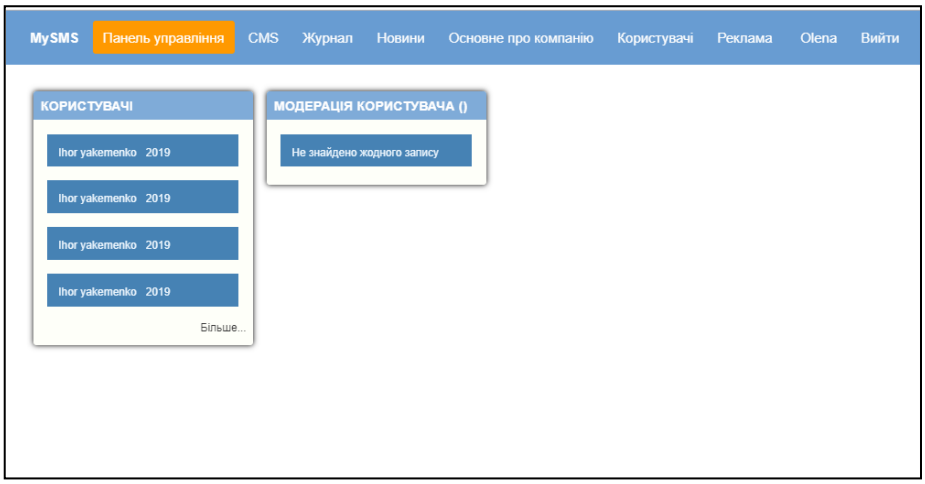


Рисунок 1.29 – Адміністративна панель системи

Сторінка для адміністрування зображує в основному пункти головного меню, які надають доступ до керування наступними складовими: загальні налаштування спеціалізованої системи на базі CMS, керування користувачами, динамічними сторінками, журналами, новинами, рекламою. Також є можливість переглянути вікно стану програмної CMS системи в реальному часі, на якому відображаєтсья іноформація про основкі складові спеціалізованої системи. Подальший функціонал CMS системи знаходиться на стадії розробки.

2 СПЕЦІАЛЬНА ЧАСТИНА

2.1 Моделювання мережевого контенту у спеціалізованій програмній CMS системі

Вхідна інформація функціонування спеціалізованої програмної систем керування мережевим контентом є свідченям призначення та умов роботи системі. Вона визначає головну мету моделювання. Вона також дає можливість сформулювати вимоги до системної формальної моделі S та моделей управління контентом. Модель спеціалізованої програмної системи керування контентом представлена формулою:

$$S = \langle X, C, V, H, Function, T, Y \rangle \quad (2.1)$$

де X – вхідні дані в системи, C – вплив на контент потоку в системі, V – вплив навколишнього середовища, H – внутрішній параметр системи, T – час транзакції управління вмістом.

Процес функціонування спеціалізованої програмної системи керування контентом, описаний функцією:

$$y_i\,(t_i + \Delta t) = Function(x_i, c_r v_l, h_k, t_i), \quad (2.2)$$

де x_i – запит користувача до системи. Характеристика компонента y_i за даними Google Analytics – кількість відвідування за певний період часу Δt, середній час перебування в Інтернеті (хв: с), показник відмов (%), досягнуто мети, динаміка (%), загальна кількість переглянутих сторінок, кількість переглядів сторінок за відвідування, нові відвідування (%), загальний кількість унікальних відвідувачів, джерела трафіку у% (пошукові системи, прямий трафік або інші сайти). Вплив значень $c_r v_l, h_k$ на y_i на

результат роботи спеціалізовану програмну системиу керування вмістом невідомо і не досліджено. Важливим та актуальним є вивчення динаміки потоку мережевого контенту та побудови моделей обробки інформаційних ресурсів в системах керування контентом.

Життєвий цикл мережевевого контенту зображено на рисунку 2.1 у вигляді наступних основних процесів:

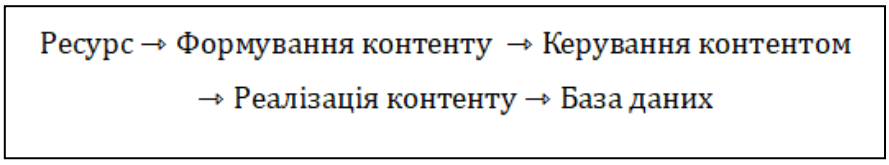


Рисунок 2.1 – Життєвий цикл мережевого контенту

Модель спеціалізованої програмної системи керування контентом:

$$S = \langle X, Formation, C, Management, Realization, Y \rangle, \qquad (2.3)$$

де $X = \{x_1, x_2 \ldots \ldots x_n\}$ – набір вхідних даних, $Formation$ – оператор формування контенту, $C = \{c, c_2 \ldots \ldots c_n\}$ – множина контенту, $Management$ – оператор керування контентом, $Realization$ – оператор реалізації контенту, $Y = \{y, y_2 \ldots \ldots y_n\}$ – набір вихідних даних.

Нижче наведена класифікація моделей управління мережевим вмістом.

1. Сторінки, що генеруються за запитом, зображеко на рисунку 2.2 у вигляді наступних основних етапів з'єднання:

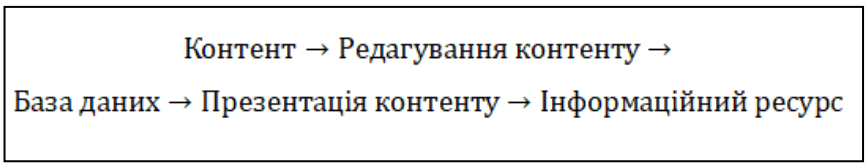


Рисунок 2.2 – Класифікація ЖЦ контенту

Сторінки генерують модель на вимогу

$$Management_Q = \langle S, C, Q, R, Edit, Y \rangle, \quad (2.4)$$

де X – вхідні дані в системи, С – набір контенту, Y – створений набір сторінок, Q – набір запитів, R – формулювання та подання сторінок, $Edit$ – редагування контенту і функція оновлення.

2. Модель генерування сторінок під час редагування зображується на рисунку 2.3:

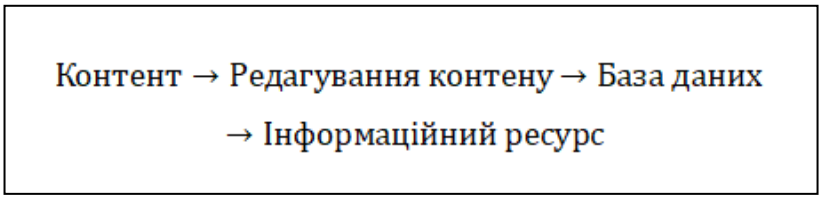


Рисунок 2.3 – Класифікація ЖЦ контенту

При внесенні змін у вміст сайту створюється статичний набір сторінок. Не враховується інтерактивність сайту між відвідувачами та вмістом. Модель системи генерації сторінок під час редагування як:

$$Management_E = \langle C, Edit, Y \rangle, \quad (2.5)$$

де С – набір контенту, Y – створений набір сторінок, $Edit$ – редагування контенту і функція оновлення.

Аналіз запитів користувачів дозволяє якісно оцінити потік контенту в спеціалізованій програмній системі. Це полегшує наступні рішення модератора: опис проблемної ситуації та пошук мети дослідження; точне визначення об’єкта та предмета дослідження; попередній аналіз об'єкта; поняття істотного уточнення та емпіричної інтерпретації; опис процедур

реєстрації властивостей та явищ; визначення загального плану дослідження; визначення типу вибірки, збору джерел.

2.2 Робота розробленої спеціалізованої програмної системи керування контентом

Спеціалізована система керування контентом веб-сайтів – це система розширеної функціональності для організації інформаційних ресурсів у мережі, архітектура якої має три рівні ієрархії (див. рис. 2.2).

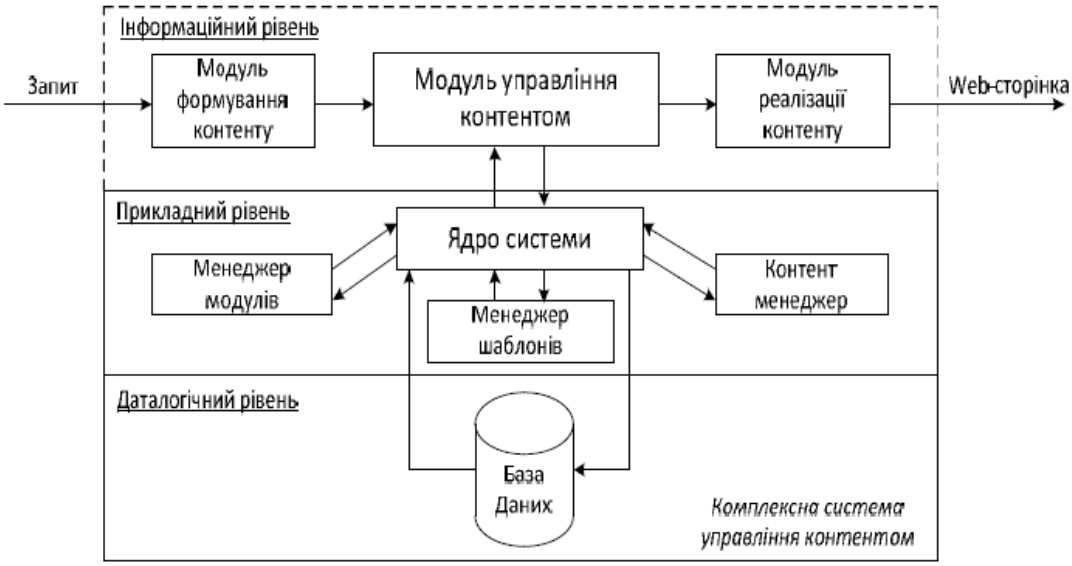


Рисунок 2.2 – Архітектура спеціалізованої CMS системи

Архітектура розробленої спеціальної програмної системи складається з прикладного, інформаційного й логічного рівня (див. рис. 2.3).

Рисунок 2.3 – Архітектура спеціалізованої програмної системи на прикладному рівні

Це надає можливість впроваджувати незалежність збереженої інформації від додатків, що їх використовують (див. рис. 2.4).

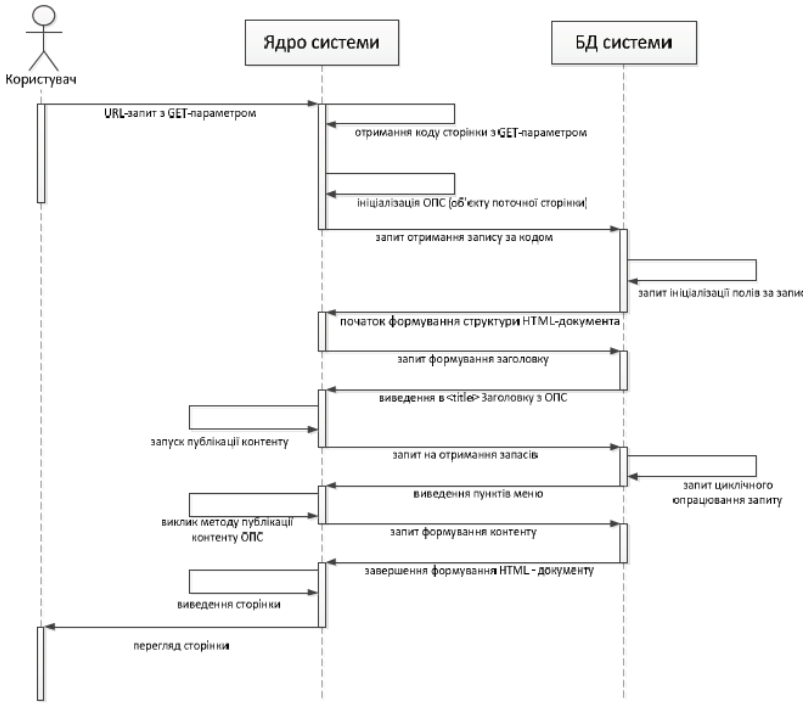


Рисунок 2.4 – Діаграма процесу подання контенту

Якщо необхідно, збережені дані переміщують на інші носії інформації з реорганізацією їх фізичної структури, змінюючи фізичні моделі даних зі збереженням змісту і логічної структури даних. Перспективи розвитку електронної діяльності спонукають до створення нових продуктивних моделей, архітектури, функцій подання контенту та різноманітних механізмів для полегшення створення веб-сайтів користувачами.

Контент-менеджер є об’єктом керування інформацією, за допомогою якої легко імплеменнувати публікації різного призначення, а також додаткові атрибути до них. Інформаційний контент може мати багато різноманітних форм: структуровані й неструктуровані: повідомлення в режимі реального часу, мультимедія, документовані дані, веб-контент, дані додатків [19]. Для керування інформацією й для аудиту відповідно до встановленої політики – вимоги масштабування для збережених даних, все частіше вимагають надання доступу до цієї інформації на вимогу. Сервісний об’єкт з керування контентом надає оптимальний функціонал для керування даними: документами, робочі процеси та інтеграції до них, звітами, цифровою інформацією, правами доступу, поштою й інформаційними архівами, записами, змістом транзакцій, архівами, а також їхньої інтеграції.

Поділ візуального частини веб-вмісту й її заповнення – основна мета спеціалізованих програмних систем управління мережевим контентом. При розробленні сайту за допомогою такої системи створюється набір сторінкових шаблонів, в яких пізніше публікується інформація. В такому випадку розробникіи обмежується тільки створенням "початкову" ІС на базі спеціалізованої системи управління мережевим контентом, потім користувачі, тобто автори вмісту і дизайнери шаблонів, самі публікують необхідну інформацію і шаблонізують її. Управління веб-сайтом, адміністратором, зводиться тільки до керуванням користувачами.

Набір засобів спеціалізованої програмної системи підтримує весь життєвий цикл контенту, а саме процес створення, публікації й постійного

оновлення інформації. Це дає змогу організаціям публікувати інформацію і документи у Інтернет. Використовуючи спеціалізовану програмну систему для керування мережевим контентом, можна швидко і безпечно на різних каналах, публікувати інформацію. Завдяки цьому підриємства можуть перекласти роботу публікації інформації на власників контенту. Унікальність використання такої спеціалізованої програмної системи полягає в тому, що сервісам не потрібно повторно вносити зміни в дані, отже, контент містить менше помилок.

Спеціалізована програмна система для керування мережевим вмістом передбачає створення та редагування вмісту в межах певного процесу публікування інформації, доставлення й адміністрування інформації для створення веб-презентацій, автоматичне трансформування контенту під різні типи представлення, надійне розмежування доступу до публічної та непублічної інформації.

Менеджер шаблонів виконує важливі задачі для побудови форм і архітектури під час функціонування самого веб-сайту. Структурно шаблон сайту складається з логічної й технічної частини. Створення гнучко налаштованого веб-шаблону – це особливість логічної частини [17].

Система керування контентом надає гнучке налаштування шаблонів зображення до наявного і нового контенту, розширюваність системи за рахунок додаткових плагінів і модулів, оновлення і відповідність сучасним стандартам, підтримка інформаційної залежності та послідовність рішень і задач, наприклад може додати статтю, але вона не буде опублікована доти, доки коректор не відредагує її, розмежування і контроль доступу до інформації на основі створення користувацьких ролей, управління документообігом – створення, перевірка, публікація, архівація і видалення документів, візуалізація контенту до розміщення на сайті, відображення контенту різними мовами [18]. Основні властивості системи керування контентом:

1) Конструктор меню. Конструктор меню дає змогу керувати різними типами меню і додавати нові будь-якого типу. Можна створювати підменю до того меню, що вже є в необмеженій кількості. Також є можливість легко і просто редагувати назву меню, видаляти, копіювати чи переміщати його.

2) Редактор контенту. Редактор дає змогу редагувати контент будь-якої сторінки згідно з потребами. За його допомогою можна створювати новий контент веб-сторінки. Ще можна форматувати контент відповідно до потреб, вставляти малюнки, посилання і flash-ролики на сторінку.

3) Адресація сторінок. Ця функція дає змогу приєднати сторінку до будь-якої іншої. Після цього ця сторінка матиме той самий контент. Є можливість відмінити аресацію і відновити оригінальний контент.

4) Властивості сторінки. Керування різними властивостями сторінки. Вони дають змогу встановлювати назву сторінки, ключові слова та мету тегу. Також можна встановлювати дату публікації і дату відміни публікації. За потреби можна сховати визначену сторінку з меню і зробити її доступною тільки у разі прямого посилання на неї.

5) Менеджер для керування веб-контента сайтів. Генерування сторінок за запитом. Такі спеціалізовані системи працюють на основі зв’язків, організованих за схемою, що зображена на рисунку 2.5:

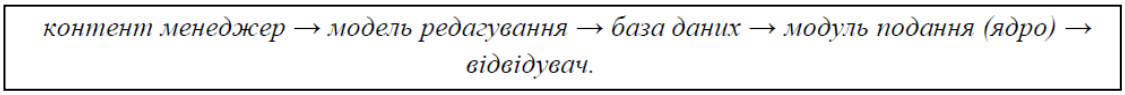


Рисунок 2.5 – Схема зв’язків спеціалізованої CMS системи

Кожна сторінка сайту має свій «URL-ідентифікатор». Веб-сторінка сайту містить: інформацію – загальний контент, навігацію, елементи сайту – шапка, підвал, елементи навігації – меню й додаткові посилання. Для будь-якої сторінки веб-сайту наприклад: інформаційної сторінки, головної сторінки, товару, сторінки розділу, новини – такі атрибути є базовим.

Сторінки-контейнери – це сторінки, головною задачею яких є подання списку посилань, наприклад, список анотованих посилань, на дочірні сторінки. Після інформаційних сторінок вони займають друге місце за частотою застосування у веб-ресурсах [20].

Публікації веб-сторінки поділяються на власне сторінку, гіперпосилання, гіперпосилання всередині контенту на іншу сторінку, текстом якого є заголовок сторінки, анотоване посилання.

2.3 Функціювання бази даних спеціалізованої системи типу WordPress

SQL означає структуровану мову запитів. Її використовується для зв'язку з базою даних. За даними ANSI (Американський національний інститут стандартів), це стандартна мова для систем управління реляційними базами даних. Операції SQL використовуються для виконання таких завдань, як оновлення даних у базі даних або отримання даних з БД. Деякі поширені системи керування реляційними БД, що використовують SQL: MySQL, Oracle, Sybase, Microsoft SQL Server. Хоча більшість систем баз даних використовують SQL, більшість з них також мають свої додаткові розширення, що використовуються лише у їхній системі . Однак стандартні команди SQL, такі як "Вибрати", "Вставити", "Оновити", "Видалити", "Створити" та "Видалити", можна використовувати для виконання майже всього, що потрібно робити з БД.

ODBC – стандартний інтерфейс програмування додатків для доступу до систем управління базами даних (СУБД). Розробники ODBC мали на меті зробити його незалежним від систем баз даних та операційних систем [36]. Архітектура ODBC зображена задопомогою чотирьо компонентами на (див. рис.2.6).

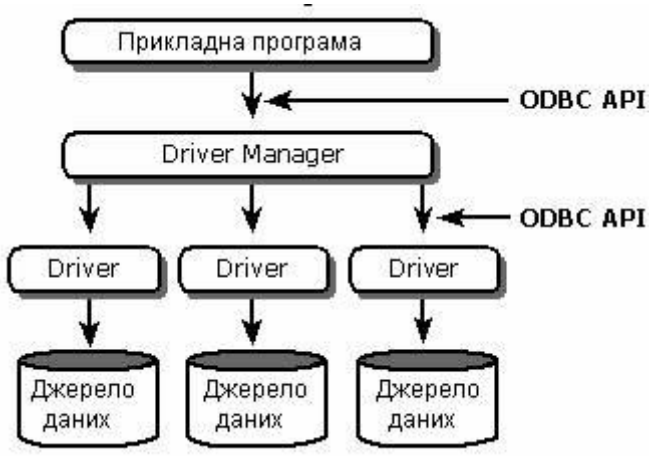


Рисунок 2.6 – Архітектура ODBC

1. Клієнт виконує функцію ODBC.
2. Менеджер драйверів обробляє виклики ODBC-функцій або передає їхньому драйверу.
3. ODBC-драйвер, передає SQL-серверу виконуваний SQL-оператор, а додатку-клієнтові – результат виконання викликаної функції.
4. Джерело даних, обумовлений як конкретна локальна або вилучена БД.

Клієнт – це будь-яка програма, сумісна з ODBC, наприклад Microsoft Excel, Tableau, Crystal Reports, Microsoft Power BI, або подібні програми такі як: Електронна таблиця, текстовий процесор, доступ до даних та витягуваний інструмент. Додаток, що підтримує ODBC, виконує обробку, передаючи заяви SQL та отримуючи результати від менеджера драйверів ODBC.

Менеджер драйверів ODBC завантажує та вивантажує драйвери ODBC від імені програми. Платформа Windows поставляється з диспетчером драйверів за замовчуванням, в той час як платформи, що не мають вікон, можуть використовувати ODBC Driver Manager з відкритим кодом, наприклад unixODBC та iODBC. Менеджер драйверів ODBC обробляє виклики функції ODBC або передає їх драйверу ODBC та вирішує конфлікти версій ODBC.

Драйвер ODBC обробляє виклики функції ODBC, подає запити SQL до конкретного джерела даних та повертає результати в додаток. Драйвер ODBC також може змінювати запит програми так, щоб запит відповідав синтаксису, підтримуваному пов’язаною базою даних. У Simba Technologies доступна основа для легкої побудови драйверів ODBC, як і драйвери ODBC для багатьох джерел даних, таких як Salesforce, MongoDB, Spark. Джерелом даних може бути файл, певна база даних в СУБД або навіть живий канал даних. Дані можуть бути розташовані на тому ж комп’ютері, що і програма, або на іншому комп'ютері десь у мережі.

Технологія ODBC дозволяє отримати максимальну сумісність: одна програма може отримати доступ до багатьох різних систем управління базами даних. Ця сумісність дозволяє розробляти, компілювати та відправляти додаток, не орієнтуючись на конкретний тип джерела даних. Потім користувачі можуть додати драйвери бази даних, які пов'язують додаток із системами управління базами даних за власним вибором. Це дозволяє розробникові створювати та поширювати додатки "клієнт-сервер" без урахування особливостей конкретної СУБД, в результаті одно і те ж застосування дістає можливість доступу до баз цих різних СУБД, що підтримують мову SQL. Подібні функціональні можливості технології ODBC дозволяють розробляти додатки СУБД різного типу. Менеджер драйверів динамічно завантажує необхідний драйвер для сервера бази даних, до якого підключається програма. Драйвер, у свою чергу, приймає виклик, відправляє SQL до вказаного джерела даних (бази даних) і повертає будь-який результат..

Функції ODBC в PHP, зазвичай звані загальними функціями ODBC, не тільки забезпечують типову підтримку ODBC, але і дозволяють працювати з деякими СУБД, що володіють власним API, через стандартний ODBC API.

Перед тим як звертатися до ODBC-сумісної базі даних із запитами, необхідно спочатку встановити з нею зв'язок. З'єднання створюється функцією ocbc_connect ().

Використання функції odbs_connect() описано у лістингу 2.3:

Лістинг 2.3 – Використання функція зєднання з базою даних

```
<? odbc_connect("myAccessDB", "user", "secret") or die( "Could
                not connect to ODBC database"); ?>
```

Після завершення роботи з ODBC-сумісної бази даних необхідно закрити з'єднання, щоб звільнити всі ресурси, використовувані відкритим з'єднанням. З'єднання закривається функцією odbc_close(). Функції реалізується за допомогою коду описаного в лістингу 2.4:

Лістинг 2.4 – Функція закриття з'єднання з базою даних

```
<? $connect=odbc_connect("myAccessDB", "user", "secret") or
die("Could not connect to ODBC database"); print "Currently
connected to ODBC database!"; odbc_close($connect); ?>
```

Запит на створення таблиці в базі даних описано в лістингу 2.5:

Лістинг 2.5 – Створення таблиця в базі даних системи

```
<? $connect = odbc_connect("myAccessDB", "user", "secret")
or die("Could not connect to ODBC database"); $query = "SELECT *
FROM customers"; $result = odbc_exec($connect, $query) or
die("Couldn't execute query!"); odbc_result_all($result,
"BGCOLOR='#c0c0c0' border='1' "); odbc_close($connect); ?>
```

Оператор на створення таблиць реалізується за допомогою коду описаного в лістингу 2.6:

Лістинг 2.6 – SQL-оператор на створення таблиці

```
CREATE TABLE user
(Id – INTEGER,
Username – VARCHAR(250) ,
Password – VARCHAR(100),
Firstname – VARCHAR(250),
Lastname – VARCHAR(250),
EmailId – VARCHAR(250),
Type – VARCHAR(45),
Created_date – DATE,
Status – INT);
```

2.4 Тестування програмної системи

Функціональне тестування – це тестування програмного забезпечення з метою перевірки можливості реалізовування функціональних вимог, тобто здатності програмного забезпечення в певних умовах вирішувати завдання, потрібні користувачам. Також за допомогою функціонального тестування перевірена працездатність посилань. Результати тестування збіглися з очікуваними.

Тест № 1 – Перехід за посиланнями меню Користувачем. Вхідні дані: користувач: переглядає всю інформацію контенту на головній сторінці; переходить на довільну вкладку головного меню; натискає на довільні гіперпосилання. Очікуваний результат: користувач повинен без труднощів знайти потрібну інформацію, переходячи по посиланнях в меню. Отриманний результат: користувач без утруднення знайшов потрібну інформацію.

Тест №2 – Публікація статті Модератором. Вхідні дані: користувач заповнює поля: Заголовок; анотація; основний текст і зображення; автор; натискає на кнопку «Опублікувати» Очікуваний результат: вікно з повідомленням «Стаття опублікована! посилання"; на головній сторінці та в розділі «Статті» з'явилася тільки що опублікована стаття. Отриманний результат: з'явилося вікно з повідомленням «Стаття опублікована!

посилання"; на головній сторінці та в розділі «Статті» з'явилася тільки що опублікована стаття. Тест пройдений.

Тест № 3 – Видалення і відновлення сторінок Адміністратором. Вхідні дані: існує сторінка, користувач виробляє дії: в розділі «Сторінки» CMS видаляє сторінку; після отримання повідомлення про видалення, відновлює сторінку з кошика. Очікуваний результат: сторінка поміщається в кошик і зникає з меню веб-сайту; після відновлення сторінки з кошика, вихідна сторінка з'являється в меню і веб-відображенні сайту в колишньому вигляді. Отриманний результат: сторінка помістилася в кошик і зникла з меню в веб-відображенні сайту; після відновлення з кошика сторінки, вихідна сторінка з'являється в меню і веб-відображенні сайту в колишньому вигляді. Тест пройдений.

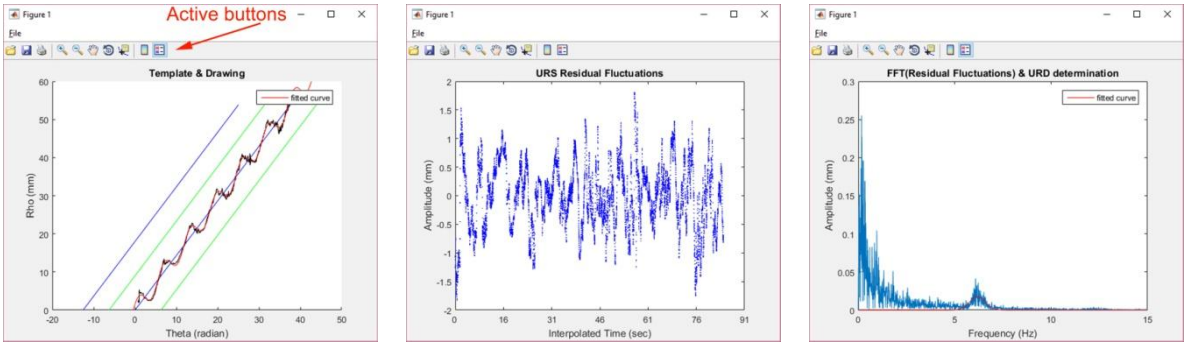
Тестування програмного забезпечення – це процес оцінювання системи або її компонента з метою виявити, чи задовольняє ПЗ вказаним вимогам. Тобто тестування – це процес виконання системи з метою ідентифікації будь-якого дефекту, помилки або невідповідність вимоги програмної системи до бажаної.

В IT сфері, великі компанії мають команду, відповідальну за оцінку розробленого програмне забезпечення в контексті заданих вимог. Більше того, розробники також проводять тестування, яке називається модульним тестуванням. У більшості випадків в цьому беруть участь наступні професіонали: тестер програмного забезпечення, Розробник програмного забезпечення, менеджер проекту, кінцевий користувач.

Іншим видом тестування є модульне тестування.Модульне тестування перевіряє функціональність і шукає дефекти в окремих частинах додатка, які доступні й можуть бути протестовані окремо модулем програми, об'єктом, классом та функцією. Модульні тести розробляються в процесі розробки ПЗ розробниками та, іноді, тестувальниками білої скриньки.

Системне тестування тестує повністю інтегровану систему, щоб переконатися, що вона відповідає її вимогам. В процесі системного тестування спеціалізованої системи (див. рис. 2.7) було протестовано основні властивості системи: серверна пам'ять (див. рис.2.7, а), продуктивність процесора (див. рис.2.7,а) і швидкодії мережі (див. рис.2.8, в).

В ході роботи підвищено рівень продуктивності запропонованих мережевих алгоритмів, що скорочує час виконання поставленої задачі на 11%.



а) Серверна пам'ять б) Продуктивність процесора в) Швидкодія мережі

Рисунок 2.7 – Робота спеціалізованої CMS системи

3 ОРГАНІЗАЦІЙНО-ЕКОНОМІЧНА ЧАСТИНА

3.1 Загальний підхід до визначення економічної ефективності розробки

Обов’язковою складовою частиною будь-якого інжинірингового проекту, є фінансові витрати на різних етапах виконання робіт. Відповідно, важливо вірно здійснити фінансову оцінку передбачуваних витрат, продуктивність, корисність та, в результаті, економічну ефективність проекту.

Оцінка вартості дослідницьких розробок базується на витратному підході: використанні первісної вартості об'єктів, виходячи з фактичних витрат на розробку та доведення до комерційного використання з урахуванням амортизації.

Згідно Статті 8 Закону № 3792-12 передбачено, що твори наукового характеру та комп’ютерні програми є об’єктами авторського права [27]. У разі реєстрації виконаної роботи як інтелектуальної власності та авторського права у державній службі інтелектуальної власності України необхідно буде сплатити збори за державну реєстрацію (518,5 грн.).

Для отримання доцільності розробки такої спеціалізованої програмної CMS системи типу WordPress потрібні відповідні затрати на дослідження та розробку. Це і становитиме основу витрат, які будуть здійснені протягом підготовки та виконання реалізації даного рішення. Уявно модель витрат можна поділити на дві основні частини: витрати, пов’язанні на дослідження предметної області, побудову математичних моделей, отримання попередніх результатів експериментів, та частину реалізації програмної системи, архітектури та тестування. Враховуючи залежність якості кінцевого продукту від кваліфікації програмістів, потрібно сконцентрувати увагу на якості та результативності розробки. Для цього потрібно провести ґрунтовний аналіз

предметної області, залучити найновіші та інноваційні технології, провести ґрунтовне тестування та оцінку результатів розробки. Співпраця з програмістами, як зареєстрованими,так і не зареєстрованими підприємцями, здійснюється на підставі цивільно-правового договору, найчастіше – договорами підряду. Щодо оподаткування виплат за договором підряду, то все залежить від того, чи зареєстрований виконавець підприємцем. Важливим етапом розробки спеціалізованої програмної системи є її тестування, яке виконується тестером за певну винагороду.

3.2 Розрахунок вартості процесу розробки та оцінка економічної ефективності проекту

Для оцінки нематеріальних активів використовують міжнародні стандарти оцінки розрахунку вартості об’єктів інтелектуальної власності, розроблені TIAVIS (The International Assets Vabulation Standarts Commitie).

Виконання розробки програмного забезпечення з огляду економічної моделі можна виконувати двома способами: процедурним та об’єктно-орієнтованим. Обидва підходи потребують залучення ресурсів у вигляді програмісті-розробників, тестерів, керівника проекту, наукового ресурсу. Різниця виникає в самій схемі розробки, тривалості періоду розробки та відповідній вартості. Процедурний підхід для розробки ПЗ в основі якого лежать процедури і функції передбачає розробку ПЗ як монолітного композиту, що в подальшому, як правило, вимагає великих витрат на супровід та модернізацію. Об’єктно-орієнтований підхід, що ґрунтується на основі об’єктів певних класів, що описують певну область, описують певну поведінку (методи) та володіють властивостями (атрибутами), орієнтовані на варіанти використання та покроковий процес розробки [28].

Для початку робіт необхідно скласти технічне завдання на розробку, яке є основним документом, що регламентує подальшу роботу, та містить

докладний опис необхідних функцій програми, інтерфейс, технології, інше. Вартість складання технічного завдання переважно складає до 10% від планованої вартості розробки. Роботу зі складання технічного завдання веде керівник проекту разом із програмістами та консультуючись із замовником.

Усі програмісти, що працюють у штаті підприємства-розробника мають встановлено певний посадовий оклад. Місячний оклад, денна заробітна плата, трудомісткість (днів) і основна заробітна плата кожного учасника техпроцесу представлено у таблиці 3.1. Всі суми наведені в національній валюті – в гривні.

Таблиця 3.1 – Розрахункова вартість технологічного процесу розробки

Посада	Місячний оклад, грн.	Денна зар. плата, грн.	Об'єктно-орієнтований підхід		Процедурний підхід	
			Днів	Сума, грн.	Днів	Сума, грн.
Менеджер проектів	14100,00	580,00	15	8650,00	16	8900,00
Веб-розробник	18660,00	730,00	21	18130,00	25	19250 ,00
Тестер	9680,00	440,00	7	3080,00	7	3080,00
Аналітик	10450,00	475,00	10	4750,00	10	4750,00
Додаткова зар. плата 20%			38	5442,00	42	5976,00
Фонд оплати праці 36,77%				10005,117		10986,876
Всього витрат на зар. плату				42657,12		46842,88
Військовий збір 1,5%				639,86		702,64
Єдиний соціальний внесок 3.6%				1535,66		1686,34
ПДВ, 15%				6398,57		7026,43
Всього				56222,21		59571,95

Згідно вимог та прорахованої кількості необхідних ресурсів на виконання, розробку, тестування та дослідницьку роботу було отримано основні часові рамки роботи над проектом. Так для об'єктно-орієнтованого підходу загальна тривалість роботи над ПЗ становить 38 робочих днів (під робочим днем розуміється 8-ми годинний робочий день), що включає роботу програміста, який, в свою чергу, являється і керівником розробки, роботу тестера та наукового працівника. Сума витрат на заробітну плату становить 56222,21 гривень включаючи всі види додаткових оплат. Для процедурного підходу до розробки суми дещо більші, адже затрачається більше часу на розробку. Так, при використанні процедурного підходу сумарна тривалість часу розробки становить 42 робочі дні, та витрати у вигляді виплат заробітної плати становлять 59571,95 гривень.

Витрати на науково-дослідницьку роботу та здійснення розробки програмних продуктів і об'єктно-орієнтованим, і процедурним способом включають:

Основна заробітна плата:

$ЗП_{осн\ 1}$ = 37210 грн; $ЗП_{осн\ 2}$ = 39880 грн.

Додаткова заробітна плата обчислюється як $ЗП_{дод} = 0,2 \cdot ЗП_{осн}$.

$ЗП_{дод\ 1} = 0,2 \cdot 37210 = 7442$ грн; $ЗП_{дод\ 2} = 0,2 \cdot 39880 = 7976$ грн.

Нарахування на фонд оплати праці (ФОП):

$ФОП_{ЄСВ} = 0,3677 \cdot ФЗП$

$ФОП_{ЄСВ1} = 0,3677 \cdot 32652 = 10005,117$ грн;

$ФОП_{ЄСВ1} = 0,3677 \cdot 35856 = 10986,876$ грн.

Всього витрат:

$В_{ЗП1} = ЗП_1 + ФОП_{ЄСВ1} + ЗП_{дод1} = 51222,21$ грн;

$В_{ЗП2} = ЗП_2 + ФОП_{ЄСВ2} + ЗП_{дод2} = 37130,97$ грн.

З цієї суми утримуються обов'язкові відрахування на заробітну плату: єдиний соціальний внесок, який складає 3,6% від суми нарахованої заробітної плати та податок на доходи фізичних осіб, який складає 15% від

суми нарахованої заробітної плати, зменшеної на суму єдиного внеску на загальнообов'язкове соціальне страхування та податкової соціальної пільги, військовий збір у розмірі 1,5%, від суми нарахувань.

До окремих витрат також відносяться витрати на куповані вироби (матеріальне забезпечення) та спец обладнання для підтримки експерименту, накладні витрати. Витрати, що будуть супроводжувати проект розробки, порівнюватимемо в двох можливих підходах розробки.

Матеріальні витрати визначаються як добуток кількості витрачених матеріалів та їх ціни (формула 3.1).

$$M_{Bi} = q_i \cdot p_i , \tag{3.1}$$

де q_i – кількість витраченого матеріалу i-го виду; p_i – ціна матеріалу i-го виду.

Матеріальні витрати в рамках проекту наведені в таблиці 3.2. Загальна сума матеріальних витрат становить 1535 гривень.

Таблиця 3.2 – Матеріальні витрати

Найменування ресурсу	Кількість, шт.	Ціна одиниці, грн	Загальна сума, грн
Флешки	4	320,00	1280,00
Папір для друку А4, арк	500	0,20	100,00
Тонер для принтера	1	100,00	100,00
Дошка для записів	1	450,00	450,00
Перманентний маркер	4	20,00	80,00
Всього			2010,00

Розрахунок витрат на електроенергію одиниці обладнання визначаються за формулою:

$$З_в = W * T * S, \quad (3.2)$$

де W – необхідна потужність, кВт; T – кількість годин роботи обладнання; S – вартість кіловат-години електроенергії, S = 2,50 грн/кВт·год.

$З_{в1}$ =0.7 * 304 * 2.50 = 532,00 грн;

$З_{в2}$ =0.7 * 336 * 2.50 = 588,00 грн;

Розрахунок суми амортизаційних відрахувань. Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 %, вартість яких перевищує 1000 грн. і визначається:

$$A = \frac{C_Б \cdot N_A \cdot T_{ФАК}}{T_{ГОД}} \quad (3.3)$$

де $C_Б$ – балансова вартість обладнання, грн; N_A – норма амортизаційних відрахувань в рік, %; $T_{ФАК}$ – фактичний час роботи обладнання по написанню програми, год; $T_{ГОД}$ – річний робочий фонд часу, год.

У даній формулі норма відрахувань на амортизацію рівна N_A = 0,6. Балансова вартість обладнання вказана в таблиці 3.3 і рівна $C_Б$ = 22589 гривень. Річний робочий фонд часу приймемо за $T_{ГОД}$ = 2120 годин. З них реальний фактичний робочий час становить $T_{ФАК}$ = 304 години згідно об'єктно-орієнтованого підходу та $T_{ФАК}$ = 336 годин згідно процедурного підходу.

Згідно вищезгаданої формули витрати на амортизацію становлять 1943,5 гривень та 2135,3 гривень для кожного підходу відповідно.

Таблиця 3.3 – Перелік необхідного обладнання

Найменування	Кількість, шт	Ціна, грн	Сума, грн	
Комп’ютер	3	9945,00	29 835,00	
Принтер	1	3499,00	3499,00	
Середовища розробки	2	безкоштовно	безкоштовно	
Операційна система (Linux)	2	безкоштовно	безкоштовно	
Всього більше 1000 грн.			33334,00	
Всього витрат на амортизацію			2035,5	2135,3
Всього			34532,5	35 469,3

Накладні витрати пов’язані з обслуговуванням виробництва, утриманням апарату керування та створення необхідних умов праці та закупівлю ресурсів та обладнання для розробки наведені в таблиці 3.3.

Залежно від організаційно-правової форми діяльності господарюючого суб’єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників. Нехай вона буде дорівнювати 30%, що становить 9795,6 грн. для об’єктно-орієнтованого і 10756,3 грн. для процедурного підходу розробки.

Проведемо розрахунок вартості створюваного програмного продукту. Вартість продукції включає у собі собівартість і планований прибуток. Найважливішим моментом для розробника, з економічної точки зору, є процес встановлення ціни.

Можна отримати загальні значення витрат на розробку та реалізацію проекту, враховуючи всі вище описані затрати та нарахування. Цей вид витрат складається з сум витрат на оплату праці (всього витрати на оплату праці), матеріальні затрати, затрати на електроенергію, накладні витрати, витрати на обладнання, враховуючи амортизації обладнання на час виконання проекту.

Собівартість продукції – це сума грошових витрат підприємства (фірми) на виробництво і збут одиниці продукції, виконання робіт та надання послуг [27].

Повна собівартість спеціалізованої програмної системи дорівнює сумі усіх витрат на його виробництво: 87613,05 грн. використовуючи об’єктно-орієнтований підхід, 92170,85 грн при процедурному підході розробки.

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Економічна ефективність (Е) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E = \frac{П}{Св}, \tag{3.4}$$

де $П$ – прибуток, $П = В - Св$; $Св$ – собівартість.

У випадку даної розробки, маючи некомерційний проект без економічно корисного результату, можна прогнозувати, що економічна ефективність прямує до 0 у обох випадках. Однак це не є причиною для негативного економічного висновку щодо даного проекту, адже такого плану розробки приносять користь у вигляді інтелектуальних ресурсів, і, переважно, фінансуються або виконуються на замовлення організацій, зацікавлених в отриманні результатів досліджень та розробок. Фінансування не можна вважати отриманим доходом від реалізації. Однак за надходження коштів на реалізацію ззовні можна вважати ефективність проекту рівною 1 (Е = 1), що означає перекриття витрат на розробку у повній мірі, тобто фінансування.

Якщо ринкова вартість програмного продукту рівна прийнятій, то економічна ефективність визначається встановленим рівнем прибутку. Поряд

із економічною ефективністю розраховують термін окупності капітальних вкладень ($T_{ок}$):

$$T_{ок} = \frac{1}{E} \tag{3.5}$$

У нашому випадку прямого прибутку не існує. Прибуток можна прогнозувати на підприємствах чи компаній, що зацікавлені в функціоналі, що дає розроблена спеціалізована програмна система. Окупність же для розробки даного ПЗ за ефективності рівній одиниці можна вважати теж рівній 1 згідно формули 3.4.

Виходячи із експертних оцінок і складності програмної системи, приймемо величину витрат на супровід і модернізацію програмного забезпечення, створеного за об'єктно-орієнтованим методом 25% (21903,26 грн) від початкових витрат, а за процедурним – 30% (27651,26 грн).

Однак варто зауважити, що розробка спрямована на короткотривалу підтримку та не прогнозує модернізації. У разі ж необхідності розробки такого ж або суміжного ПЗ можна вважати за доцільно розпочинати розробку з початкових етапів, що потягне за собою нові витрати у повній мірі. Тобто у разі короткотермінової підтримки все ж за доцільніше обирати об'єктно-орієнтовану модель розробки, адже вартість короткотермінової підтримки згідно цієї моделі не вплине значно на сукупну вартість розробки.

Здана в експлуатацію система не завжди цілком завершена, її треба змінювати протягом терміну експлуатації. Внаслідок змін система стає більш складною і погано керованою. Об'єктно-орієнтоване представлення програми дозволяє навіть середньому програмісту швидко і ефективно супроводжувати і модернізувати програми, що значно скорочує подальші витрати на супровід і модернізацію [27].

Сумарні дані економічного розрахунку розробки даного проекту наведені в таблиці 3.4.

Таблиця 3.4 – Загальні витрати

Вид витрат	Об’єктно-орієнтований підхід, грн.	Процедурний підхід, грн.
Зарплата основна	37210,00	39880,00
Зарплата додаткова	7442,00	7976,00
Фонд заробітної плати	42652,00	45856,00
Відрахування на ФОП	11005,12	11986,88
Разом на оплату праці	51555,21	547445,95
Матеріальні витрати	2010,00	2010,00
Електроенергія	532,00	588,00
Амортизація	1943,50	2135,30
Накладні витрати	9795,60	10756,30
Обладнання	22589,00	22589,00
Разом на ін. витрати	36390,84	37598,9
Собівартість	87613,05	92170,85
Прибуток	відсутній	відсутній
Вартість розробленого ПЗ	97613,05	102170,85
Модернізація і супровід	31903,26	37651,26
Загальні витрати на розробку	129516,31	139822,11
Економія ($ЗВ_1$-$ЗВ_2$)		10305,8

Загальна вартість пропонованих робіт становить 139822,11 гривень для процедурного і 129516,31гривень для об’єктно-орієнтованого підходів розробки. В даному випадку реалізації проекту варто вибрати об’єктно-орієнтований підхід для розробки даного ПЗ, адже фінансово це більш вигідно. Також у даній методиці розробки кращі часові рамки та перспективи підтримки і модернізації. У оцінці вартості продукту варто враховувати можливість фінансування проекту, що дозволить гнучко та ефективно підійти до розробки та організації праці.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності [39].

Під час виконання дипломної роботи було розроблено спеціалізовану програмну систему керування контентом. Відповідно, в ході розроблення веб-сайтів для торгових компаній необхідно буде працювати з комп’ютерною технікою та сучасними мобільними пристроями. Тому надзвичайно важливим фактором безпеки праці є дотримання правил користування технікою, норм та правил охорони праці. Потрібно забезпечити користувачам максимально комфортні та безпечні умови для їх перебування в приміщенні та якісного, ефективного виконання поставлених завдань.

Працівники установ, де розгортається система, повинні дотримуватися правил внутрішнього трудового розпорядку, виконувати правила особистої гігієни та гігієни приміщення, володіти знаннями з техніки безпеки користування технікою, електробезпеки, пожежної безпеки та не виконувати дії, які суперечать правилам охорони праці. Для забезпечення норм охорони праці та безпеки використання програмної системи користувачі повинні проходити первинний інструктаж з охорони праці на робочому місці та бути проінформованими щодо правил користування відповідних приладів. В подальшому вони проходять повторні інструктажі з охорони праці на робочому місці раз в півріччя [38].

Основні санітарно-гігієнічні вимоги до умов праці в офісних приміщеннях, де користуватимуться розроблюваною програмною системою:

– площа, відведена на одне робоче місце має становити не менше 6 кв.м, а об’єм – не менше 20 куб.м;

– для облаштування потрібно використовувати лише ті матеріали, які пройшли відповідну сертифікацію, не містять шкідливих речовин та дозволені для використання в приміщеннях;

– ергономічне розташування робочого місця, виробничих меблів з урахуванням антропометричних характеристик людини; раціональне компонування обладнання на робочих місцях;

– достатня освітленість робочих місць;

– гарантії електробезпеки та пожежної безпеки;

– приміщення не повинно межувати з джерелами шуму і вібрації , що перевищують допустимі норми;

– щодня має здійснюватися вологе прибирання та регулярне провітрювання приміщення.

– вимоги щодо рівня неіонізуючих електромагнітних випромінювань, електростатичних і магнітних полів, а також інтенсивність потоків інфрачервоного та ультрафіолетового випромінювань встановлюються відповідно до ДСанПіН 3.3.2.007-98 і ДСанПіН 3.3.6.096-2002.

Для внутрішнього оздоблення приміщень з персональними комп‘ютерами слід обирати світлі нейтральні кольори стін. Покриття підлоги та поверхня має бути рівною, неслизькою, з антистатичними властивостями.

Основним нормативним документом, що регулює забезпечення охорони праці користувачів комп’ютерної техніки є 3.3.2.007-98 «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин». У виробничих приміщеннях та на робочих місцях з ВДТ та ПК мають бути забезпечені оптимальні значення параметрів мікроклімату – температури повітря, відносної вологості, швидкості руху повітря. Для цього приміщення, в яких розташовані комп'ютеризовані робочі місця повинні бути обладнані системами опалення,

кондиціонування, які автоматично підтримують задані параметри мікроклімату.

Згідно ГОСТ 12.1.005-88 “Повітря робочої зони [39]. Загальні санітарно-гігієнічні вимоги”, СН 4088-86 температура навколишнього середовища повинна бути в межах +18℃ та +22℃, відносна вологість повітря близько 55% швидкість руху повітря – 0,1–0,2 м/сек. Рівні позитивних і негативних іонів у повітрі мають відповідати СН 2152-80. Допустима інтенсивність шуму на робочих місцях з ЕОМ має відповідати вимогам ДСанПіН 3.3.2-007-98: оптимальна – до 45 дБ, гранична – до 60 дБ.

Потрібно створити сприятливі умови для зорової роботи, які б мінімізували втому очей, виникнення професійних захворювань та сприяли підвищенню продуктивності праці. Тому освітлення повинне відповідати вимогам ДБН В.2.5-28-2018 «Природне і штучне освітлення» [38]. Основною вимогою є необхідність створення на робочій поверхні освітленості, що відповідає характеру зорової роботи і знаходиться в межах встановлених норм. Освітлення у приміщенні має бути суміщеним. Відтак, недостача денного природнього освітлення компенсується необхідною для приміщення кількістю штучного освітлення. Як джерело штучного освітлення в приміщеннях, де встановлено комп’ютерну техніку, бажано використовувати люмінесцентні лампи. Освітленість робочого місця у горизонтальній площині на висоті 0,8 м від рівня підлоги повинна бути не менш 400 лк. Для захисту від прямих сонячних променів повинні бути передбачені сонцезахисні пристрої, жалюзі, штори. Безпосередньо при виконанні роботи з обладнанням можливе використання місцевого освітлення в комбінації з загальним.

Одним із важливих параметрів охорони праці в ході експлуатації ЕОМ є ергономіка користування. Розробляючи спеціалізовану програмну систему типу WordPresss, що містить текстову інформацію, важливо врахувати фізіологічні властивості розробників. При розробці програної системи

важливо ретельно підійти до питання вибору кольорової гами, розмірів рисунок та графікі. Так, для кращого сприймання та розрізнення інформаційного ресурсу, який подається на дисплеї через програмну систему керування мережвим контентом, рекомендовано виконати текст в темних тонах на світлому фоні, таким чином збільшується акцент на поданій інформації.

При розробці програмного продукту та в ході його експлуатації користуються лініями електромереж. Для побудови системи потрібно використовувати лише якісні та сертифіковані пристрої та засоби. Персональні комп'ютери і периферійні пристрої повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. В них, окрім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Конструкція вилки має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Усі електроприлади, згідно з ДНАОП 0.00-1.21-98, повинні бути заземленні за допомогою нульового захисного провідника.

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), мають бути надійно захищені діелектричними щитками або сітками з метою недопущення потрапляння працівника під напругу.

Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією.

Приміщення мають бути оснащені системою автоматичної пожежної сигналізації і вогнегасниками відповідно до вимог чинного законодавства України. Проходи до засобів пожежогасіння мають бути вільними. Згідно

техніки пожежної безпеки пристрої повинні розташовуватися не ближче одного метра від джерел тепла. Також на них не повинні падати прямі сонячні промені, щоб виключити можливість перегріву компонентів та вбудованих акумуляторів. Адже уже відомі приклади небезпек, які спричинили нагріті до критичних температур акумуляторні батареї [37].

Щоразу, як виникає необхідність для роботи з електронними пристроями, потрібно дотримуватися відповідних правил:

– перед початком роботи потрібно оглянути робоче місце на наявність пошкоджень, диму, неприємного запаху; перевірити правильність під'єднання обладнання до електричної мережі;

– виконувати роботу лише передбачену регламентом робіт;

– підтримувати порядок і чистоту робочого місця;

– устаткування, що використовується в офісі та працює від електромережі, повинне бути заземленим;

– при будь-яких випадках порушень роботи технічного обладнання або програмного забезпечення негайно викликати представника технічної служби з питань експлуатації обчислювальної техніки.

Забороняється:

– самостійно розбирати та ремонтувати пристрій;

– користуватися несправними пристроєм або електромережою;

– залишати без догляду увімкнуте в мережу живлення устаткування, прилади, що використовуються при проведенні робіт;

– відключення захисних пристроїв;

– попадання вологи на поверхні приладів;

– приймання їжі та напоїв на робочому місці.

Користувачі спеціалізованої програмної системи керування контентом повинні дотримуватися правил та норм поведінки з комп'ютерною технікою та портативними пристроями. Під час розробки, тестування та впровадження

системи керування мережевим контентом розробниками та інженерами були дотримано усі вимоги, норми та державні стандарти з охорони праці.

4.2 Фактори ризику і можливого порушення здоров'я користувачів ПК

Застосування комп'ютера, як свідчать медики-гігієністи України та інших країн світу тягне за собою ряд небезпек для здоров'я користувачів.

На думку багатьох вчених найбільш уразливими при роботі з ВДТ ПК є нервова, імунна, зорова, ендокринна, опорно-рухова та репродуктивна системи користувачів. Всесвітня організація охорони здоров'я (ВООЗ), Директиви Європейського союзу, Міжнародні стандарти ISO-9241 відносять комп'ютеризовані робочі місця до категорії небезпечних для стану здоров'я людини. У Німеччині робота на ПК віднесена до 10 найнебезпечніших професій для здоров'я людини.

Небезпечні та шкідливі фактори, що діють на користувача комп'ютера в процесі роботи зображено на рис 4.1.

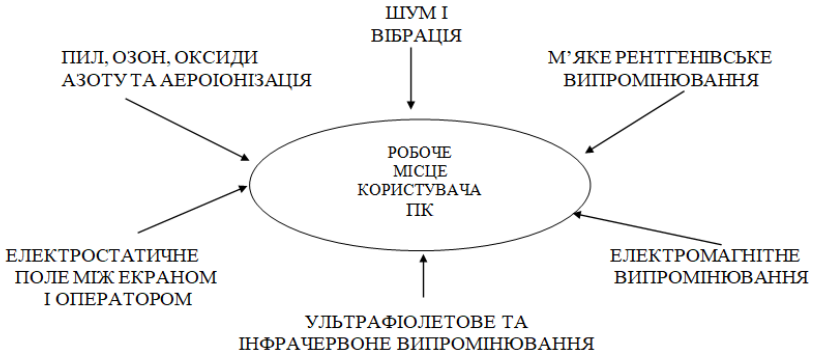


Рисунок 4.2 – Небезпечні та шкідливі фактори, які діють на користувача ПК

Дія цих небезпечних та шкідливих факторів призводить до порушення здоров'я. Причини відхилень в здоров'ї користувача ПК зображені на рисунку 4.2:

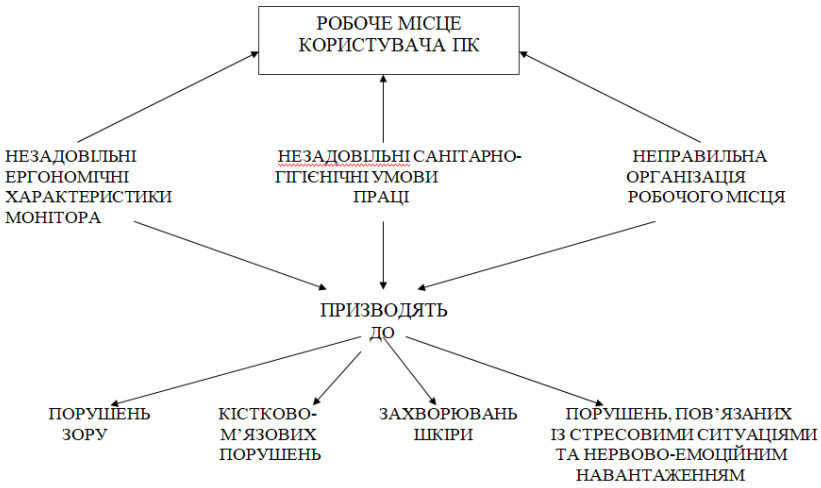


Рисунок 4.2 – Причини відхилень в здоров'ї користувача ПК

Робота користувачів ПК ДСЗ протікає в умовах нервово-емоційних, психофізичних, розумових, тривалих статичних, підвищених зорових навантажень, обмеженої рухової активності, що призводить до гострої перевтоми, нервових розладів [36].

Головним джерелом несприятливого впливу на здоров'я користувачів персональних комп'ютерів (ПК) є монітор. Поряд із ергономічними параметрами: знижений контраст зображення, дзеркальні бліки від екрану, мерехтіння зображення, небезпеку становить електромагнітне випромінювання (ЕМВ). Найбільш шкідливий у цьому плані монітор на електроннопроменевій трубці. Але випромінювати електромагнітне поле може також системний блок, схеми управління та формування інформації на рідиннокристалічних екранах та інші складові апаратури. Про механізми дії на організм ЕМВ відомо небагато. До його біологічної дії найбільш чутливими системами організму є нервова, імунна, ендокринна та статева [36]. Підвищений ризик розвитку захворювань мають підлітки — особи, які переживають період активного росту, становлення ендокринної, нервової,

серцево-судинної та інших систем. Особливо вразливими є особи, які вже мають функціональні захворювання та хронічну органічну патологію.

У процесі роботи з ПК у користувачів може розвиватися комп'ютерний зоровий синдром (КЗС), який ґрунтується на зоровому стомленні. Зорові симптоми КЗС:

- зниження гостроти зору;
- його затуманення;
- труднощі при переводі погляду з ближніх предметів на дальні і назад; відчуття змін забарвлення предметів;
- двоїння.

Очні симптоми включають:

- біль у ділянці очних ямок і лоба;
- біль під час руху очей;
- почервоніння очних яблук;
- відчуття піску за віями;
- сльозотечу;
- печіння і різь в очах.

Можливе зменшення об'єму акомодації, тимчасова короткозорість [36]. При тривалому перебуванні перед екраном внаслідок зменшення частоти моргання може розвиватися синдром сухого ока – порушення зволоження роговиці слізною рідиною. Синдром проявляється печінням в очах, почервонінням кон'юнктиви, появою судинної сітки на бічних поверхнях очей. Особливо шкідливими для органів зору є робота з яскравими дрібними елементами, читанням та введенням тексту, малюванням.

Вимушена поза з нахилом уперед вносить зміни в конфігурацію хребетного стовпа і призводить до звуження грудної клітки, що відбивається на заповненні шлуночків серця кров'ю і серцевому ритмі. Можливе погіршення живлення міокарда, що є причиною виникнення функціональних порушень серця. Порушення з боку серцево-судинної системи проявляються

у лабільності пульсу та артеріального тиску, болі у ділянці серця [36]. Нерухоме положення людини протягом тривалого часу є причиною порушення мікроциркуляції крові. Застійні явища особливо виражені на рівні органів малого таза та кінцівок. Тривалі порушення мікроциркуляції сприяють стійким змінам судин. Вимушене положення може стати чинником порушення метаболізму та гіпоксії у м'язах (насамперед, шиї, спини та плечового поясу), скривлення хребта, остеохондрозу, утрудненого дихання, тому що іонізоване повітря викликає сухість слизових оболонок, грудна клітка знаходиться у стиснених умовах, захворювань кистей рук.

Характерний розвиток карпального синдрому – оніміння та біль у пальцях кисті. На більш пізніх стадіях можливі ускладнення: тендовагініт, синдром «тенісного ліктя» – запалення загального сухожилля м'язів - розгиначів, розташованих біля ліктя, що розвивається внаслідок неправильного положення рук при роботі з клавіатурою та мишею.

Тривале перебування за комп'ютером може бути причиною порушень харчування. Це недостатнє, нерегулярне, незбалансоване харчування, наслідком якого є порушення роботи травного тракту (гастрити, виразкова хвороба шлунка та дванадцятипалої кишки, хронічні запори), вітамінна та мінеральна недостатність, або зловживання продуктами швидкого приготування, фастфудом на тлі гіподинамії, що є причиною надмірної маси тіла, метаболічних та ендокринних порушень. У користувачів ПК частіше відзначаються захворювання жовчного міхура і печінки, що виникають, очевидно, у зв'язку з частим перебуванням у вимушеній позі, при якій посилюється тиск на жовчні протоки і жовчний міхур, що викликає стаз.

Можуть виникати статеві порушення, пов'язані зі зміною регуляції з боку нервової та нейроендокринної систем. Багаторазове опромінення ЕМП викликає зниження активності гіпофіза. Відзначено специфічну дію ЕМП на статеву функцію жінок, розвиток ембріона, якість сперматогенезу, тобто репродуктивну функцію молоді .

Рекомендації щодо зменшення негативних факторів для здоров’я під час роботи на комп’ютері:

– сприятливий мікроклімат у приміщенні, тобто температура 19,5 ± 0,5 °С, вологість повітря не більше 75 %, трьохкратний обмін повітря за годину, щоденно — вологе прибирання;

– достатнє рівномірне освітлення;

– правильна організація робочого місця – площа на 1 місце не менше 6 $м^2$, об’єм — не менше 20 $м^3$, розміщення комп’ютера на відстані не менше 1 м від стін, відстань між задньою поверхнею монітора одного ПК та екраном іншого ПК не менше 2,5 м, мінімальна відстань від екрана до користувача повинна дорівнювати 50–70 см, клавіатура та руки повинні розташовуватися якомога далі від монітору, площина екрана має бути перпендикулярною нормальній лінії зору користувача, висота робочого крісла має регулюватися в межах 400–500 мм;

– встановлення оптимальних параметрів монітору – частота кадрової розгортки не менше 85 Гц для електронно-променевої трубки, для рідкокристалічних моніторів — 75 Гц, індивідуально підібрані яскравість та контрастність монітору;

– регулярні перерви під час роботи (короткочасні з використанням вправ для очей та більш тривалі із загальними фізичними вправами).

Необхідним є проведення профілактичних заходів щодо попередження про шкідливий вплив ПК. Вони повинні включати як широке інформування населення щодо ризиків безконтрольного використання комп’ютера, мережі Інтернет.

ВИСНОВКИ

В результаті проведеної роботи була розроблена система керування контентом. Система розроблена на мові PHP з використанням високопродуктивних спеціалізованних алгоритмів і програмних засобів систем керування для веб-контенту.

Наведено структурну схему програмного засобу та опис його роботи. З функціональної точки зору наведено список функцій використовуваних програмним засобом.

На етапі проектування було проаналізовано предметну область, описано поставлені завдання, обґрунтовано архітектуру системи, описано варіанти використання, діаграму послідовності та реалізацію програми. На завершальному етапі реалізації описано основні вимоги до використання розробленого ПЗ, виконано тестування готової розробки, вказано результати оптимізації. В реалізації програми наявні всі заплановані функції та можливості для роботи, реалізовано поставлені задачі.

В якості автоматизації процесу керування контенту було спроектовано та реалізовано спеціалізовану програмну систему, що містить набір методів та алгоритмів для створення, редагування та видалення контенту. Наведено опис основних функцій програми. З метою перевірки програмної системи на функціональність та продуктивність, було проведено тестування. Для більш ґрунтовного підходу до дослідження та реалізації необхідних методів було розглянуто основні параметри системи, досліджено експериментальні дані та суміжні розробки, проаналізовано існуючі програмні рішення щодо керування контенту, переваги та недоліки кожної з них.

Доцільність розробки опирається на обґрунтуванні техніко-економічних показників. Так було розраховано витрати на розробку та впровадження, підтримку проекту. З точки зору організації процесу реалізації у вигляді

програмного продукту було обґрунтовано, що об’єктно-орієнтований підхід більш прийнятний для використання в такого типу проектах.

Задля дотримання державних стандартів та норм з Охорони праці у галузі розробки програмного забезпечення, було проаналізовано нормативно-правові акти, правила та норми. Було створено та забезпечено всі необхідні умови праці для розробки та тестування програмного забезпечення з дотриманням правил експлуатації комп’ютерної техніки.

Загалом задача подання спеціалізованих інформаційних об’єктів в контексті CMS є дотичною до напрямку об’єктно-реляційного відображення ORM (Object-relational mapping) [12], об’єктно-реляційних баз даних [33] та інших напрямів, пов’язаних із моделюванням даних та знань. Тому подальші дослідження у цій галузі мають перспективу вдосконалити процес керування контентом сучасних інформаційних веб-систем та підвищити продуктивність створення та підтримки веб-ресурсів різного призначення.

Дипломна робота описує процес та основні етапи дослідницької роботи та створення програмної системи типу WordPress для трейдингу із використанням СУБД MySql.

Було досліджено системи керування контентом та бази даних для побудови спеціалізованої програмної системи. Запропоновано високопродуктивну методологію розробки спеціалізованих програмних систем керування контентом на основі використання СУБД, що значно підвищує рівень мобільності програми.

Отриманий подальший розвиток спеціалізована програмна система керування контентом, що дозволяє раціонально використовувати мережевий ресурс. Під час розробки також, підвищено рівень продуктивності запропонованих мережевих алгоритмів, що скорочує час виконання поставленої задачі на 11%;

Теоретичне дослідження ґрунтуються на застосуванні системного аналізу, об’єктно-орієнтованого проектування, теорії алгоритмів, інженерії

знань, теорії множин, теорії графів, алгебраїчної теорії і методології функціонального моделювання.

Система управління контентом є багатофункціональним інструментальним засобом, який надає розробнику спеціалізованої програмної систем широкий спектр можливостей щодо організації процесів формування, опрацювання та поширення інформаційних продуктів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мохнацька О.А. Удосконалення програмних систем керування контентом типу WordPress/ Мохнацька О.А. Кінах Я.І.// “- 2019. – 1 с.
2. Берко А.Ю. Системи електронної контент-комерції / А. Ю. Берко, В. А. Висоцька, В. В. Пасічник // Видавництво Національного університету “Львівська політехніка”. – Львів, 2009 . – 612 с.
3. Content Management Interoperability Services. Version 0.5. [Електронний ресурс] // Part I – Introduction, General Concepts, Data Model, and Services. EMC Corporation, IBM Corporation, Microsoft Corporation . – 8/28/2008. – CMIS Part I – Domain Model v0.5.pdf . – 76 p. – Назв. з екрана.
4. Ларман К. Применение UML и шаблонов проектирования. Введение в объектно- ориентированный анализ и проектирование. – М.: Изд. дом “Вильямс”, 2001.
5. Берко А.Ю. Застосування маркетингових методів для аналізу життєвого циклу комерційного web-контенту / А. Ю. Берко, В. А. Висоцька / Вісн. Нац. ун-ту «Львівська політехніка» «Комп’ютерні науки та інформаційні технології». – Львів, 2011. – № 699. – С. 3–12.
6. Клифтон Б. Google Analytics: профессиональный анализ посещаемости веб-сайтов / Б. Клифтон. – М. : Вильямс, 2009. – 400 с.
7. Локалізація програмного забезпечення [Електронний ресурс] / Wikipedia. – Режим доступу статті: http://en.wikipedia.org/wiki/ Localization. – Назва з екрану.
8. Стаття з Вікіпедії. Rapid application development. [Електронний ресурс] - Режим доступу: URL https://en.wikipedia.org/wiki/Rapid_application_development

9. Стаття у Вікіпедії. Життєвий цикл програмного забезечення. [Електронний ресурс]. – Режим доступу: URL : https://uk.wikipedia.org/wiki/Життєвий_цикл_програмного_забезечення.

10. Основы моделирования и оценки электронных информационных потоков / [Д. Ландэ, В. Фурашев, С. Брайчевский, О. Григорьев]. – К. : Інжиніринг, 2006. – 348 с.

11. Ландэ Д. Основы интеграции информационных потоков: монография / 348 Д. Ландэ. – К. : Інжиніринг, 2006. – 240 с.

12. Поспелов Д. Ситуационное управление: теория и практика / Д. Поспелов. – М.: Наука, 1986. – 288 с.

13. Локалізація програмного забезпечення [Електронний ресурс] / Wikipedia. – Режим доступу статті: http://en.wikipedia.org/wiki/ Localization. – Назва з екрану.

14. CM Lifecycle Poster [Electronic resource] / Content Management Professionals. – Retrieved 20 July 2010. – Access mode: http://www.cmprosold.org/ resources/poster/. – Title from the screen.

15. EMC. Content Management Interoperability Services. Appendices. Version 0.5 / EMC, IBM, Microsoft. – Hopkinton : EMC, 2008. – 17 p.

16. EMC. Content Management Interoperability Services. Part I. Version 0.5 / EMC, IBM, Microsoft. – Hopkinton : EMC, 2008. – 76 p.

17. EMC. Content Management Interoperability Services. Part II – REST protocol binding. Version 0.5 / EMC, IBM, Microsoft. – Hopkinton : EMC, 2008. – 79 p.

18. EMC. Content Management Interoperability Services. Part II – SOAP protocol binding. Version 0.5 / EMC, IBM, Microsoft. – Hopkinton : EMC, 2008. – 37 p.

19. Hackos J. Content Management for Dynamic Web Delivery / J. Hackos. – Hoboken : Wiley, 2002. – 432 p.

20. Halvorson K. Content Strategy for the Web / K. Halvorson. – Reading : New Riders Press, 2009. – 192 p.

21. McGovern G. Content Critical / G. McGovern, R. Norton. – Upper Saddle River : FT Press, 2001. – 256 p.

22. McKeever S. Understanding Web content management systems: evolution, lifecycle and market / S. McKeever // Industrial Management & Data Systems (MCB UP), 2003. – № 103 (9). – P. 686–692.

23. Nakano R. Web content management: a collaborative approach / R. Nakano. – Boston: Addison Wesley Professional, 2002. – 222 p.

24. Osgood C. The nature and measurement of meaning / C. Osgood // Psychological Bulletin, 49 (1952). – P. 197–237.

25. Papka R. On-line News Event Detection, Clustering, and Tracking : thesis for the degree doctor of philosophy / R. Papka. – Amherst : Massachusetts University, 1999. – 154 p.

26. Stone W. R. Plagiarism, Duplicate Publication and Duplicate Submission: They Are All Wrong! / W. R. Stone // IEEE Antennas and Propagation, 2003. – Vol. 45. – № 4. – P. 47–49.

27. Форма №2 “Звіт про фінансові результати”: методика підготовки[Електронний ресурс]. – Режим доступу: URL: http://osvita.ua/vnz/reports/accountant/17368/.

28. Методичні вказівки до виконання магістерської роботи освітнього рівня “магістр” студентами усіх форм навчання для напряму підготовки 121 – “Інженерія програмного забезпечення” / Укладачі : Петрик М.Р., Михалик Д.М., Кінах Я.І., Гладьо С.В., Цуприк Г.Б. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2016 – 26 с.

29. Рекомендації щодо розроблення навчальних планів [ТЕКСТ] / Уклад. В. П. Головенкін. – К. : Нац. техн. ун-т України «Київ. політех. ін-т», 2012. – 23 с. – 250 прим.

30. Database [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Database.

31. Introduction to the Oracle Database [Електронний ресурс] // Oracle. – 2017. – Режим доступу до ресурсу: https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm.

32. Microsoft SQL Server [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Microsoft_SQL_Server.

33. MySQL [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/MySQL.

34. DB-Engines Ranking [Електронний ресурс] // DB-engines. – 2017. – Режим доступу до ресурсу: https://db-engines.com/en/ranking.

35. Branson T. 8 Major Advantages of Using MySQL [Електронний ресурс] / Tony Branson // Datamation – Режим доступу до ресурсу: http://www.datamation.com/storage/8-major-advantages-of-using-mysql.html.

36. Правила безпечної експлуатації електроустановок споживачів [Текст] : ДНАОП 0.00-1.21-98. - Київ : Держнаглядохоронпраці, 2003. - 383 с.

37. Жидецький В. Ц. Охорона праці користувачів комп’ютерів. – Львів: Афіша, 2000. - 176 с.

38. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин (ДСан ПіН 3.3.2.007-98). – К., 1998.

39. Правила охорони праці під час експлуатації електронно-обчислювальних машин (ДНАОП 0.00-1.31-99). – К., 1999.

40. Методичні вказівки для виконання розділу дипломної роботи щодо техніко-економічного обґрунтування вибору проектного рішення розробки та оцінки якості програмного забезпечення / Упор. Петрик М.Р., Кінах Я.І., Головатий А.І., Рогатинська Л.Р. – Тернопіль: Вид-во ТНТУ ім. І. Пулюя. – 2013. – 34 с.

ДОДАТКИ

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

Міністерство освіти і науки України

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп’ютерно-інформаційних систем і програмної інженерії

Кафедра програмної інженерії

1 **ЗАТВЕРДЖУЮ**

Завідувач кафедрою

програмної інженерії

“___ “_____ 2019 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської дипломної роботи

на тему: «Розробка спеціалізованої програмної CMS системи типу WordPress із використанням СУБД MySQL для трейдингових компаній»

1.1

Керівник роботи:

к.т.н. Кінах Я. і.

“___” _____ 2019р.

_____

Виконавець:

студент групи СПм-62

Мохнацька Олена Андріївна

“___” _____ 2019р.

м. Тернопіль – 2019

1.2 **ЗМІСТ**

1 ПІДСТАВИ ДО РОЗРОБКИ

Розробка проводиться у відповідності до графіку навчального плану на 2019 рік, та згідно наказу на виконання дипломної роботи студента-магістра.

Тема проекту: «Розробка спеціалізованої програмної CMS системи типу WordPress із використанням СУБД MySQL для трейдингових компаній».

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

Система управління контентом є багатофункціональним інструментальним засобом, який надає розробнику спеціалізованої програмної систем широкий спектр можливостей щодо організації процесів формування, опрацювання та поширення інформаційних продуктів та послуг. Така система значно полегшує розміщення й опрацювання контенту в мережі.

Метою дипломної роботи є подальший розвиток спеціалізованої програмної системи керування мережевим контентом на основі використання СУБД My-SQL для трейдингових підприємств.

В якості автоматизації процесу керування контенту потрібно спроектовати та реалізувати спеціалізовану програмну систему, що містить набір методів та алгоритмів для створення, редагування та видалення контенту. З метою перевірки програмної системи на функціональність та продуктивність, провести тестування. Для більш ґрунтовного підходу до дослідження та реалізації необхідних методів необхідно розглянути основні параметри системи, дослідити експериментальні дані та суміжні розробки, проаналізовати існуючі програмні рішення щодо керування контенту, переваги та недоліки кожної з них.

За результатами виконаної роботи необхідно отримати систему керування контентом.

3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

3.1 Функціональні характеристики

Програмне забезпечення має виконувати наступні дії:

- використовувати високопродуктивну методологію розробки спеціалізованих програмних систем керування контентом;
- надавати можливість керування веб-контентом;
- візуально представляти інформаційні ресурси;
- виконуватися у будь-якому браузері;
- можливість імпортувати та експортувати базу даних;

3.2 Склад та параметри технічних засобів

1) ПК або планшетний комп’ютер з 1024 Мб оперативної пам‘яті, встановленою системою Windows XP, Vista, Seven, 8, 8.1, 10, MacOS, Android OS 5.0.5-6.0.1, IOS 10.2. Не менше 200 Мб вільного місця на жорсткому диску. Двоядерний процесор з тактовою частотою від 1.2 GHz і більше.

2) наявність доступу до браузера, сервер Apache 2.0 – HTTP, підключення до бази даних

3.3 Інформаційна та програмна сполучність

Програмний продукт повинен коректно функціонувати в різних браузерах та на різних операційних системах. Розроблювана програмна система повинна бути пристосована до використання у інформаційних системах та програмних засобах. Розробку виконувати з використанням мови програмування PHP в середовищі програмування PhpStorm з використання СУБД MySQL.

4. СТАДІЇ РОЗРОБКИ

В ходів реалізації роботи проект повинен пройти крізь наступні стадії розробки:

- аналіз предметної області;
- аналіз існуючих рішень;
- проектування архітектури та баз даних;
- реалізація класів програмної системи та баз даних;
- реалізація інтерфейсу програмної системи;
- тестування результатів розробки;
- оформлення супровідної документації;
- здача роботи.

5. ПРОГРАМНА ДОКУМЕНТАЦІЯ

Для програмного продукту повинні бути розроблені наступні документи:

- Пояснювальна записка;
- Технічне завдання;
- Презентаційний матеріал;
- Додатки.

6. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Розроблений програмний продукт має виконувати всі вимоги, що складаються з перерахованих у п. 3.1 характеристик.

Приймання проводиться спеціально створеною екзаменаційною комісією в термін до:

“_____” 2019р.

УДК 004.422.83

О.А. Мохнацька– магістрантка
Тернопільський національний технічний університет імені Івана Пулюя, Україна
Я.І. Кінах – кандидат технічних наук, доцент
Тернопільський національний технічний університет імені Івана Пулюя, Україна

УДОСКОНАЛЕННЯ ПРОГРАМНИХ СИСТЕМ КЕРУВАННЯ КОНТЕНТОМ ТИПУ WORDPRESS

O.A. Mokhnatska – magistrate, I.I. Kinakh – Ph.D, Assoc. Prof.

IMPROVING SOFTWARE SYSTEMS CONTENT MANAGEMENT TYPE WORDPRESS

В наш час стає поширеним купівля будь-яких товарів в Інтернет-магазинах. Шопінг - заняття корисне, іноді захоплююче, що приносить як позитивні емоції, так і розчарування. Уходять у минуле ті часи, коли для того, щоб стати власником бажаного предмету, потрібно було на своїх «двух» здійснювати тривалі подорожі по офлайновим торговим місцям. Тепер в наше життя активно входять інтернет-магазини, ми все частіше робимо покупки, залишаючись на робочому місці, шляхом кількох натисків миші.

CMS (від англ. Content Management System) – це система управління контентом сайту, яка включає програмне забезпечення для роботи з вмістом сайту (додавання текстів і мультимедійних файлів, створення нових сторінок і розділів, редагування контенту, зміни дизайну сайту і т. д.)[1].

CMS WordPress частіше за інших систем управління сайтами використовується користувачами для створення неперевершених сайтів з різним наповненням, змістом і обсягом. Однією з головних особливостей WordPress є структура організації бази даних. Гнучкість і функціональність зв'язків дозволяють створювати і виводити на сторінку матеріал будь-якого виду з будь-якими параметрами. WordPress - система керування контентом (CMS) з відкритим вихідним кодом, розповсюджувана безкоштовно. Написано цю систему на PHP, як система керування базами даних і використовує MySQL[2]. Область застосування цього движка досить широка. За допомогою WordPress можна створювати персональні сайти, складні новинні ресурси або навіть Інтернет-магазини.

Адміністраторська панель Wordpress - це спеціальна панель керування, що містить кілька розділів.

Сторінка містить загальну інформацію про історію даного підприємства, його послуги, кошик для онлайн покупки товарів, контакти, а також новини.

Матеріал сторінки періодично змінюється згідно зміни продукції і цін, а також ведеться постійний блог у вигляді новин.

Отже, можна зробити висновок, що програмна модернізація сайту є необхідною для сучасного користувача. Особливу увагу потрібно приділити вибору простішої системи адміністрування даних, тому перед використанням її необхідно модернізувати, щоб вони відповідали всім вимогам, були зручними і зрозумілими у використанні та встановленні не тільки в професіоналів, але і в звичайних користувачів.

Література

1. Tris Hussey Using WordPress. Б.М .: Que, 2012, - 425p. 2. Chris Coyier, Jeff Starr Digging Into WordPress v3. Б.М .: WordPress, 2012, - 442 p.
2. Крістіан Даруй, Богдан Брінзаре, Філіп Черчез-Тоза, Міхай бусика Ajax і PHP. Розробка динамічних веб-додатків. Москва: Символ, 2007 - 332 с.

Лістинг сторінки Config.php

```
<?php
ini_set( "display_errors", true );
date_default_timezone_set(   "Australia/Sydney"   );      //
http://www.php.net/manual/en/timezones.php
define( "DB_DSN", "mysql:host=localhost;dbname=cms" );
define( "DB_USERNAME", "username" );
define( "DB_PASSWORD", "password" );
define( "CLASS_PATH", "classes" );
define( "TEMPLATE_PATH", "templates" );
define( "HOMEPAGE_NUM_ARTICLES", 5 );
define( "ADMIN_USERNAME", "admin" );
define( "ADMIN_PASSWORD", "mypass" );
require( CLASS_PATH . "/Article.php" );
function handleException( $exception ) {
  echo "Sorry, a problem occurred. Please try later.";
  error_log( $exception->getMessage() );
}
set_exception_handler( 'handleException' );
?>
```

Лістинг сторінки article.php

```
<?php
/**
 * Class to handle articles
 */
class Article
{
  // Properties
  /**
  * @var int The article ID from the database
  */
  public $id = null;
  /**
  * @var int When the article was published
  */
  public $publicationDate = null;
  /**
  * @var string Full title of the article
  */
  public $title = null;
  /**
  * @var string A short summary of the article
  */
  public $summary = null;
  /**
  * @var string The HTML content of the article
  */
  public $content = null;
  /**
```

```
      * Sets the object's properties using the values in the
supplied array
      *
      * @param assoc The property values
      */
      public function __construct( $data=array() ) {
        if ( isset( $data['id'] ) ) $this->id = (int)
$data['id'];
        if ( isset( $data['publicationDate'] ) ) $this-
>publicationDate = (int) $data['publicationDate'];
        if ( isset( $data['title'] ) ) $this->title =
preg_replace ( "/[^\.\,\-\_\'\"\@\?\!\:\$ a-zA-Z0-9()]/", "",
$data['title'] );
        if ( isset( $data['summary'] ) ) $this->summary =
preg_replace ( "/[^\.\,\-\_\'\"\@\?\!\:\$ a-zA-Z0-9()]/", "",
$data['summary'] );
        if ( isset( $data['content'] ) ) $this->content =
$data['content'];
      }
      /**
      * Sets the object's properties using the edit form post
values in the supplied array
      *
      * @param assoc The form post values
      */
      public function storeFormValues ( $params ) {

        // Store all the parameters
        $this->__construct( $params );
        // Parse and store the publication date
        if ( isset($params['publicationDate']) ) {
          $publicationDate = explode ( '-',
$params['publicationDate'] );
          if ( count($publicationDate) == 3 ) {
            list ( $y, $m, $d ) = $publicationDate;
            $this->publicationDate = mktime ( 0, 0, 0, $m, $d,
$y );
          }
        }
      }
      /**
      * Returns an Article object matching the given article ID
      *
      * @param int The article ID
      * @return Article|false The article object, or false if
the record was not found or there was a problem
      */
      public static function getById( $id ) {
        $conn = new PDO( DB_DSN, DB_USERNAME, DB_PASSWORD );
        $sql = "SELECT *, UNIX_TIMESTAMP(publicationDate) AS
publicationDate FROM articles WHERE id = :id";
```

```
    $st = $conn->prepare( $sql );
    $st->bindValue( ":id", $id, PDO::PARAM_INT );
    $st->execute();
    $row = $st->fetch();
    $conn = null;
    if ( $row ) return new Article( $row );
  }
  /**
  * Returns all (or a range of) Article objects in the DB
  *
  * @param int Optional The number of rows to return
(default=all)
  * @return Array|false A two-element array : results =>
array, a list of Article objects; totalRows => Total number of
articles
  */

  public static function getList( $numRows=1000000 ) {
    $conn = new PDO( DB_DSN, DB_USERNAME, DB_PASSWORD );
    $sql      =      "SELECT      SQL_CALC_FOUND_ROWS      *,
UNIX_TIMESTAMP(publicationDate) AS publicationDate FROM articles
            ORDER BY publicationDate DESC LIMIT :numRows";

    $st = $conn->prepare( $sql );
    $st->bindValue( ":numRows", $numRows, PDO::PARAM_INT );
    $st->execute();
    $list = array();

    while ( $row = $st->fetch() ) {
      $article = new Article( $row );
      $list[] = $article;
    }

    // Now get the total number of articles that matched the
criteria
    $sql = "SELECT FOUND_ROWS() AS totalRows";
    $totalRows = $conn->query( $sql )->fetch();
    $conn = null;
    return ( array ( "results" => $list, "totalRows" =>
$totalRows[0] ) );
  }
  /**
  * Inserts the current Article object into the database,
and sets its ID property.
  */
  public function insert() {
    // Does the Article object already have an ID?
    if ( !is_null( $this->id ) ) trigger_error (
"Article::insert(): Attempt to insert an Article object that
already has its ID property set (to $this->id).", E_USER_ERROR
);
```

```
        // Insert the Article
        $conn = new PDO( DB_DSN, DB_USERNAME, DB_PASSWORD );
        $sql = "INSERT INTO articles ( publicationDate, title,
summary, content ) VALUES ( FROM_UNIXTIME(:publicationDate),
:title, :summary, :content )";
        $st = $conn->prepare ( $sql );
        $st->bindValue(        ":publicationDate",        $this-
>publicationDate, PDO::PARAM_INT );
        $st->bindValue( ":title", $this->title, PDO::PARAM_STR
);
        $st->bindValue(       ":summary",       $this->summary,
PDO::PARAM_STR );
        $st->bindValue(       ":content",       $this->content,
PDO::PARAM_STR );
        $st->execute();
        $this->id = $conn->lastInsertId();
        $conn = null;
      }
      /**
      * Updates the current Article object in the database.
      */
      public function update() {
        // Does the Article object have an ID?
        if   (   is_null(   $this->id   )   )   trigger_error   (
"Article::update(): Attempt to update an Article object that
does not have its ID property set.", E_USER_ERROR );
          // Update the Article
        $conn = new PDO( DB_DSN, DB_USERNAME, DB_PASSWORD );
        $sql        =        "UPDATE        articles        SET
publicationDate=FROM_UNIXTIME(:publicationDate),    title=:title,
summary=:summary, content=:content WHERE id = :id";
        $st = $conn->prepare ( $sql );
        $st->bindValue(        ":publicationDate",        $this-
>publicationDate, PDO::PARAM_INT );
        $st->bindValue( ":title", $this->title, PDO::PARAM_STR
);
        $st->bindValue(       ":summary",       $this->summary,
PDO::PARAM_STR );
        $st->bindValue(       ":content",       $this->content,
PDO::PARAM_STR );
        $st->bindValue( ":id", $this->id, PDO::PARAM_INT );
        $st->execute();
        $conn = null;
      }
      /**
      * Deletes the current Article object from the database.
      */
      public function delete() {
        // Does the Article object have an ID?
```

```
        if ( is_null( $this->id ) ) trigger_error ( "Article::delete(): Attempt to delete an Article object that does not have its ID property set.", E_USER_ERROR );

        // Delete the Article
        $conn = new PDO( DB_DSN, DB_USERNAME, DB_PASSWORD );
        $st = $conn->prepare ( "DELETE FROM articles WHERE id = :id LIMIT 1" );
        $st->bindValue( ":id", $this->id, PDO::PARAM_INT );
        $st->execute();
        $conn = null;
      }
    }
    ?>
```

ДОДАТОК Г – ДИСК