

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

на тему: «Автоматизована система інтеграції ресурсів мережі
інтернет-магазину»

ШЕЛЕПЕНЬ ЯРОСЛАВ СЕРГІЙОВИЧ

(прізвище, ім'я та по батькові студента повністю)

Київ 2023 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

к.т.н., доцент Гончаренко Т.А.

“ ___ ” _____ 2023 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»**

на тему: «Автоматизована система інтеграції ресурсів мережі
інтернет-магазину»

Виконав: студент II-го курсу, групи КНм-II

Спеціальності: 122 «Комп'ютерні науки»
технології»

Освітня програма: Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

Шелепень Я.С.
(прізвище та ініціали)

Керівник д.т.н., проф. Терент'єв О.О.
(прізвище та ініціали)

Рецензент к.т.н., доц. Шабала Є.Є.
(прізвище та ініціали)

Київ, 2023 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: інформаційних технологій

Освітній рівень: “магістр за ОПП”

Спеціальність: 122 «Комп’ютерні науки»

Освітня програма: «Комп’ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

к.т.н., доцент Гончаренко Т.А.

“ ____ ” _____ 2023 року

ЗАВДАННЯ

**ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»**

Шелепень Ярослав Сергійович

1. Тема роботи: “Автоматизована система інтеграції ресурсів мережі інтернет-магазину”

затверджена наказом ректора КНУБА №____ від “_____” 2023 року

2. Керівник роботи: Терентьев Александр Александрович, доктор технічних наук, професор кафедри ІТППМ

3. Строк подання студентом роботи до захисту грудень 2023 року

4. Зміст пояснювальної записки за розділами:

Р. 1. Аналіз предметної області та постановка задачі

Р. 2. Концептуальне та функціональне проектування системи

Р. 3. Проектування Баз даних інформаційного забезпечення

Р. 4. Алгоритми та програмне забезпечення системи

Р. 5. Приклад використання системи

5. Інформаційні слайди:

С. 1. Титульний слайд

С. 2. Мета, завдання

С. 5. Актуальність роботи

С. 7. Основні проблеми

С. 9. Постановка задачі

- С. 10. Дерево цілей
- С. 11. Дерево функцій
- С. 21. Приклад використання системи
- С. 22. Висновки

6. Календарний план виконання робіт:

Види робіт та їх зміст	Дата виконання
Розділ 1. Аналіз предметної області та постановка задачі	Вересень 2023 р.
Розділ 2. Концептуальне та функціональне проектування системи	Жовтень 2023 р.
Розділ 3. Проектування Баз даних інформаційного забезпечення	Листопад 2023 р.
Розділ 4. Алгоритми та програмне забезпечення системи	Грудень 2023 р.
Розділ 5. Приклад використання системи	Грудень 2023 р.
Остаточне оформлення роботи	Грудень 2023 р.
Направлення роботи на рецензування, перевірку на плагіат	02 грудня 2023 р.
Попередній захист роботи на кафедрі	05 грудня 2023 р.

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Прийом програмного продукту	к.т.н., доц. Шабала Є.Є.		

8. Дата видачі завдання: 05 вересня 2023 року

Керівник

Терентьев О.О.

 (підпис) (прізвище та ініціали)

Магістрант

Шелепень Я.С.

 (підпис) (прізвище та ініціали)

АНОТАЦІЯ

Шелепень Я.С. “Автоматизована система інтеграції ресурсів мережі інтернет-магазину”.

Атестаційна випускова робота магістра за спеціальністю: 122 «Комп'ютерні науки», освітня програма: «Комп'ютерні науки». – Київський національний університет будівництва та архітектури. – Київ, 2023.

Робота присвячена розробці інформаційної підсистема інтеграції ресурсів мережі інтернет-магазину, з максимально зручним інтерфейсом як для користувачів так і для адміністраторів.

Ключові слова: інтернет-магазин, тестування, веб-сайт, інформаційні технології, база даних, сервер, додатки.

SUMMARY

Shelepen Ya.S. “Automated system of integration of online store network resources”.

Certification master's thesis in the specialty: 122 "Computer Science", educational program: "Computer Science". - Kyiv National University of Construction and Architecture. - Kyiv, 2023.

The work is devoted to the development of an information subsystem for the integration of online store network resources, with the most user-friendly interface for both users and administrators.

Keywords: online store, testing, website, information technology, database, server, applications.

ЗМІСТ

Вступ.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	10
1.1 АНАЛІЗ ВИКОРИСТАННЯ ВЕБ-САЙТІВ У СУЧАСНОМУ ЖИТТІ.....	10
1.2 КЛАСИФІКАЦІЯ ВЕБ-САЙТІВ.....	12
1.3 ВИМОГИ ДО ІНТЕРНЕТ-МАГАЗИНУ.....	13
1.4 ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНИХ ЗАСОБІВ ПОБУДОВИ ЕЛЕКТРОННОГО МАГАЗИНУ	14
1.5 АНАЛІЗ ПЛАТІЖНИХ СИСТЕМ	16
1.6 АНАЛІЗ ІСНУЮЧИХ РОЗРОБОК	20
1.7 ОБГРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЇ РОЗРОБКИ	23
2 КОНЦЕПТУАЛЬНЕ ТА ФУНКЦІОНАЛЬНЕ ПРОЕКТУВАННЯ СИСТЕМИ.....	24
2.1 АРХІТЕКТУРА ЕЛЕКТРОННОГО МАГАЗИНУ	24
2.2 ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ	25
2.3 РОЗРОБКА ОПЕРАЦІЙНОЇ ТА СЕРВЕРНОЇ ЧАСТИН	26
2.4 ДЕРЕВО ЦІЛЕЙ ТА ФУНКЦІЙ	30
3 ПРОЕКТУВАННЯ БАЗИ ДАНИХ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ.....	34
3.1 ВИЗНАЧЕННЯ СУТНОСТЕЙ БД СИСТЕМИ	34
3.2 ПРЕДСТАВЛЕННЯ ДАНИХ У ВИГЛЯДІ АГРЕГАТИВ	36
3.3 СЛАБКІ АСІД ВЛАСТИВОСТІ	38
3.4 РОЗПОДІЛЕНІ СИСТЕМИ, БЕЗ СПІЛЬНО ВИКОРИСТОВУВАНИХ РЕСУРСІВ.....	40
3.5 ФІЗИЧНЕ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ.....	43
<u>4 АЛГОРИТМИ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ.....</u>	<u>49</u>
<u>4.1 ДЕКОМПОЗИЦІЯ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ СИСТЕМИ.....</u>	<u>49</u>
4.1.1 Zend Framework 2.....	49

4.1.2 MongoDB.....	52
4.1.3 Bootstrap.....	54
4.1.4 Git. Система контролю версій.....	55
<u>4.2 ДІАГРАМА ОСНОВНИХ КЛАСІВ СИСТЕМИ ТА ОПИС МОДУЛЬНОСТІ.....</u>	<u>59</u>
<u>4.3 ОПИС ОСНОВНИХ АЛГОРИТМІВ.....</u>	<u>60</u>
<u>5 ПРИКЛАД ВИКОРИСТАННЯ СИСТЕМИ.....</u>	<u>61</u>
<u>5.1 ВВЕДЕННЯ ПОЧАТКОВИХ ДАНИХ ДЛЯ КОНТРОЛЬНОГО ПРИКЛАДУ.....</u>	<u>61</u>
5.1.1 Магазин.....	62
5.1.2 Функціонал групових дій.....	64
5.1.3 Карточка товару.....	65
5.1.4 «Категорії».....	67
5.1.5 «Типи товарів».....	69
5.1.6 «Производители».....	71
5.1.7 «Коллекции».....	71
5.1.8 «Заказы».....	72
5.1.9 «Скидки».....	72
5.1.10 «Валюты».....	73
5.1.11 «Настройка импорта».....	73
5.1.12 «Импорт товаров».....	74
5.1.13 «Импорт картинок для товаров»	75
5.1.14 «Импорт типов товаров»	75
<u>5.2 РОЗРАХУНОК КОНТРОЛЬНОГО ПРИКЛАДУ ТА ОТРИМАННЯ ВИХІДНИХ ДАНИХ.....</u>	<u>75</u>
Висновки.....	79
<u>СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....</u>	<u>81</u>

ВСТУП

У світі, а зокрема в Україні, величезними темпами зростає кількість користувачів Інтернет, і як наслідок, кількість маркетингових компаній і потенційних «електронних» покупців.

Обрана тема вважається актуальною на сьогоднішній день, так як мільйони людей щодня, не виходячи з дому, купують різні товари в електронних магазинах.

Електронний магазин має незаперечувану перевагу перед звичайними магазинами. Він надає зручність і комфорт покупцеві зробити своє замовлення не виходячи з дому. Цей момент є істотним, при переході виробників зі «звичайної» торгівлі на «електронну».

Висока якість продукції, вміння донести інформацію про продукт до споживача і ефективна система збуту, робить підприємство успішним на ринку.

Електронний бізнес виступає, передусім, як досить складна прикладна інформаційна система.

Метою дипломного проекту є не лише розробка Інтернет-магазину для реального замовника, а й донести до слухачів важкість інтеграції кожної компоненти системи керування контентом і створення повноцінної продовольчої мережі.

Одним з найбільш вигідних бізнесів в глобальній мережі є інтернет магазин. Сьогодні в Інтернеті існує тисячі різних за тематикою і функціональним особливостям порталів, що спеціалізуються на надання торговельних послуг.

Купити в мережі на даний момент можна практично все. Часом виникають ситуації, коли інтернет надає набагато більш широке поле для вибору, ніж реально існуючі магазини і супермаркети, іншими словами в мережі можна знайти те, що в реальному часі неможливо придбати через прилавок. Якщо торкнутися теоретичної основи, то інтернет-магазин надається собою сайт або

портал, метою якого є продажі за допомогою мережі Інтернет. Користувачам надається можливість в режимі реального часу, не виходячи з дому замовити товар, вибрати спосіб доставки і оплати.

На сьогоднішній день багато користувачів мережі прагнуть самостійно створити інтернет-магазин, і таким чином поправити своє матеріальне становище, реалізувати себе як успішного бізнесмена. Однак навіть у такому зовні простому, вигідному і привабливому справі є свої підводні камені. Навіть для того, щоб просто створити інтернет-магазин з нуля необхідний стартовий капітал. Безумовно, в глобальній мережі існує маса програм і порталів, що пропонують безкоштовну і самостійну розробку проекту. Але варто розуміти, що інтернет-магазин має суттєву відмінність від інших інтернет сторінок: нехай навіть Ваш магазин знаходиться у віртуальному просторі, але Ви продаєте реальний товар!

Проведені дослідження було представлено на міжнародній науковій конференції “World science: problems, prospects and innovations”.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз використання веб-сайтів у сучасному житті

Актуальність створення інтернет-магазину сьогодні незаперечна.

Розробка і створення інтернет-магазину являє собою відмінне маркетингове рішення не тільки для організацій, що працюють для пересічного споживача, але для таких видів бізнесу, чиїми клієнтами є інші компанії.

Розробка інтернет-магазину здійснюється для прямих продажів з ресурсу. Саме тому створення інтернет магазину допомагає вирішити такі проблеми:

- скорочення числа необхідних торговельних, складських приміщень;
- скорочення числа продавців;
- суттєве розширення географії та обсягів продажів (при організації доставок в інші регіони).

Створення і розкручування інтернет-магазину обов'язково починається зі складання досить чіткого і грамотного технічного завдання (або ТЗ). Адже безпосередньо для такого типу сайту дуже важливе дотримання основних принципів юзабіліті (тобто зручність використання).

До початку створення інтернет-магазину необхідно повною мірою визначити, як саме користувач стане пересуватися по сайту, і як буде виглядати сама картка товару, а також які функції повинна містити Ваша корзина замовлень та інше[1].

Створення і розкручування інтернет-магазину зазвичай передбачає наступні пункти:

- можливості пакетного завантаження продукції з наявної бази даних;
- можливості редагування окремого товару безпосередньо з панелі управління;
- можливості прикріпити до будь-якого товару необмежене число зображень і файлів (наприклад, технічні характеристики і прайс);
- збереження для користувачів історії замовлень з можливостями здійснення повторного замовлення, змінивши число або додавши інші товари;
- повідомлення про зроблені в інтернет-магазині замовленнях безпосередньо по електронній пошті для більш оперативного зв'язку з покупцями;

- можливості самостійного редагування, додавання і видалення спецпропозицій, банерів з хітами продажів і новинками.

Розробка інтернет магазину дещо відрізняється від сайту-каталогу саме наявністю модуля кошика замовлень, можливістю реєстрації користувача.

В кошику замовлень відвідувач зможе поміняти число товарів, видалити всі непотрібні позиції, у випадку, якщо він передумав здійснювати покупку. Коли кошик остаточно заповнена, відвідувач готовий здійснити замовлення, тоді він відправляє заявку самому адміністратору сайту.

Крім того, в деяких інтернет-магазинах є можливість оплати за допомогою інтерфейсу платіжної системи, встановленої на сайті.

Погодьтеся, відносно недавно такі можливості ще не були доступні ні власникам сайтів, ні їх клієнтам. Саме тому кажуть, що актуальність створення інтернет-магазину є непорушною реальністю і істиною. Це відмінний інструмент продажів, який приносить прибуток власнику і багато зручності придбання товарів - користувачеві. На сьогоднішній день практично кожна організація має власний веб-сайт. В умовах використання сучасних інформаційних технологій – це необхідний чинник існування, що дозволяє розширити поле рекламної діяльності і привернути тим самим додаткових клієнтів.

Створення і розробка сайтів включає:

- Затвердження первинного технічного завдання на розробку сайту.
- Визначення структурної схеми сайту - розташування розділів, контенту і навігації.
- Веб-дизайн - створення графічних елементів макету сайту, стилів і елементів навігації.
- Розробка програмного коду, модулів, бази даних і інших елементів сайту необхідних в проекті.
- Тестування і розміщення сайту в мережі Інтернет.

Можемо виділити декілька основних методологій по створення сайту

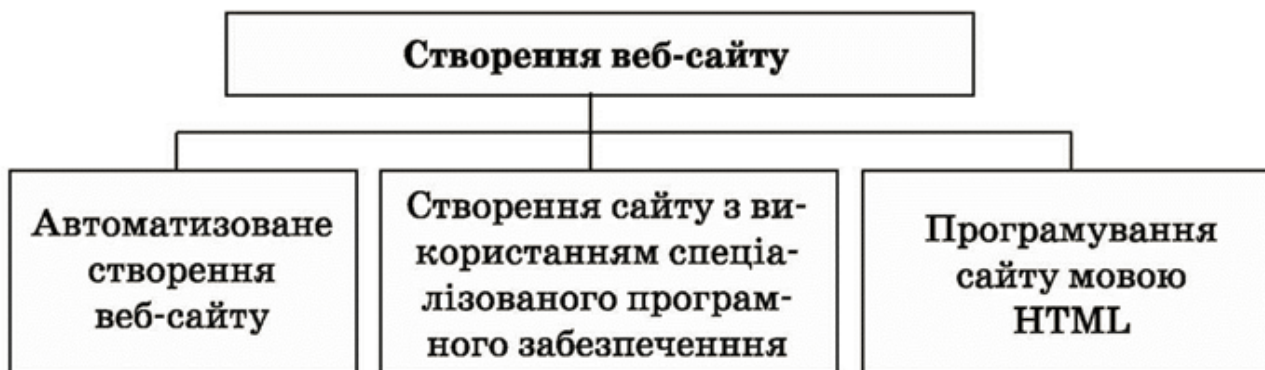


Рисунок 1.1 Методології створення веб-сайту

1.2 Класифікація веб-сайтів

В даний час у всесвітній павутині розміщено декілька мільйонів веб-сайтів і їх число постійно росте. Їх створюють для того, щоб знайти друзів по інтересах, розширити свій кругозір, свій світ[2].

Інформаційні сайти. До них відносяться сайти учбових закладів, співтовариств по інтересах, фірм і інше.

Сайти-портали. До них відносяться крупні веб-сайти, які пропонують велику кількість сервісів, починаючи від каталогу зображень до форумів і поштових скриньок.

Інтернет-магазини. Такі сайти в наш час набирають дуже велику популярність з боку, як користувачів, так і з боку підприємців. Це веб-сайти, в яких кожна людина може купити будь-який товар не відходячи від свого місця біля комп'ютера.

Соціальні сайти. Це веб-сайти на яких кожна людина має змогу спілкуватися зі старими друзями та знаходити нових.

1. Методом роздрібного продажу товарів в мережі: Інтернет-магазини; Веб-вітрини, торговельні системи; торгові ряди; контентні проекти (споживчі енциклопедії, системи Інтернет замовлень товарів і т.д.);

2. За бізнес-моделі: повністю он-лайнний магазин і суміщення оф-лайнного бізнесу з он-лайнних (коли Інтернет-магазин створюється на

основі вже діючої реальної торгової структури);

3. Взаємовідносинами з постачальниками: магазини, що володіють власним складом (наявність реальних товарних запасів); магазини, що працюють за договорами з постачальниками (відсутність значних товарних запасів);

4. За ступенем автоматизації серед торгових систем електронних магазинів розрізняють веб-вітрини, власне Інтернет-магазини і торгові інтернет-системи (ТІС).

1.3 Вимоги до інтернет-магазину

Кожен інтернет-магазин має свій власний список вимог, який дозволить йому функціонувати і бути конкурентоспроможним на ринку.

- платіжні системи, введення / висновок, конвертацію грошових коштів в електронну готівку;
- веб-панель управління вмістом;
- форма зворотного зв'язку, відгуки, опис, кошик, порівняння товару;
- автоматичне зміна розміру картинок при завантаженні, і можливість проставлення на них свого водяного знака;
- імпорт прайс-листа з Excel;
- пошук товару по артикулу, назві, опису;
- сортування товару за назвою, ціною, наявності;
- опитування і голосування для користувачів;
- статистика переглядів товарів;
- підбір товару по параметрам;
- партнерська програма (промо-коди для агентів, при введенні коду клієнт отримує знижку, а агент винагороду);
- можливість підключення платіжних шлюзів;
- реєструйте доменне ім'я на себе, замовляйте хостинг на своє ім'я;

- при верстці використовувати стандарти Schema.org (підтримуються Google, Yandex, Bing, Yahoo);
- довідкова інформація про інтернет-магазин;
- список товарів, що продаються;
- повнота опису споживчих характеристик товарів (робіт, послуг);
- опис процедури замовлення товару (роботи, послуги);
- інформація по доставці товару (роботи, послуги);
- контроль отримання замовлень своїми клієнтами;
- експортні обмеження з доставки товару (роботи, послуги);
- опис процедур повернення грошових коштів, обміну товарів.

1.4 Порівняльна характеристика програмних засобів побудови електронного магазину

Типовий варіант інтернет-магазину складається з наступних функціональних частин:

- Каталог товарів;
- Пошукова система;
- Призначена для користувача корзина;
- Реєстраційна форма;
- Форма відправки замовлення.

Каталог товарів. Каталог являє собою складну і багаторівневу структуру даних, яка повинна простим і зрозумілим способом виробляти упорядкування товарів. Найпростіше такий каталог представити у вигляді дерева об'єктів, верхній рівень якого складається зі списку розділів. Розділи можуть містити підрозділи або посилання на конкретний товар і т.д. Таке упорядкування просто необхідно для зручного та швидкого пошуку і замовлення товарів[3].

Пошукова система. Пошукова система є обов'язковим елементом динамічного каталогу й реалізується на стороні сервера. Незважаючи на те, що

каталог забезпечує упорядкування і угруповання даних, пошукова система дає користувачеві можливість швидкого пошуку інформації, що особливо важливо в тому випадку, коли каталог являє собою досить розгалужену структуру даних з великою кількістю розділів, підрозділів і товарів, користувач погано уявляє в якому розділі може перебувати цікавить його товар і чи є він в каталозі взагалі. Пошукова система в деяких випадках дозволяє значно скоротити кількість переходів між сторінками каталогу для доступу до інформації, що цікавить.

Особливість реалізації пошуку в Інтернеті полягає в тому, що тут відбувається вибірка всіх записів, які задовольняють умовам запиту (даний механізм пошуку називають пошуком з надлишком). У разі великої вибірки даних висновок результатів пошуку здійснюється посторінково для того, щоб відвідувачам не доводилося довго чекати завантаження всієї вибірки, яка може включати в себе сотні, тисячі і більше записів. Як правило, відвідувачі не переглядають всі сторінки вибірки, обмежуючись двома або трьома. Тому даний механізм пошуку в багатьох випадках працює вкрай повільно і неефективно. Однак він дозволяє здійснити вибірку однакових товарів від різних постачальників, порівняти їх параметри між собою і вибрати оптимальний варіант.

Користувацька кошик. Користувацька кошик представляє собою деякий масив даних, який служить для зберігання замовленого користувачем товару.

Реєстраційна форма. Служить для введення персональних даних користувачів. Надалі ця інформація використовується для їх ідентифікації між сеансами роботи з інтернет-магазином. Дана інформація може зберігатися як на стороні сервера, так і на стороні клієнта.

Форма відправки замовлення. Форма відправки замовлення служить для введення контактної інформації замовника і відправки її та замовлення на електронну скриньку організації.

1.5 Аналіз платіжних систем

Так як створюваний програмний додаток є електронним магазином, неможливо оминати тему оплати товару в мережі інтернет. Для початку розглянемо існуючі системи оплати в мережі Інтернет.

Традиційні методи оплати, включаючи готівку, банківські перекази, чеки, пластикові картки, винайдені задовго до виникнення електронної комерції. Тому немає нічого дивного в тому, що вони не повністю відповідають її потребам. Безумовно, при покупці звичайного «фізичного» товару гроші можна взяти з покупця при доставці. Але якщо купується «цифровий» товар або інформація, обов'язково повинен бути спосіб оплатити покупку прямо «на місці», тобто на сайті продавця. Ось чому в електронній комерції надзвичайно гостро стоїть питання розробки універсального способу оплати покупок в Інтернет, який би дозволив робити дешеві і безпечні платежі в режимі реального часу[4].

До способів оплати покупок в Інтернет, проведеним електронним шляхом, можна віднести наступні:

- Оплата електронними грошима;
- Оплата платіжними картами.
- Оплата кур'єру
- Оплата через банк

Сьогодні на ринку з'явилася прекрасна можливість оплачувати послуги або товари різних компаній через мобільний телефон. Ви можете робити покупки в Інтернет - магазинах, оплачувати комунальні платежі і навіть переводити гроші за банківськими реквізитами. Незалежно від того, що і як Ви оплачуєте, процес оплати займає всього пару хвилин, а сам платіж відбувається практично миттєво.

Оплата електронними грошима

Цифрові гроші є основою платіжної системи Інтернет. Вони можуть бути класифіковані за видами організації їх функціонування, за рівнем безпеки, а також за способом розрахунку.

Цифрову готівку можна уявити собі як файли-жетони, які замінять готівку. Продавці та покупці можуть вільно обмінюватися цими "монетами" по мережі, оплачуючи ними товари і послуги. Для зазначеної мети учасники системи встановлюють у себе на комп'ютерах особливу програму - "електронний гаманець", який забезпечує облік і передачу жетонів, а також перевірку їх достовірності. Цифрові гроші можуть необмежено довго звертатися в мережі, але також можуть в будь-який час бути обмінені на справжні гроші в організаторів системи або в банках, що беруть участь в ній.

Оплата платіжними картами

Кредитна картка - іменний платіжно-розрахунковий документ у вигляді пластикової картки, що видається банком своїм вкладникам для безготівкової оплати ними товарів і послуг у роздрібній торговельній мережі, забезпеченій комп'ютерними пристроями, які передають запит на оплату товару в банк.

Кредитна картка засвідчує наявність у її власника поточного рахунку в банку. У різних країнах використовуються кредитні картки американських кредитно-фінансових груп типу Visa, MasterCard, American Express і т.д.

Дебетові картки можуть використовуватися при оплаті товарів і послуг через Інтернет в режимі он-лайн так само, як при отриманні готівки в банкоматі: для здійснення платежу клієнт повинен ввести номер картки та PIN-код.

До покупок, оплачених не електронний шляхом можна віднести:

- Оплата кур'єру готівкою;
- Передоплата за безготівковим розрахунком;
- Оплата післяплатою.
- Оплата кур'єру готівкою

Один з найпростіших способів оплати це оплата кур'єру. Ви просто вручаєте гроші кур'єру та розписуєтесь в отриманні замовлення. На жаль цей спосіб можливий, не скрізь.

Після вибору товару в інтернет-магазині покупець розміщує замовлення, в якому вказує адресу для доставки (додому, в офіс, або кудись ще). Кур'єр

привозить товари за вказаною адресою і отримує гроші.

Переваги цього способу, напевно, можна перераховувати безкінечно:

- гарантія отримання товару (покупцем) і грошей (продавцем).

Обдурити тут, звичайно, можна, але змоги не більше, ніж у звичайному магазині.

- можливість перевірити товар (і комплектність) відразу, і повернути його, відмовившись від покупки.

- подивитися в очі живій людині, попросити його розповісти про товар, показати його в роботі.

- отримати всі необхідні документи, як-то: гарантійний талон, товарний чек.

Недоліки також очевидні:

- не завжди приємно бачити у себе вдома (або в офісі) незнайомої людини (кур'єра), спілкуватися з ним.

- найчастіше очікування кур'єра перетворюється на тортури, і відбиває всяке бажання що-небудь купувати.

Можна зробити передоплату вартості замовлення в найближчому для Вас відділенні Ощадбанку (або в будь-якому іншому) на наш. Для цього треба роздрукувати бланк, який ви отримаєте по e-mail або, який буде згенерований відразу по завершенню процедури оформлення замовлення (цей бланк, також, можна використовувати просто як зразок для заповнення типографського бланка в Банку). Як тільки гроші прийдуть на рахунок, магазини відправляють замовлення.

Оплачувана покупка при отриманні на поштовому відділенні називається оплатою післяплатою. При оплаті післяплатою, пошта стягує 8% від вартості замовлення. Порядок оплати такої:

- на вказану адресу надсилається замовлення, або заповнюється певна форма оплати запропонована інтернет - магазином.

- вказуються свої Ф.І.О., повна адреса з індексом, контактний телефон та e-mail.

- при отриманні поштового повідомлення, перераховується поштові витрати і вартість замовлення на зазначені банківські реквізити.
- далі по отриманому в банку квітку поштовими працівниками видається замовлення.

На початковому етапі в проєкті інтернет - магазину будуть реалізовані способи оплати як оплата поштовим переказом і оплата кур'єру готівкою. Так як спочатку інтернет магазин націлений на збільшення прибутку магазину, на інформаційне просвітництво потенційних клієнтів. Доставка кур'єром буде націлена на працююче населення міста, що має можливість виходу в інтернет, але не має часу відвідати магазин у зв'язку з незручним для них графіком роботи магазину. Доставка кур'єром буде здійснюватися тільки для жителів міста. Для жителів району та області, не мають можливість особисто відвідати магазин, але бажаних здійснити покупку буде надано спосіб отримання товару післяплатою.

1.6 Аналіз існуючих розробок

Аналіз аналогічних сайтів - це оцінка ряду факторів сайтів схожих за певним ключовим запитом. Розглянемо декілька сайтів на приклад:



Рисунок 1.2 Головна сторінка city.com.ua

Опишемо плюси та мінуси сайту у вигляді таблиці 1.1.

Таблиця 1.1

Плюси та мінуси сайту №1

Плюси	Мінуси
виведена вкладка новини;	зайва реклама;
коректна і чітка інформація про магазин;	не всі продукти виведені в зображення.
розділ вакансій;	
присутній готівковий і безготівковий розрахунок;	
каталог продуктів розбитий на категорії	
проста реєстраційна форма.	

Бутік "П'ятий Океан" - ваш провідник у дивовижний світ напоїв і делікатесних продуктів, у світ зі своєю культурою та історією.



Рисунок 1.3 5ocean.com.ua

Таблица 1.2

Плюси та мінуси сайту №2

Плюси	Мінуси
дизайн сайту витриманий в строгому стилі;	каталог продуктів створений у вигляді списку;
товар розбитий на великі категорії;	оплата тільки готівковим розрахунком.
оформлення подарунків;	
вкладка - поради сомельє;	
детальна контактна інформація;	
покупець може задати свої питання;	

Таблица 1.3

Плюси та мінуси сайту №3

критерії	" Супермаркет Сити " city-market74.ru	" Пятый Океан " www.5ocean.ru
навігація	3	5
організація інформації	4	5
швидкість завантаження	середня	висока
використовувані технології	PHP	JavaScript SWFObject
інтерактивність	3	5
дизайн	5	3
частота оновлення	раз в два тижня	раз в тиждень
загальне враження	Позитивне	Добре

Інтернет-магазин повинен бути зручний і робітникам, не містити зайвої інформації, а покупець міг з легкістю переглянути товар який хоче замовити. Сайт своїм виглядом повинен зацікавити чергового покупця - користувача інтернету, і після покупки товару клієнт знову повертався до сайту і радив його своїм друзям і знайомим.

Найважливішим на сайті є кошик для покупця - список товарів, обраний покупцем у магазині.

Вибір програмних засобів для розробки і створення інтернет - магазину досить широкий. Для швидкого підняття сайту краще використовувати відкриті CMS.

Спосіб оплати так само важливий для інтернет - магазину. Дуже зручним варіантом оплати є електронні гроші і оплата готівкою.

На початковому етапі роботи сайту магазин буде приймати тільки

готівковий розрахунок. Так як, з точки зору безпеки, ці платежі є найбільш безпечними.

1.7 Обґрунтування вибору технології розробки

Система управління сайтом - це інформаційна система або комп'ютерна програма, яка використовується для забезпечення і організації спільного процесу створення, редагування і управління контентом.

Для того щоб правильно вибрати систему управління для свого сайту потрібно розглянути:

- інтерфейс:
- зручності у використанні CMS;
- зовнішній вигляд повинен бути приємним, витриманим у єдиному стилі;
- подивитися основні інтерфейсні елементи управління;
- все має бути очевидно і зрозуміло, а найголовніше - зручно;
- інтерфейс не повинен бути "перевантажений" текстом і графікою, він повинен надавати тільки потрібну інформацію і не містити зайвого;
- з'ясувати сумісність з браузером, встановленим на комп'ютері.
- швидкість:
- звернути увагу на швидкість роботи.
- стандарти:
- якісно зроблена CMS зобов'язана підтримувати прийняті стандарти.
- функціональність:
- система повинна володіти всіма функціональними можливостями;
- дізнатися, скільки коштує базовий комплект (ядро + набір модулів)
- редагування тексту:

- повинен бути візуальний редактор, що зовні нагадує популярний текстовий редактор MS Word;
- людино-зрозумілі URL-адреси дуже важливі;
- система повинна підтримувати мультимовність (мовні локалізації).
- сервіс блокування користувачів і антіспамний фільтр;

2. КОНЦЕПТУАЛЬНЕ ТА ФУНКЦІОНАЛЬНЕ ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Архітектура електронного магазину

Архітектура сайту - систематизація інформації та навігації по ній з метою допомогти відвідувачам більш успішно знаходити потрібні їм дані. Добре продумана грамотна архітектура сайту гарантує, що користувачі витратять менше часу на пошук потрібної інформації.

Розробка архітектури сайту повинна вестися з урахуванням найбільш важливої інформації з точки зору просування товарів / послуг на інтернет-ринку. У процесі створення структури нового сайту, або оптимізації структури вже існуючого, необхідно концентрувати увагу споживачів саме на цій інформації і управляти відвідуваністю сайту потенційними клієнтами саме в найбільш важливих розділах сайту відповідно до позиціонуванням на ринку, просуваються товарів / послуг.

Грамотне розподіл пріоритетів між розділами і сторінками сайту, зробить їх основними точками входу на сайт, що дозволить потенційному споживачеві швидко знайти необхідну йому інформацію про товари послуги та підвищити успішність бізнесу в інтернеті[5].

2.2 Проектування інтернет-магазину

Архітектура інтернет - магазину повинна бути проста і інтуїтивно зручна. І складається з Клієнтської частини, Програмної частини і Адміністрування як показано на рисунку 2.1.



Рисунок 2.1 Архітектура інтернет-магазину

Система дозволяє досить зручно розробляти сайти різної складності. Бездоганно слідкує за захистом інформації та структури клієнтських даних. Полегшує пошук необхідної інформації та надає змогу розмежовування ресурсів сайту.

Програмна частина архітектури інтернет - магазину розглядається як взаємозв'язок операційної і серверної частини.

В операційній частині розглядається середовище розробки інтернет магазину.

Серверна частина містить у собі розміщення інтернет магазину на сайті провайдера, що підтримують технології, використовувані при створенні інтернет - магазину.

2.3 Розробка операційної та серверної частин

Інтернет магазин розробляється в середовищі php. Для відповіді обґрунтування вибору було проведено порівняння PHP з іншими мовами програмування Web-додатків. Це його основні конкуренти - Perl, ASP.NET, ColdFusion і Java.

1. PHP і Perl

Мова Perl був створений програмістом Ларрі Уоллом (Larry Wall) і розшифровується як Practical Extraction and Report Language. Перекласти це можна як "мова для практичного вилучення даних і складання звітів". Perl є інтерпретатором і з'явився набагато раніше, ніж Web. З появою інтернету мова Perl виявився зручним інструментом для створення динамічних Web-сторінок. Велике поширення Perl отримав тому, що був доступний на кожному Web-сервер, оскільки практично всі вони працювали під управлінням UNIX, і альтернативою йому був лише більш складну мову C.

Perl програє PHP в витонченість, притому, що PHP пропонує функціональність, аналогічну Perl при значно меншій надлишковості та порівняльній простоті синтаксису. Надмірність мови Perl пов'язана з тим, що він розроблявся для найрізноманітніших додатків, що не могло не позначитися на його синтаксисі, в той час як PHP спочатку розроблявся для Інтернету. На думку дуже багатьох користувачів, що не знали спочатку ні PHP, ні Perl, вивчити PHP

набагато простіше[6].

2. PHP і Java

Взагалі кажучи, розрізняють мову Java і технологію Java. Мова Java являє собою C-подібну мову, яка розроблялася як "покращений C ++". Технологія Java включає в себе клієнтську і серверну частину, а також доступ до баз даних, тому найбільш коректно проводити порівняння технології Java зі зв'язкою Apache / PHP / MySQL. Технологія Java розроблялася як міжплатформена технологія, що дозволяє створювати Web-додатки корпоративного масштабу. Основними достоїнствами цієї технології є міжплатформенна переносимість та об'єктно-орієнтована мова, що дозволяє створювати складні і об'ємні програми. До числа недоліків можна віднести повільне виконання, споживання великої кількості пам'яті (плата за міжплатформову переносимість) і складність розробки Web-додатків в порівнянні з PHP. Тим часом PHP майже не поступається Java в гнучкості і масштабованості створених додатків. Крім того, використання Java для розробки досить простих додатків навряд чи виправдано, а при роботі над серйозними завданнями Java обходиться дорожче.

3. PHP і ASP.NET

ASP (Active Server Pages) - мова сценаріїв від Microsoft. Якщо мова ASP значно поступався PHP по багатьом параметрам, в першу чергу, за часом виконання сценаріїв, то з появою технології ASP.NET ситуація змінилася.

Основна перевага ASP.NET полягає в можливості застосування всієї потужності інтерфейсу Windows для розробки Web-додатків. Це дозволяє розробляти сторінки зі складним інтерфейсом, не піклуючись про збереження, шифруванні та передачі змінних, використовуваних на динамічних сторінках.

ASP.NET тісно інтегрований в нову технологію NET компанії Microsoft. Головною особливістю розробки Web-додатків за допомогою ASP.NET є можливість використовувати будь-яку мову програмування середовища виконання NET. Це може бути як модифікований варіант Visual Basic - Visual Basic.NET, так і нову мову C # або клон Java - J #. Більш того, в рамках одного

Web-додатки можуть бути використані кілька мов програмування. Все це дозволяє об'єднувати в рамках одного проекту програмістів, що володіють різними мовами програмування. Код Web-додатки на ASP.NET компілюється в проміжний мову MSIL (Microsoft Intermediate Language), іноді просто званий IL. Апаратно-незалежний код на мові IL поміщається в кеш Web-сервера для підвищення продуктивності, по аналогії з байт-кодом Java. Але на відміну від технології Java, що забезпечує платформену незалежність, NET забезпечує незалежність від мови розробки. Виконання Web-додатків ASP.NET на серверах, що працюють не під Windows, практично неможливо. Більш того, якщо Web-додаток інтенсивно використовує компоненти NET, перегляд динамічної сторінки буде можливий тільки під Windows і тільки за допомогою браузера Internet Explorer. Тому в сенсі переносимості PHP, безумовно, відрізняється від ASP.NET в кращу сторону. Інший вкрай важливий аргумент на користь PHP полягає у швидкому виправленні виявлених помилок. Теоретично виправити їх можна самостійно, якщо володієте достатньою кваліфікацією. Також можна написати листа розробникам, і з великою часткою ймовірності, у наступній версії помилка буде виправлена. А версії PHP виходять дуже часто. З іншого боку, спробуйте написати про помилку в Microsoft і попросити її виправити. Якщо ви не глава Пентагону, вам це навряд чи вдасться. Ще одним плюсом PHP, особливо важливим для початківців, є те, що його легше використовувати для побудови Web-додатків.

Таким чином, при виборі між ASP.NET і PHP слід виходити з того, яке стоїть завдання. Якщо в мережі Інтранет великої корпорації більшість серверів і клієнтських машин працюють під управлінням Windows, застосування ASP.NET більш ніж виправдане. У той же час для розробки Web-сайту, що розміщується в Інтернеті, доцільніше використовувати PHP.

4. PHP і ColdFusion

Пакет ColdFusion розроблений фірмою Allaire і призначений для швидкої розробки як інтерактивних, так і динамічних Web-документів шляхом обробки

інформації, отриманої з бази даних. Мінусом ColdFusion є досить низька переносимість. PHP працює практично на всіх платформах, а ColdFusion - тільки на чотирьох: Win32, Solaris, HP / UX і Linux. Крім того, ColdFusion, так само як і ASP, є комерційною розробкою. ColdFusion досить мало поширений. У порівнянні з PHP він більш вимогливий до ресурсів. Плюсом ColdFusion є хороша інтегрована середовище розробки і, як наслідок, більш прості, ніж в PHP мовні конструкції і більш швидке написання коду. Правда, через таку високорівневих мов деякі завдання на ньому вирішуються складніше.

5. Для створення інтернет магазину була використана платформа PHP 5.3, а надалі планується перейти на PHP 7.0, так як основні відмінності полягають у реалізації об'єктно-орієнтованої мови поділу. PHP 5.0 працював на двигуні Zend Engine 1.0. До виходу PHP 5.0 фірма Zend повністю переписала двигун і випустила в світ Zend Engine 2.0, котрий, за словами творців, реалізує "нову об'єктну модель". У порівнянні з PHP 5.0, в п'ятій версії мови з'явилися такі особливості:

1. Введено закриті (private) і відкриті (protected) члени і методи класу, а також абстрактні класи і методи.

2. Введено інтерфейси, при цьому клас може реалізовувати довільним ний список інтерфейсів.

3. У PHP 5 також присутні розширені можливості для клонірова ня об'єктів. Призначений для цього конструктор копіювання по зволяєт не просто ідентично клонувати об'єкти, а змінювати при цьому деякі їх властивості.

4. Введено постійні члени і методи класу.

5. Одним з найважливіших нововведень є наявність деструкторів.

6. Доброю новиною є введення в PHP 5.0 статичних членів класу. У PHP 4.0, якщо виникала необхідність у використанні статичних членів, доводилося користуватися штучним прийомом, за ключает в комбінації глобальних змінних зі статичними методами. При цьому виходили члени, які діяли так само, як статичні. Прийом цей, м'яко кажучи, не дуже хороший, оскільки привносив з

собою всі проблеми, пов'язані з використанням глобаль-них змінних. Тепер в PHP 5.0 статичні методи можна оголошувати явно за допомогою ключового слова `static`.

7. У PHP 5.0 запроваджено механізм обробки виключень, відсутній в PHP 5.0. Таким чином, в результаті подібних нововведень в об'єктну модель PHP 5.0 став повністю професійною мовою програмування, що дозволяє використовувати його для вирішення завдань будь-якого ступеня складності.

У серверній частині архітектури розглядається робота інтернет - магазину в мережі Інтернет, взаємодія програмного забезпечення магазину і сервісів, що надаються власниками серверів, провайдерів. Так як після виготовлення інтернет-магазину необхідно буде його розмістити в мережі інтернет і доведеться зіткнутися з вибором місць розміщення, інакше кажучи вибрати хостинг.

Слово "хостинг" походить від англ. "Hosting". (Host - головна машина, господар)

Сервіси, що пропонують свої послуги хостингу, дають можливість розміщувати інформацію, інтернет сайт на своєму сервері. Таким чином, не потрібно встановлювати свій власний сервер в Інтернет, що дозволяє істотно заощадити гроші. Залежно від тарифного плану, пропонується певний обсяг місця на жорстких дисках сервера, e-mail, можливість роботи з CGI, MSQ, і т.д.

Будь-яка людина може без особливих труднощів використовувати як платні, так і безкоштовні послуги. Як відомо, безкоштовний сир буває тільки мишоловці і в будь-якому випадку доведеться платити. У першому (платному) варіанті потрібно розплачуватися грошима, а в другому - якістю зв'язку, обмеженнями на розмір файлів і рекламою, яка буде вивішена в магазині.

Як правило, на платному хостингу - домен буде виглядати так: `hostname.com/yourname`, а на безкоштовному - надається субдомен та адреса буде виглядати наступним чином: `yourname.hostname.com`[7].

2.4 Дерево цілей та функцій

Дерево цілей – це графічне зображення взаємозв'язку і підпорядкованості цілей, що відображає розподіл місії і мети на цілі, під цілі, завдання та окремі дії.

Основна ідея щодо побудови дерева цілей – декомпозиція.

Декомпозиція – це метод розкриття структури системи, при якому за однією ознакою її поділяють на окремі складові. Декомпозиція використовується для побудови дерева цілей, щоб пов'язати генеральну мету зі способами її досягнення, що сформульовані у вигляді завдань окремим виконавцям[8].

Основне правило побудови дерева цілей – це повнота редукції – процес зведення складного явища, процесу або системи до більш простих складових.

Для реалізації цього правила використовують такий системний підхід:

- ціль вищого рівня є орієнтиром, основою для розробки (декомпозиції) цілей нижчого рівня;
- цілі нижчого рівня є способами досягнення мети вищого рівня і мають бути представлені так, щоб їхня сукупність зумовлювала досягнення початкової цілі.

Спочатку розглянемо дерево цілей, яких необхідно досягти, щоб розробити підсистему інтернет-магазину, яка відповідає поставленим вимогам (рис. 2.1).

В результаті аналізу предметної області було визначено цілі системи. Дерево цілей зображено на рисунку 2.1.

- Побудова структурованого інтернет-магазину на основі клієнтських потреб, моделей, ресурсів та технічного завдання
- Проектування архітектури – процес, який визначає усі тонкощі гнучкості та безпеки веб-додатку.
- Забезпечення виділеного сервера отримується шляхом забезпечення стартового грошового капіталу та середовищем розробки сайту.
- Забезпечення клієнтського доступу відокремлює користувачів і дає змогу отримати ту, чи іншу інформацію згідно з правом доступу до неї.

- Технічне завдання – формується за допомогою побажання клієнта та макетів, які обговорюються в процесі створення сайту.

Дослідивши предметну область було запроєктовано дерево функцій системи (рис. 2.2).

- Інтернет-магазин – надає методології з організаційних питань функціонування сайту.
- Продаж товару залежить від наявності товару в інтернет-магазині та менеджменту, який його супроводжує.
- Презентація товару – функція залежності показу товару від його характеристик.

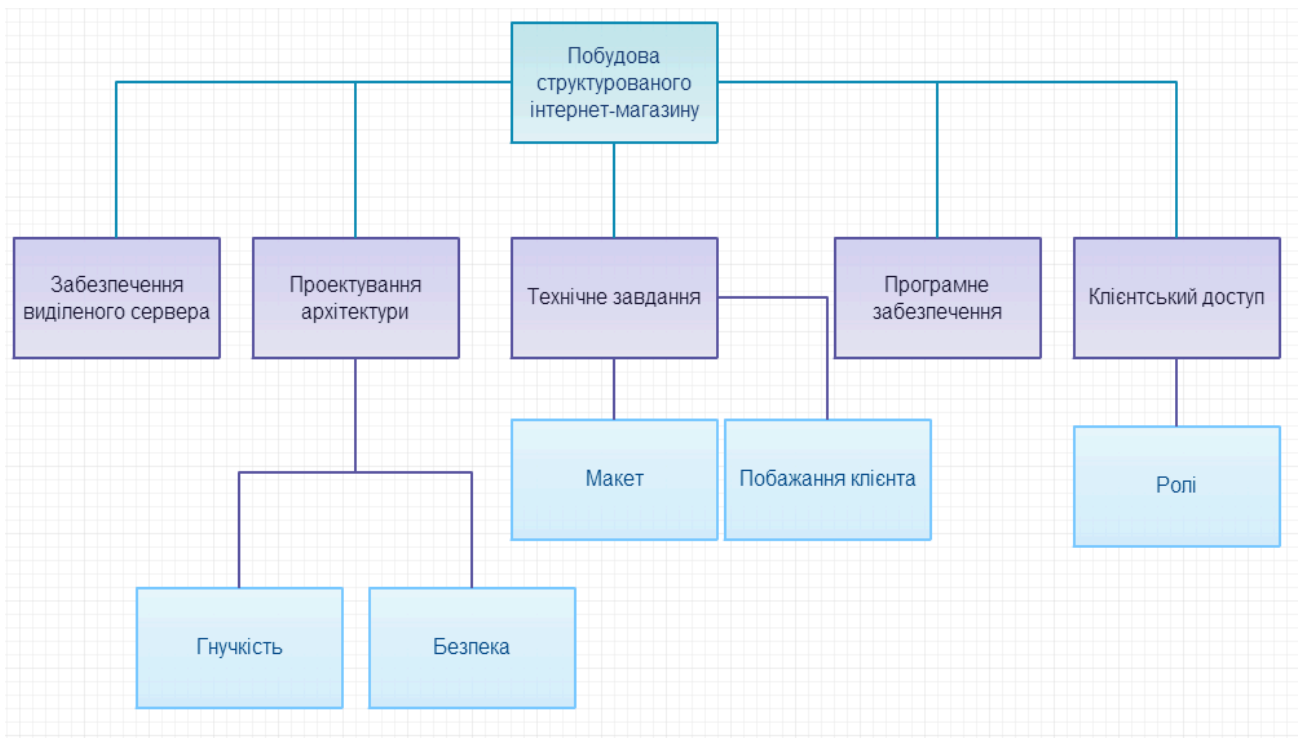


Рисунок 2.2 Дерево цілей

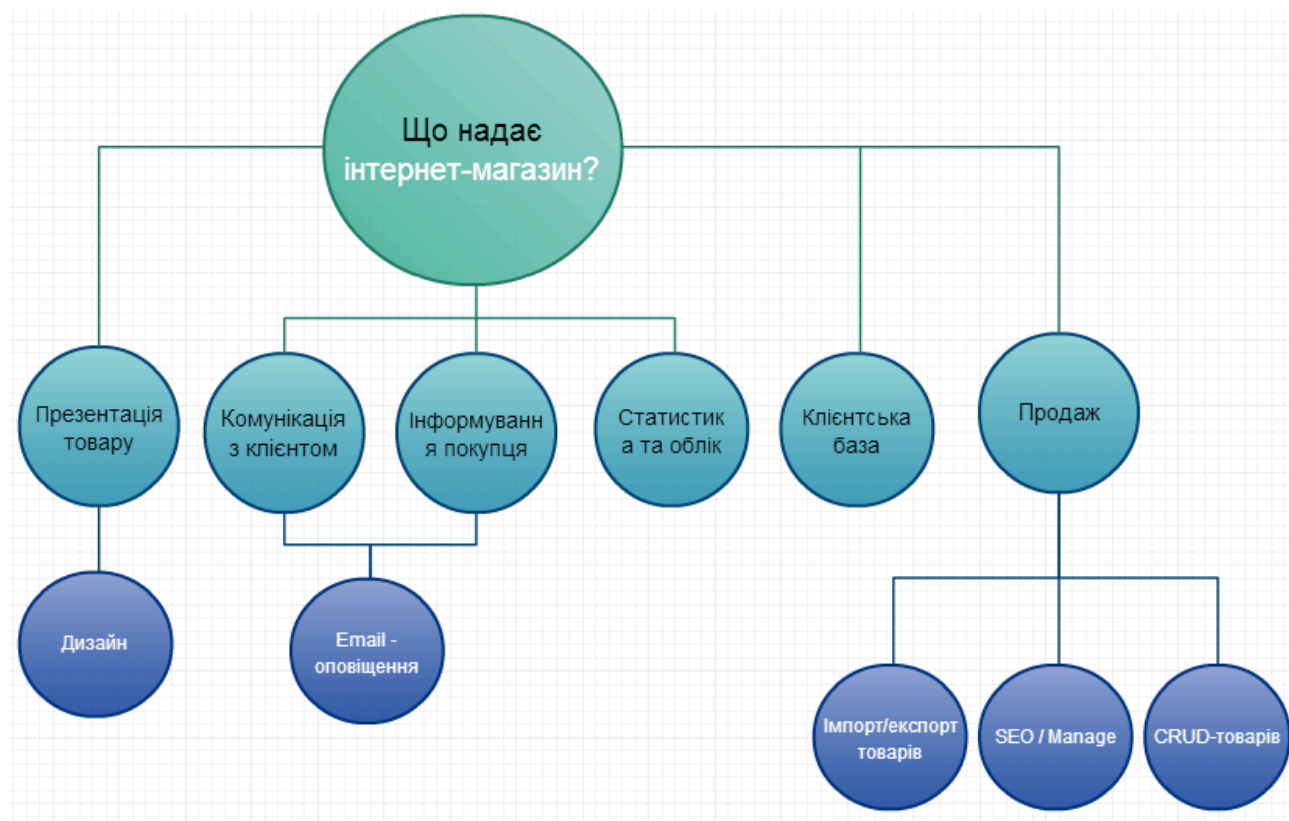


Рисунок 2.3 Дерево функцій

Функціонально CMS складається з 2 -х частин:

- Адміністративна частина
- Фронт-енд

Фронт-енд, в свою чергу, відповідає за формування потоку HTML для перегляду його браузером кінцевого користувача.

Адміністративна частина CMS складається з 2-х незалежних функціональних блоків:

- редактор контенту
- редактор структури

А також тут присутній блок налаштувань для зв'язку CMS з іншими системами.

3 . ПРОЕКТУВАННЯ БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Визначення сутностей БД системи

Розглядаючи концепт нашої системи можна розписати, які там будуть присутні сутності та які в них будуть основні атрибути. Все оформлено у вигляді таблиці основних сутностей (табл. 3.1).

Таблиця 3.1

Сутності та їх атрибути

Назва сутності	Атрибути
Товар	Назва товару
	Урл
	Ціна
	Категорія
	Тип товару
	Код
	Пріоритет
	Опис
	Стара ціна
	В наявності
	Міста
	Доставка
	Прапор відображений
	Новий товар
	Зарекомендований
	Розпродаж
	Фільтри
Метадані	

Продовження таблиці 3.1

	Валюта
	Фотографія
Сайт	Назва
	Домен
	За замовчуванням
Контент	Назва сторінки
	Урл
	Контент
	Прапор відображення
	Метадані
Замовлення	Ім'я клієнта
	Email
	Телефон
	Адреса
	Місто
	Коментар
	Дата замовлення
	Інформація про товар
	Історія замовлення
	Ціна
	Корзина
Дата додавання	
Кількість товарів	
Категорія	Назва
	Прапор відображення дочірніх категорій
	Дочірні категорії
	Метадані

Продовження таблиці 3.1

	Url
	Прапор відображення дочірніх товарів
	Опис категорії
	Фотографія
	Прапор відображення
Колекція	Назва
	Список товарів в колекції

3.2 Представлення даних у вигляді агрегатів

На відміну від реляційної моделі, яка зберігає логічну бізнес-сутність програми в різні фізичні таблиці з метою нормалізації, NoSQL сховища оперують з цими сутностями як з цілісними об'єктами:

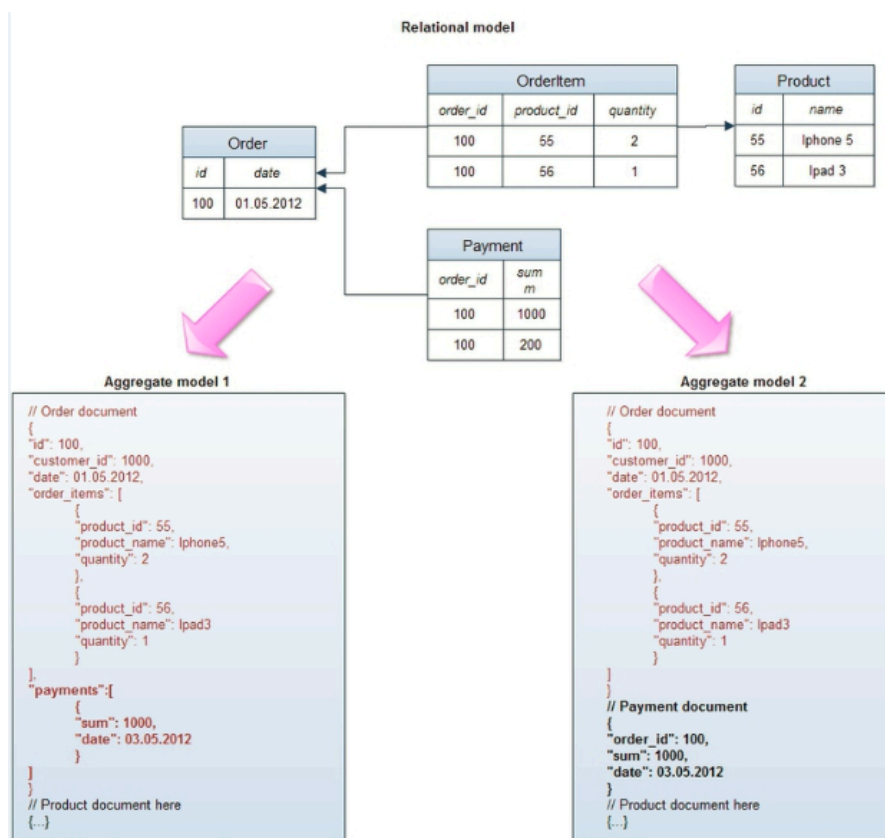


Рисунок 3.1 Агрегати для стандартної концептуальної реляційної моделі електронної комерції

На рис. 3.1 продемонстровані агрегати для стандартної концептуальної

реляційної моделі e-commerce "замовлення - позиції замовлення - платежі - продукт". В обох випадках замовлення об'єднується з позиціями в один логічний об'єкт, при цьому кожна позиція зберігає в собі посилання на продукт і деякі його атрибути, наприклад, назва (така денормалізація необхідна, щоб не запитувати об'єкт продукту при витяганні замовлення - головне правило розподілених систем - мінімум "Джоін" між об'єктами). В одному агрегаті платежі об'єднані із замовленням і є складовою частиною об'єкта, в іншому - винесені в окремий об'єкт. Цим демонструється головне правило проектування структури даних в NoSQL базах - вона повинна підкорятися вимогам програми та бути максимально оптимізованою під найбільш часті запити. Якщо платежі регулярно витягуються разом із замовленням - має сенс їх включати в загальний об'єкт, якщо ж багато запитів працюють тільки з платежами - значить, краще їх винести в окрему сутність.

Багато хто заперечить всі переваги, зауваживши, що робота з великими, часто денормалізованими, об'єктами чревата численними проблемами при спробах дістатися до даних, коли запити не вкладаються в структуру агрегатів. Що ж, якщо ми використовуємо замовлення разом з позиціями і платежами за замовленням (так працює додаток), але бізнес просить нас порахувати, скільки одиниць певного продукту було продано в минулому місяці? У цьому випадку замість сканування таблиці OrderItem (у разі реляційної моделі) нам доведеться витягувати замовлення цілком в NoSQL сховище, хоча більша частина цієї інформації нам буде не потрібна. На жаль, це компроміс, на який доводиться йти в розподіленій системі: ми не можемо проводити нормалізацію даних як у звичайних односерверних систем, оскільки це створить необхідність об'єднання даних з різних вузлів і може привести до значного уповільнення роботи бази[9].

Плюси і мінуси обох підходів я спробував згрупувати в (табл 3.2):

Таблиця 3.2

Плюси і мінуси РСУБД та NoSql

Нормалізація даних	Дані у вигляді агрегатів
<ul style="list-style-type: none">Цілісність інформації при оновлені	<ul style="list-style-type: none">Оптимізація тільки під певний тип запитівПроблематичність при оновлені денормалізованість даних
<ul style="list-style-type: none">Неефективна в розподіленій середі	<ul style="list-style-type: none">Кращий спосіб досягнути максимальної швидкості зчитування в розподіленій середі

3.3 Слабкі ACID властивості

Довгий час консистентність(consistency) даних була "священною коровою" для архітекторів і розробників. Всі реляційні бази забезпечували той чи інший рівень ізоляції - або за рахунок блокувань при зміні та блокуючого читання, або за рахунок undo-логів. З приходом величезних масивів інформації і розподілених систем стало ясно, що забезпечити для них транзакційність набору операцій з одного боку і отримати високу доступність і швидкий час відгуку з іншого - неможливо. Більше того, навіть оновлення одного запису не гарантує, що будь-який інший користувач моментально побачить зміни в системі, адже зміна може відбутися, наприклад, у майстер-ноді, а репліка асинхронно скопіюється на слейв-ноду, з якою і працює інший користувач. У такому випадку він побачить результат через якийсь проміжок часу. Це називається eventual consistency і це те, на що йдуть зараз всі найбільші інтернет-компанії світу, включаючи Facebook і Amazon.

Останні з гордістю заявляють, що максимальний інтервал, протягом якого користувач може бачити неконсистентні дані становлять не більше секунди. Приклад такої ситуації показаний на малюнку 3.2

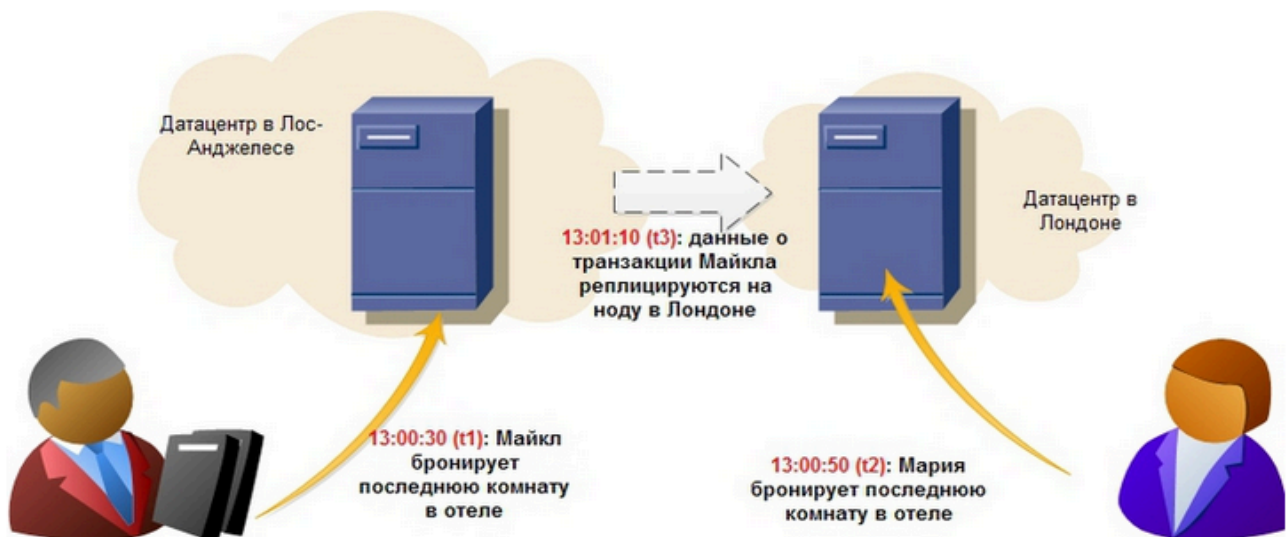


Рисунок 3.2 Eventual consistency

Логічне запитання, яке з'являється в такій ситуації - а що робити системам, які класично висувають високі вимоги до атомарності-консистентності операцій і в той же час потребують швидких розподілених кластерах - фінансовим, інтернет-магазинах і т.д? Практика показує, що ці вимоги вже давно неактуальні: ось що сказав один розробник фінансової банківської системи: "Якби ми дійсно чекали завершення кожної транзакції у світовій мережі АТМ (банкоматів), транзакції займали б стільки часу, що клієнти тікали б геть в люті. Що відбувається, якщо ти і твій партнер знімаєте гроші одночасно і перевищуєте ліміт? - Ви обоє отримаєте гроші, а ми поправимо це пізніше. "Інший приклад - бронювання готелів, показаний на картинці. Онлайн-магазини, чия політика роботи з даними припускає eventual consistency, зобов'язані передбачити заходи на випадок таких ситуацій (автоматичне рішення конфліктів, відкат операції, оновлення з іншими даними). На практиці готелі завжди намагаються тримати "пул" вільних номерів на непередбачений випадок і це може стати вирішенням спірної ситуації.

Насправді слабкі ACID властивості не означають, що їх немає взагалі. У більшості випадків додаток, що працює з реляційною базою даних, використовує транзакцію для зміни логічно пов'язаних об'єктів (замовлення - позиції замовлення), що необхідно, так як це різні таблиці. При правильному

проектуванні моделі даних в NoSQL базі (агрегат представляє з себе замовлення разом з переліком пунктів замовлення) можна домогтися такого ж самого рівня ізоляції при зміні одного запису, що і в реляційній базі даних[10,11].

3.4 Розподілені системи, без спільно використовуваних ресурсів

Знову ж таки, це не стосується баз даних, чия структура за визначенням погано розноситься по віддаленим Нодам.

Це, можливо, головний лейтмотив розвитку NoSQL баз. З лавиноподібним зростанням інформації у світі і необхідності її обробляти за розумний час постала проблема вертикальної масштабованості - зростання швидкості процесора зупинився на 3.5 ГГц, швидкість читання з диска також зростає тихими темпами, плюс ціна потужного серверу завжди більше сумарної ціни декількох простих серверів. У цій ситуації звичайні реляційні бази не здатні вирішити проблему швидкості, масштабованості і пропускну здатності. Єдиний вихід із ситуації - горизонтальне масштабування, коли декілька незалежних серверів з'єднуються швидкою мережею і кожен володіє / обробляє лише частину даних та / або тільки частина запитів на читання-оновлення. У такій архітектурі для підвищення потужності сховища (ємності, часу відгуку, пропускну спроможності) необхідно лише додати новий сервер в кластер. Процедурами шардінга, реплікації, забезпеченням відмовостійкості (результат буде отриманий навіть якщо один або кілька серверів перестали відповідати), перерозподілу даних у разі додавання ноди займається сама NoSQL база.

Коротенько представлю основні властивості розподілених NoSQL баз:

Реплікація - копіювання даних на інші вузли при оновленні. Дозволяє як домогтися більшої масштабованості, так і підвищити доступність і збереження даних. Прийнято поділяти на два види:

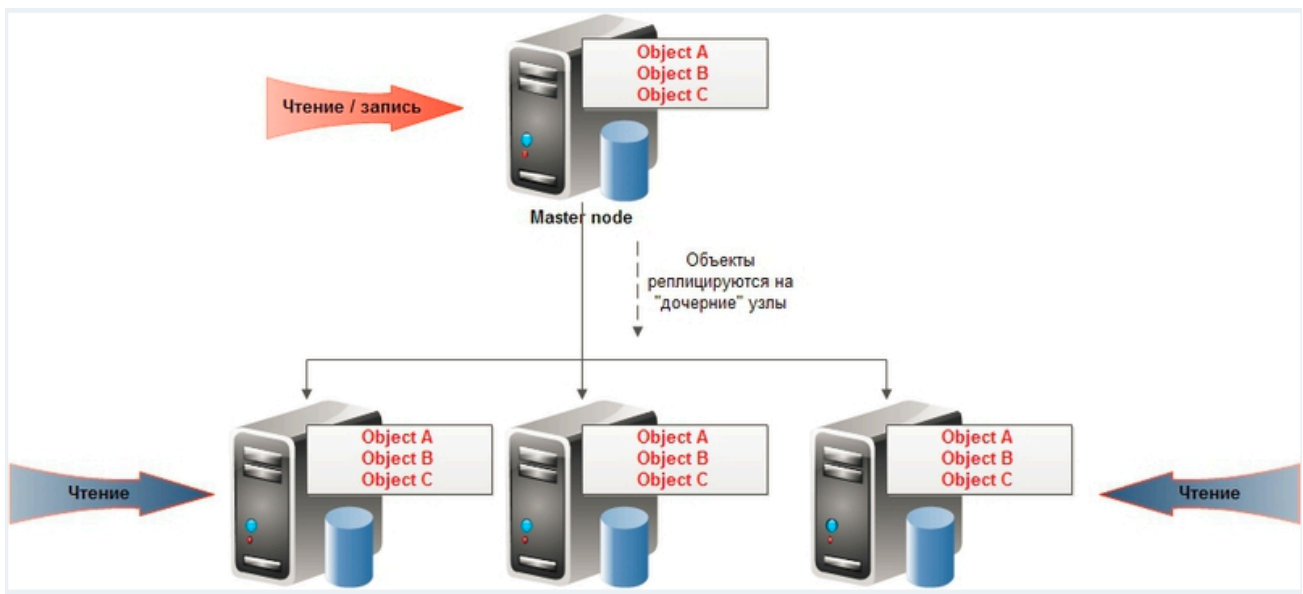


Рисунок 3.3 Реплікація виду master-slave

Та peer-to-peer:

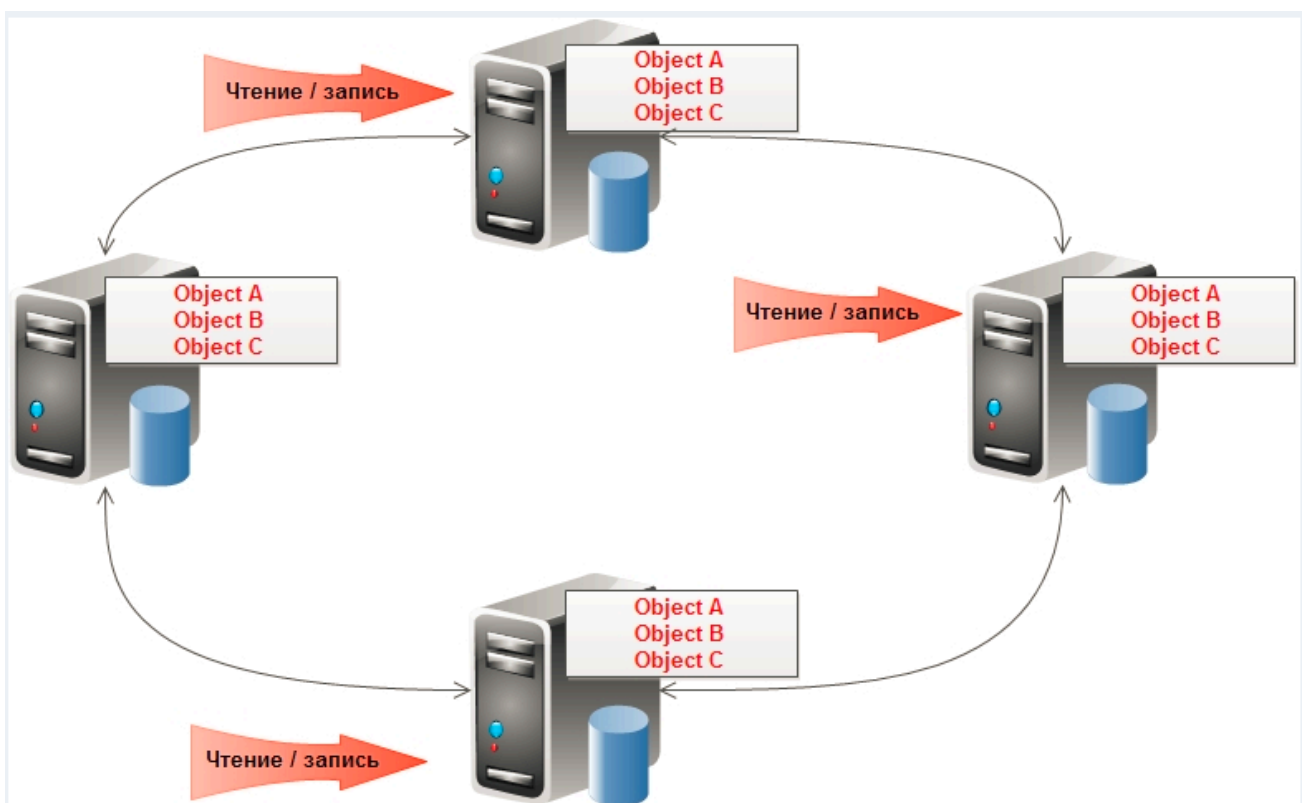


Рисунок 3.4 Реплікація виду peer-to-peer

Перший тип передбачає хорошу масштабованість на читання (може відбуватися з будь-якого вузла), але немасштабований запис (тільки в майстер вузол). Також є тонкощі із забезпеченням постійної доступності (у разі падіння

майстра або вручну, або автоматично на його місце призначається один з решти вузлів).

Розглядаючи існуючі методи взаємодії усіх концепцій розвитку масштабування можна розробити певний шаблон.

В галузі телекомунікацій і програмного забезпечення, масштабовність є бажаною властивістю системи, мережі, або процесу, яка свідчить про здатність системи обробити більший обсяг роботи або бути легко розширеною.

Шардінг - поділ даних по вузлах.



Рисунок 3.5 Шардінг даних

Шардінг часто використовувався як "милиця" до реляційних баз даних з метою збільшення швидкості та пропускної здатності: користувальницький додаток партіціровав дані на декілька незалежних баз даних і при запиті відповідних даних користувачем зверталвся до конкретної бази. У NoSQL базах даних шардінг, як і реплікація, проводяться автоматично самою базою і користувальницький додаток відокремлено від цих складних механізмів.

3.5 Фізичне проектування інформаційного забезпечення

Фізичне проектування – це процес створення опису реалізації інформаційного забезпечення на вторинних запам'ятовуючих пристроях із указівкою структур збереження і методів доступу, використовуваних для організації ефективної обробки даних. Фізичний етап проектування забезпечує вибір раціональної структури збереження даних і методів доступу до них, виходячи з арсеналу методів і засобів, що надаються розроблювачу конкретної СУБД[20].

Основною метою фізичного проектування є опис способу фізичної реалізації логічного проекту.

Таблиця 3.3

Фізична структура колекції Товар

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
id	_id	Об'єкт BSON	Первинний ключ
Назва товару	Title	текст	
Урл	url	текст	Вторинний ключ
Ціна	price	дійсне	
Категорія	categories	Масив Об'єктів BSON	Масив Вторинних ключів
Тип товару	type_id	Об'єкт BSON	Вторинний ключ
Код	product_code	текст	
Пріоритет	Priority	текст	
Опис	Description	текст	
Стара ціна	Old_price	Дійсне	
В наявності	In_stock	ціле	

Продовження
таблиці 3.3

Міста	Cites	текст	
Доставка	Shipping	текстовий масив	
Прапор відображений	Enabled	булевий	
Новий товар	new_item	булевий	
Зарекомендований	recommended	булевий	
Розпродаж	Sale	булевий	
Фільтри	Product_filters	текстовий масив	
Метадані	meta	текстовий масив	
Валюта	currency	текст	
Фотографія	photos	текстовий масив	
Ідентифікатор сайту	site_id	Об'єкт BSON	Вторинний ключ

Таблиця 3.4

Фізична структура колекції Content

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
id	_id	Об'єкт BSON	Первинний ключ
Назва сторінки	title	ціле	
Урл	name	текст	
Контент	full_content	текст	
Прапор відображений	enabled	булевий	
Метадані	meta	текстовий масив	
Ідентифікатор сайту	site_id	Об'єкт BSON	Вторинний ключ

Фізична структура колекції Comments

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
id	_id	Об'єкт BSON	Первинний ключ
Ідентифікатор сторінки	page_id	текст	
Ідентифікатор користувача	user_id	текст	
Прапор відображений	show	булевий	
Коментар	comment	текст	
Дата додавання	date_add	ціле	
Відповідь	parent_id	текст	
Ідентифікатор сайту	site_id	Об'єкт BSON	
Група коментаря	group	текст	
Ір користувача	ip	текст	

Фізична структура колекції Cart

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
id	_id	Об'єкт BSON	Первинний ключ
Дата додавання	date_add	дійсне	
Список товарів	products	Об'єкт BSON	
- Прапор відображений	product_id	текст	
- Кількість товару	quantity	ціле	
- Ціна товару	price	дійсне	

Фізична структура колекції Orders

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
id	_id	Об'єкт BSON	Первинний ключ
Ідентифікатор користувача	user_id	текст	
Номер замовлення	order_number	ціле	
Email	email	текст	
Ім'я	name	текст	
Телефон	phone	текст	
Місто	city	текст	
Вулиця	address	текст	
Коментар	notice	текст	
Дата додавання	date_add	ціле	
Статус замовлення	status	текст	
cart_info	cart_info	масив	
Ідентифікатор сайту	site_id	Об'єкт BSON	
Всього	order_price	Дійсне	
Історія замовлення	status_history	масив	

Таблиця 3.8

Фізична структура колекції Uploads

Назва поля	Ідентифікатор	Тип	Призначення
1	2	3	4
Id	_id	Об'єкт BSON	Первинний ключ
Каталог завантаження	upload_path	текст	
Каталог знаходження	upload_src	текст	

Оригінальне ім'я	file_original_name	текст	
Фактичне ім'я	file_name	текст	
Дата додавання	date_add	ціле	
Тип	type	текст	
Ідентифікатор завантаження	Key	текст	
Розмір файлу	file_size	дійсне	
Тип файлу	file_type	текст	
Ідентифікатор сайту	site_id	Об'єкт BSON	
Інші файли	files_processed		

База даних може мати нуль або більше «колекцій». Колекція настільки схожа на традиційну «таблицю», що можна сміливо вважати їх одним і тим же.

Колекції складаються з нуля або більше «документів». Знову ж, документ можна розглядати як «рядок». Список колекцій зображений на малюнку 4.3

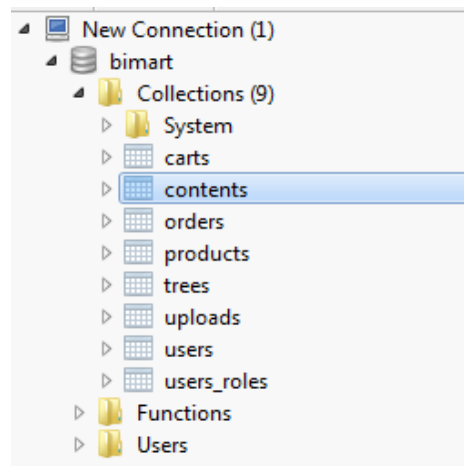


Рисунок 3.6 Список колекцій

Кожен документ містить свій власний список полів.

	_id	type	title	name	show_child_cate	show_child_cate	enabled	meta	description	
1	ObjectId("5...	content-category	Категория ...	kategorija-...	TF	true	null	TF	true	{ 4 fields }
2	ObjectId("5...	products-mass-action								
3	ObjectId("5...	category	Холодильн...	kholodilniki	TF	true	null	TF	true	{ 5 fields }
4	ObjectId("5...	category	Стиральн...	stiralnye-m...	TF	true	null	TF	true	{ 5 fields }
5	ObjectId("5...	category	Электроника	jelektronika	TF	true	null	TF	true	{ 5 fields }
6	ObjectId("5...	slider	Главный с...					TF	true	

Рисунок 3.7 Колекція trees

Колекція trees містить в собі багато різних об'єктів з різною кількістю полів. Основні об'єкти, які містить в собі ця колекція зображені на рисунку 3.8

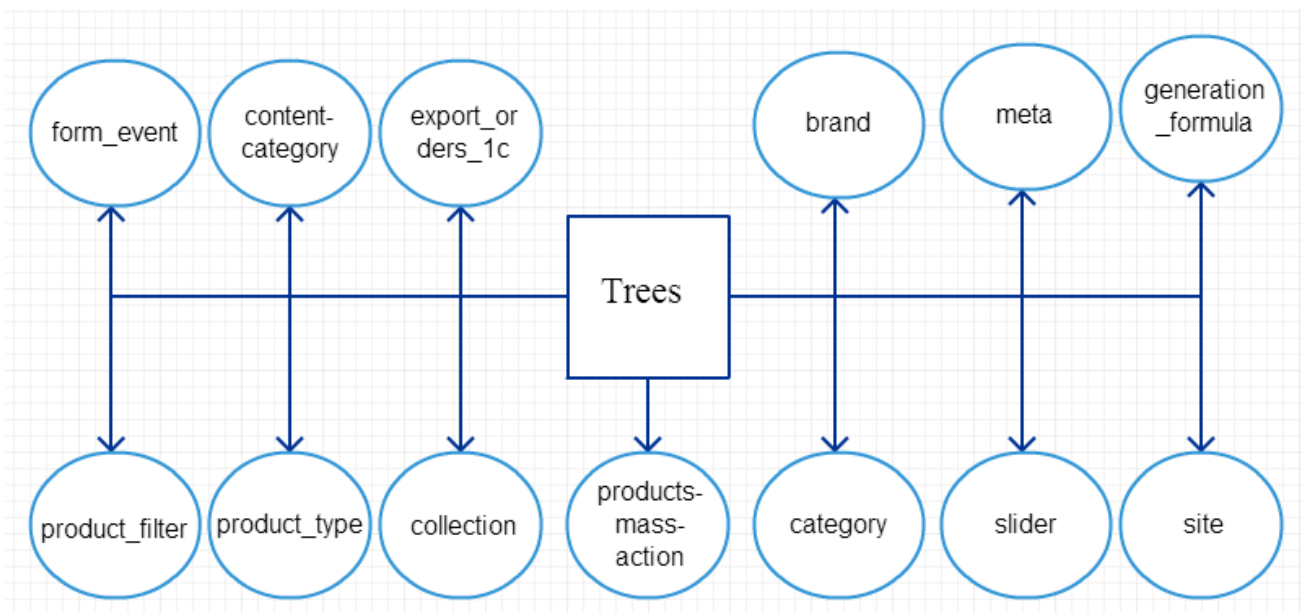


Рисунок 3.8 Основні об'єкти колекції trees

Документ має наступну структуру:

```

{
  "_id" : ObjectId("5550c5d76d8c2d581600002a"),
  "action_type" : "delete",
  "product_ids" : [
    ObjectId("554fcf656d8c2dbc19000029")
  ],
  "date_add" : 1.43136e+009,
  "type" : "products-mass-action",
  "date_start" : 1.43136e+009,
  "title" : "",
  "site_id" : ObjectId("555891426d8c2dbc02000029")
}
  
```

4. АЛГОРИТМИ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

4.1 Декомпозиція інструментальних засобів

Використання високорівневих концепцій для забезпечення більш гнучкої архітектури веб-програми забезпечується не лише завдяки вибору інструментально-програмних засобів, а й якісним підходом до реалізації алгоритмів та проектування структури сgm-системи. Опишемо більш детально інструментарій і почнемо ми з вибору фрейворку, вибір якого впав на zf2.

4.1.1 Zend Framework 2

Zend Framework 2 є фреймворком з відкритим вихідним кодом для розробки веб-програм на PHP 5.3+. Використовує тільки об'єктно-орієнтований код і всі нововведення PHP 5.3, такі як: namespaces (Простір імен), late static binding (Пізнє статичну зв'язування), lambda functions and closures (Анонімні функції і замикання).

Zend Framework 2 еволюціонував від Zend Framework 1 - фреймворку з більш ніж п'ятнадцятьма мільйонами сайтів, які використовують його.

Важливо: ZF2 не сумісний з ZF1 оскільки використовує всі нововведення PHP 5.3 +. Так само були переписані всі компоненти ядра.

Кожен компонент Zend Framework 2 є унікальним і розроблений з мінімальними залежностями від інших компонентів. ZF2 слідує компонентному принципу створення програм. Така слабозв'язана архітектура дозволяє використовувати тільки ті компоненти, які вам необхідні. Фреймворк використовує PEAR і Composer для установки і відстеження залежностей, як для всього проекту в цілому, так і для кожного з компонентів окремо[12].

Хоч кожен компонент Zend Framework 2 може бути використаний окремо, стандартний набір бібліотек робить його дуже потужним та гнучким засобом

розробки веб-програм. Крім того, він пропонує надійну і високопродуктивну реалізацію MVC, абстракцію бази даних, яка проста у використанні. Реалізація HTML5 форм візуалізації, перевірки і фільтрації даних, можна розробити завдяки об'єктно-орієнтованого інтерфейсу. Такі компоненти, як Authentication і Acl, забезпечують аутентифікацію і авторизацію користувача облікових даних.

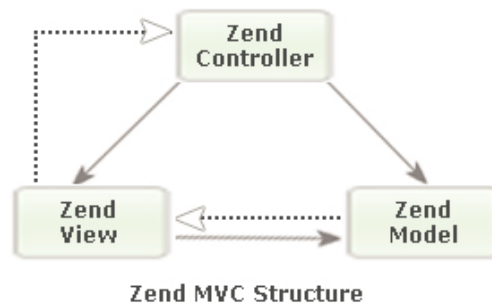


Рисунок 4.1 Структура mvc в Zend framework 2

«*Controller*» - це втілення бізнес-логіки додатка. Контролер керує запитамися користувача. Його основна функція - викликати і координувати дію необхідних ресурсів та об'єктів, потрібних для виконання дій, що задаються користувачем. Зазвичай контролер викликає відповідну модель для задачі і вибирає підходящий вид.

«*Model*» - це набір правил та концепцій, які використовуються для роботи з даними, які представляють управління додатком. У будь-якій веб-програмі структура моделюється як дані, які обробляються певним чином.

«*View*» - забезпечує різні способи подання даних, які отримані з моделі.

Він може бути шаблоном, який заповнюється даними, або чистою інформацією.

Що ж, досить звична картина з класичного MVC. Які ж характерні особливості присутні саме у zf2? Структура запитів зображена на малюнку 4.2.

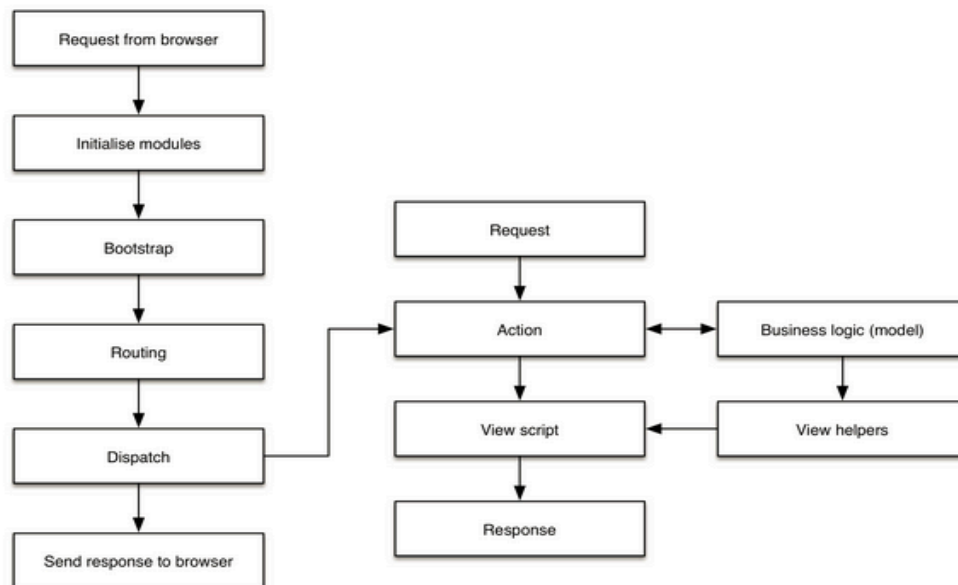


Рисунок 4.2 Загальний алгоритм роботи сайту

Опишемо кожен пункт більш детально:

- Браузер запитує сторінку у сайта.
- Index.php
 - Підключає конфіг та встановлює роути
 - Підключаємо ядро та запускаємо програму
- Bootstrap → run
 - Налаштовуємо завантажувач
 - Налаштовуємо конфігурацію
 - Налаштовуємо View та Layout
 - Ініціалізуємо адаптер бази даних
 - Підключаємо маршрутизацію
 - Налаштовуємо front контроллер
 - Запускаємо програму
- Диспетчеризація

- Controller
 - Оброблюємо дані
 - Отримаємо дані з моделей
 - Визначення змінних
- Model
 - Працюємо за даними (select,insert,delete,update)
- View
 - Підстановка змінних
- Браузер відображення отриманої сторінки

4.1.2 MongoDB

Що в першу чергу слід знати про MongoDB?

MongoDB - це документо-орієнтована СУБД. Дані в MongoDB зберігаються в документах, які об'єднуються в колекції. Кожен документ являє собою JSON-подібну структуру. Провівши аналогію з реляційними СУБД, можна сказати, що колекціям відповідають таблиці, а документам - рядки в таблицях. На відміну від РСУБД MongoDB не потребує опису схеми бази даних - вона може поступово змінюватися в міру розвитку програми, що дуже зручно.

В mongodb підтримуються індекси, у тому числі по масивах і вкладеним документам, а також геопросторові індекси. Також підтримуються унікальні і складені індекси.

В MongoDB є атомарні операції, compare-and-swap, курсори, запис без підтвердження і навіть MapReduce. Розмір колекції в MongoDB може бути обмежений числом документів або мегабайтами. Якщо колекція занадто розростеться, старі документи будуть видалені.

У запитах можуть використовуватися функції на мові JavaScript. У MongoDB підтримується журналювання, а також асинхронна реплікація двох видів - master-slave реплікація та replica sets. Розробники MongoDB

рекомендують використовувати останній тип реплікації. Набір реплік являє собою той же master-slave, але в разі падіння master-а серед реплік автоматично вибирається новий майстер. Після відновлення своєї роботи колишній майстер стає реплікою;

Мабуть, сама визначна особливість MongoDB полягає в тому, що документи можуть бути автоматично сегментовані на декілька наборів реплік. Сегментування проводиться за діапазоном; щоб віднести документ до конкретного діапазону, використовується сегментний ключ (shard key). Дані розподіляються між наборами реплік так, щоб кожен набір містив приблизно однаковий обсяг даних.

Про масштабування

Уявіть собі типовий випадок - є база даних, в яку здійснюється запис і читання даних. У динамічно зростаючих системах, обсяги даних, як правило, швидко збільшуються і рано чи пізно можна зіткнутися з проблемою, коли поточних ресурсів машини буде не вистачати для нормальної роботи.

Для вирішення цієї проблеми застосовують масштабування. Масштабування буває 2-х видів - горизонтальне і вертикальне. Вертикальне масштабування - нарощування потужностей однієї машини - додавання процесора, оперативної пам'яті, жорсткого диска. Горизонтальне масштабування - додавання нових машин до існуючих і розподіл даних між ними. Перший випадок найбільш простий, тому не вимагає додаткових налаштувань програми та будь-якої додаткової конфігурації БД, але його недолік в тому, що потужності однієї машини теоретично рано чи пізно упрутся в глухий кут. Другий випадок складніший в конфігурації та виглядає наступним чином:

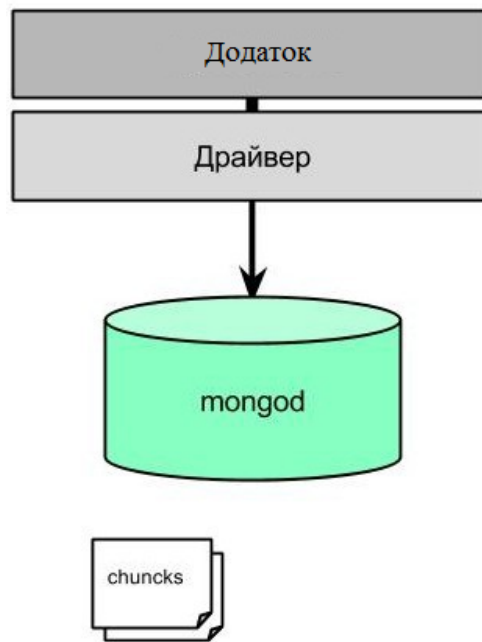


Рисунок 4.3 Проста схема, без шардування

4.1.3 Bootstrap

Bootstrap є найбільш популярним HTML, CSS, та JS фреймворком для зовнішнього вигляду та інтерактивності веб сторінок. Структура bootstrap наведена на рис. 4.6.

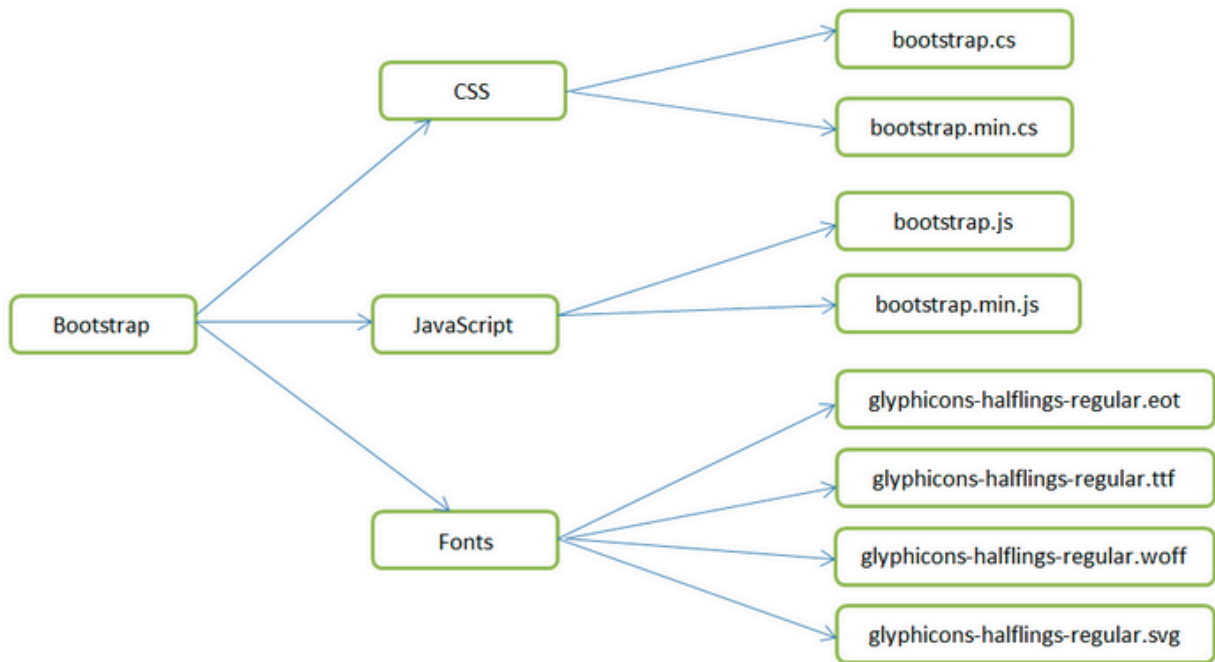


Рисунок 4.6 Структура bootstrap

Html - найпоширеніший і простий спосіб створення Web-інтерфейсу. Всі view appicaliton document складаються з набору HTML-документів, статичних чи генерованих динамічно.

JavaScript - інтерпретована об'єктно-базована мова високого рівня, що дозволяє писати програми, які виконуються в середовищі Web-оглядача. В даний час JavaScript є найбільш широко розповсюдженою мовою в Web для написання сценаріїв, що виконуються на стороні клієнта.

CSS – спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних.

Fonts – набір шрифтів, які підключають через css.

4.1.4 Git. Система контролю версій.

Що таке контроль версій, і навіщо він вам потрібен? Система контролю версій(CVS) - це система, яка реєструє зміни в одному або декількох файлах для того, що б надалі була можливість повернутися до попередньої версії проекту.

СКВ дає можливість повертати окремі файли до колишнього вигляду, повертати до попереднього стану весь проект, переглядати, які зміни відбулися з часом, визначати, хто останнім вносив зміни і чому раптово перестав працювати модуль, хто і коли вніс в код якусь помилку, і багато іншого. Взагалі, якщо, користуючись СКВ, ви все зіпсуєте або втратите файли, все можна буде легко відновити.

Локальні системи контролю версій

Багато хто віддає перевагу контролювати версії, просто копіюючи файли в інший каталог (як правило додаючи поточну дату до назви каталогу). Такий підхід дуже поширений, тому що простий, але він і частіше дає збої. Дуже легко забути, що ти не в тому каталозі, і випадково змінити не той файл, або скопіювати файли не туди, куди хотів, і затерти потрібні файли.

Щоб вирішити цю проблему, програмісти вже давно розробили локальні СКВ з простою базою даних, в якій зберігаються всі зміни потрібних файлів (див. Рисунок 4.67).

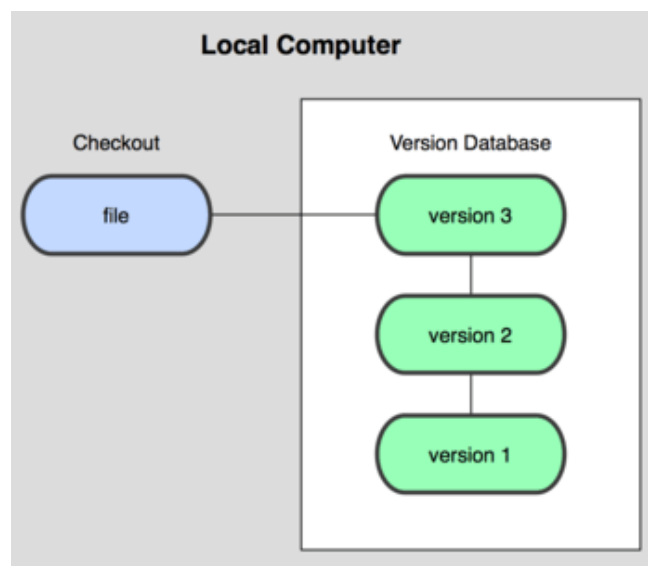


Рисунок 4.7 Схема локальної СКВ

Централізовані системи контролю версій

Наступною основною проблемою виявилася необхідність співпрацювати з розробниками, які працюють за своїми комп'ютерами. Щоб вирішити її, були створені централізовані системи контролю версій (ЦСКВ). У таких системах,

наприклад CVS, Subversion і Perforce, є центральний сервер, на якому зберігаються всі файли під версійним контролем, і ряд клієнтів, які отримують копії файлів з нього. Багато років це було стандартом для систем контролю версій (див. Рис. 4.8).

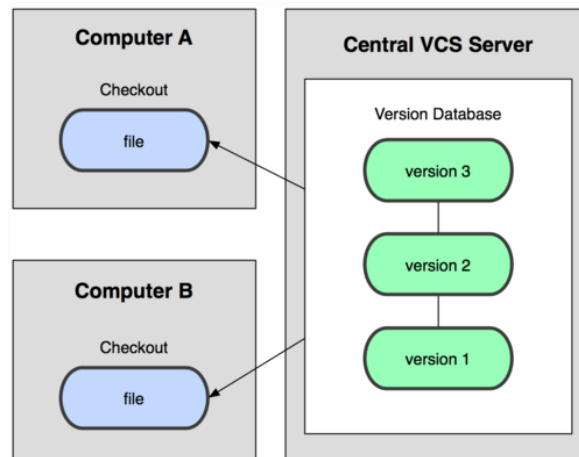


Рисунок 4.8 Схема централізованого контролю версій

Такий підхід має безліч переваг, особливо над локальними СКВ. Наприклад, всі знають, хто і чим займається в проекті. У адміністраторів є чіткий контроль над тим, хто і що може робити, і, звичайно, адмініструвати ЦСКВ набагато легше, ніж локальні бази на кожному клієнті.

Однак при такому підході є й кілька серйозних недоліків. Найбільш очевидний - централізований сервер є вразливим місцем всієї системи. Якщо сервер вимикається на годину, то протягом години розробники не можуть взаємодіяти, і ніхто не може зберегти нову версію своєї роботи. Якщо ж пошкоджується диск з центральною базою даних і немає резервної копії, ви втрачаєте абсолютно все - всю історію проекту, хіба що за винятком кількох робочих версій, збережених на робочих машинах користувачів. Локальні системи контролю версій мають ту ж проблему: якщо вся історія проекту зберігається в одному місці, ви ризикуєте втратити все.

Розподілені системи контролю версій

І в цій ситуації в гру вступають розподілені системи контролю версій (РСКВ). У таких системах як Git, Mercurial, Bazaar або Darcs клієнти не просто

вивантажують останні версії файлів, а повністю копіюють весь репозиторій. Тому у випадку, коли "вмирає" сервер, через який йшла робота, клієнтський репозиторій може бути скопійований назад на сервер, щоб відновити базу даних. Кожен раз, коли клієнт забирає свіжу версію файлів, він створює собі повну копію всіх даних (див. Рисунок 4.9).

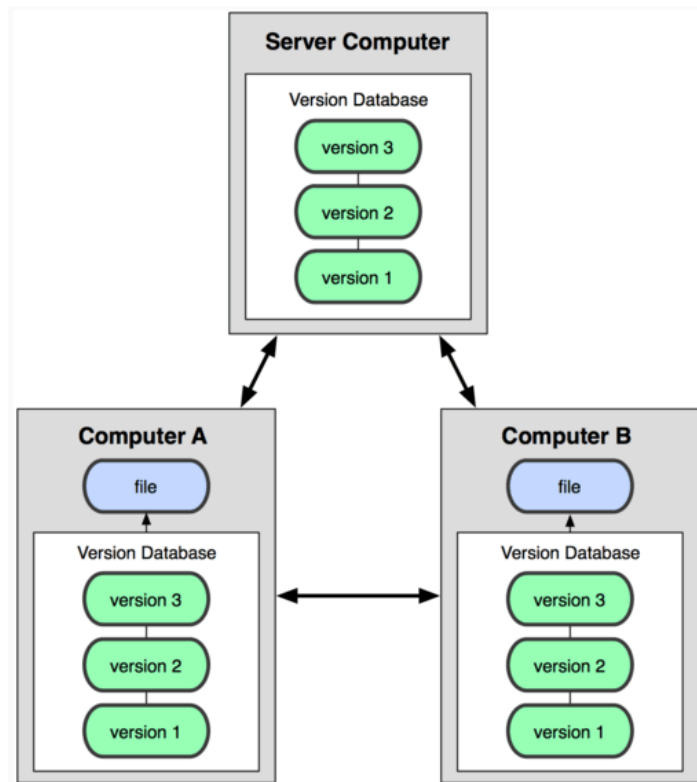


Рисунок 4.9 Схема розподіленої системи контролю версій

4.2 Діаграма основних класів системи та опис модульності

В основі додатків побудованих на Zend Framework 2 використовується модульна система. У кожному модулі можлива початкове налаштування (bootstrapping), конфігурування помилок і роутінга для всього модуля.

Весь наш код знаходиться в окремому модулі «Application», який містить конфігураційні файли(config), список мов, які він підтримує(language), функціональну частину(src), в якій знаходяться контролери (controllers), моделі (models), сервіси(service), плагіни(Plugin) та хелпери(Helpers), та шаблони виду

(views). Розглянемо кожен каталог та опишемо кожен створений клас .

Загальна архітектура системи зображена на рисунку 4.10.

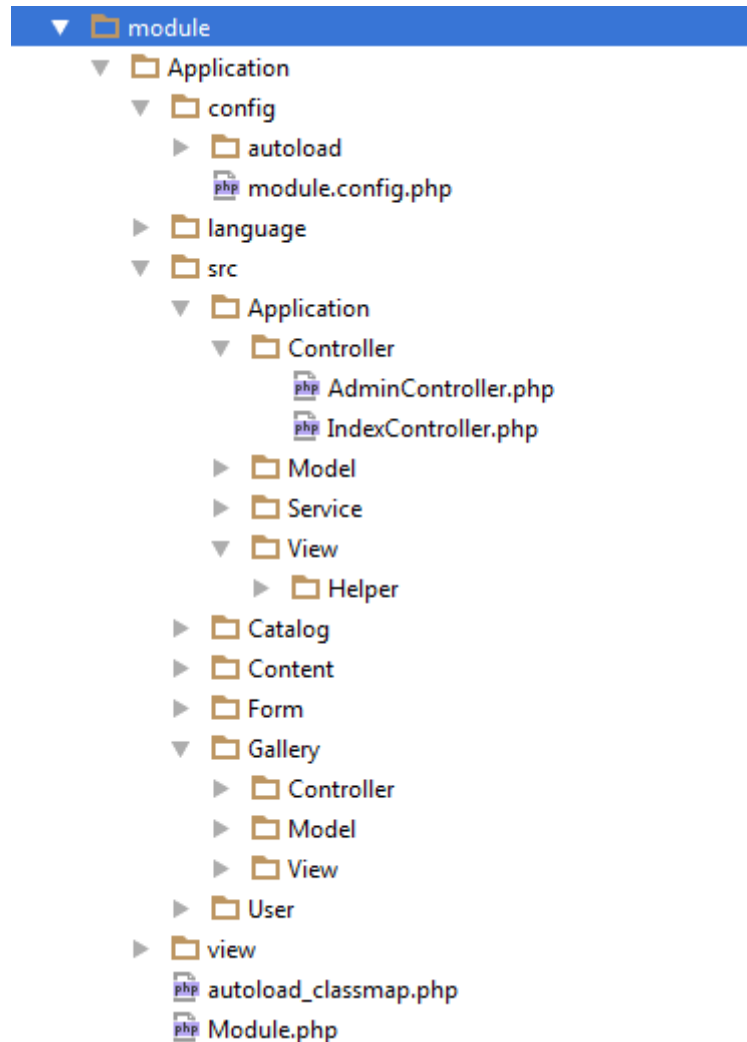


Рисунок 4.10 Функціональна структура модуля Application

4.3 Опис основних алгоритмів

Алгоритм пошуку на сайті:

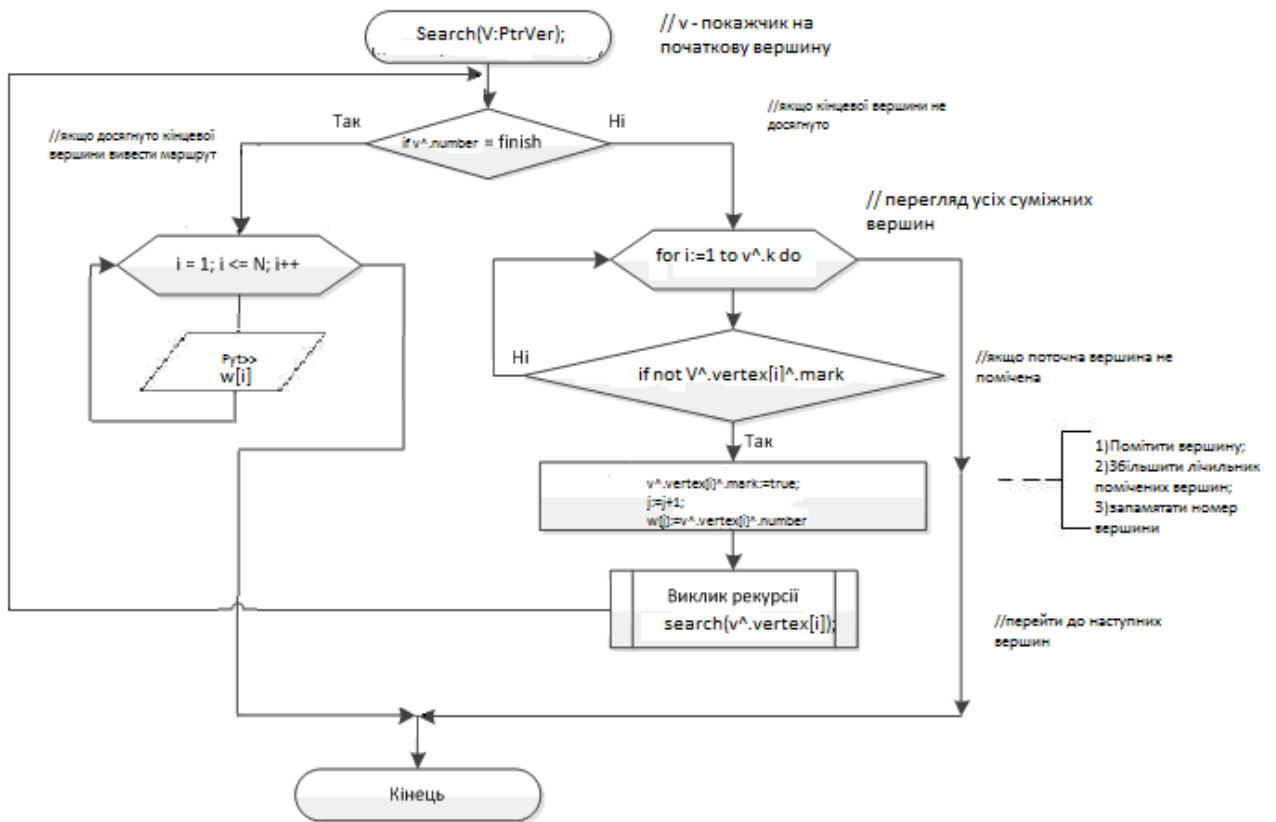


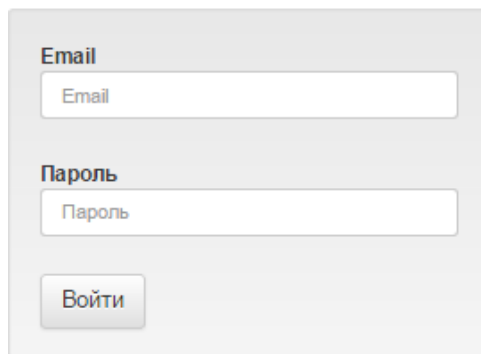
Рисунок 4.15 Опис основних алгоритмів

5 ПРИКЛАД ВИКОРИСТАННЯ СИСТЕМИ

5.1 Введення початкових даних для контрольного прикладу

Тестовий приклад проводиться на локальному комп'ютері(ноутбук). Тож звертаємося по url beemart2.loc/admin.

Потрапляємо на наступну форму аутентифікації :



The image shows a login form with a light gray background. It contains two text input fields. The first field is labeled 'Email' and has the placeholder text 'Email'. The second field is labeled 'Пароль' (Password) and has the placeholder text 'Пароль'. Below these fields is a button labeled 'Войти' (Login).

Рисунок 5.1 Форма аутентифікації

Вводимо Email та Пароль. Тестовий користувач був створений заздалегідь з правами адміністратора для демонстрації усього функціоналу.

Після аутентифікації потрапляємо на наступну сторінку:



The image shows a web interface with a dark header. On the left, there are navigation links: 'Магазин', 'Материалы', 'Галереи', and 'Формы'. On the right, it shows 'beemart2.loc' and a settings icon. Below the header, the text 'Последние зарегистрированные пользователи' is displayed above a white box containing the email address 'test@beemart.loc'.

Рисунок 5.2 Головна сторінка адмінки

Наведемо опис та перелік усього функціоналу:

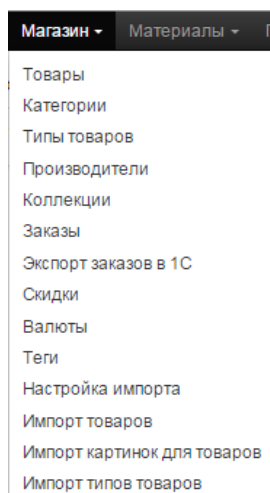


Рисунок 5.3 Список функционалу интернет–магазина

5.1.1 Магазин

«Магазин» – містить перелік всього основного функціоналу по керуванню інтернет–магазином.

«Товары» – містить перелік всіх товарів конкретного домену.

 A screenshot of an admin interface showing a table of products. At the top, there is a breadcrumb 'Админ. центр / Товары' and a '+ добавить' button. Below is a search bar with 'Точный поиск' and 'Опубликован:' dropdown, and a price filter 'Цена: [] - []' with a 'фильтровать' button. The table has columns: 'Наименование', 'Категория', 'Цена', and 'Опубликован'. It lists five products with their categories, prices, and status icons. At the bottom right, there are links for 'параметры таблицы' and 'скачать таблицу'.

Наименование	Категория	Цена	Опубликован
Однокамерный холодильник ATLANT MX 5810-72	Бытовая техника -> Холодильники	9890	
Dell Latitude 3540 (CA002L35401EM)	Электроника -> Ноутбуки	14000 9333	
Ноутбук Acer Aspire E5-511G-P2VL	Электроника -> Ноутбуки	7888 40228	
Мягкая игрушка Vambi Винни Пух 60 см (MP 0399 F)	Детские товары	665	
Однокамерный холодильник CANDY CCTOS 502WH	Бытовая техника -> Холодильники	3800 4200	

Рисунок 5.4 Карточка товару

Опишем функционал в розділі *Таблицы* в розділі «Товары».

«*Параметры таблицы*» – визначає, які колонки будуть відображатися в таблиці.

«*Скачать таблицу*» – зберігає поточні данні таблиці в ексель файл та завантажує його на комп'ютер користувача.

«Параметры фильтра» – визначає, які поля будуть відображатися для фільтрації товарів.

The image shows two side-by-side configuration panels. The left panel, titled 'параметры таблицы', has a gear icon and a download icon. It contains a list of 15 items with checkboxes: 'Редактирование' (checked), 'Приоритет' (unchecked), 'Наименование' (checked), 'Изображение' (unchecked), 'Категория' (checked), 'Цена' (checked), 'Опубликован' (checked), 'Рекомендуемый' (unchecked), 'Новинка' (unchecked), 'Распродажа' (unchecked), 'Артикул' (unchecked), 'Наличие' (unchecked), 'Краткое описание' (unchecked), and 'Полное описание' (unchecked). A blue 'сохранить' button is at the bottom. The right panel, titled 'параметры фильтра', has a gear icon. It contains a list of 8 items with checkboxes: 'Поиск' (checked), 'Точный поиск' (checked), 'Опубликован' (checked), 'Товары без фото' (unchecked), 'Категория' (unchecked), 'Цена' (checked), 'Наличие' (unchecked), 'Описание' (unchecked), 'Распродажа' (unchecked), and 'Фильтры товаров' (unchecked). A blue 'сохранить' button is at the bottom.

Рисунок 5.5 Параметры таблицы та фільтра

«Поиск» – текстове поле, яке виконує пошук товарів. Для пошуку спочатку необхідно створити індекс в mongodb для того, щоб відібрати ті поля, по яким буде проходити пошук.

«Точный поиск» – буде шукати тільки точне входження слова.

«Опубликован» – публікація товару.

«Цена» – пошук по ціновому діапазону.

The image shows a search and filter panel. It includes a text input field for 'Поиск:', a checkbox for 'Точный поиск' (unchecked), a dropdown menu for 'Опубликован:' with a downward arrow, and two input fields for 'Цена:' separated by a hyphen. A blue 'фильтровать' button is on the right.

Рисунок 5.6 Панель фільтрації

5.1.2 Функціонал групових дій

Ми маємо змогу вибрати певні товари на сторінці і відредагувати їх, видалити, перерахувати вартість кожного вибраного товару та оновити параметри всіх вибраних нами товарів.

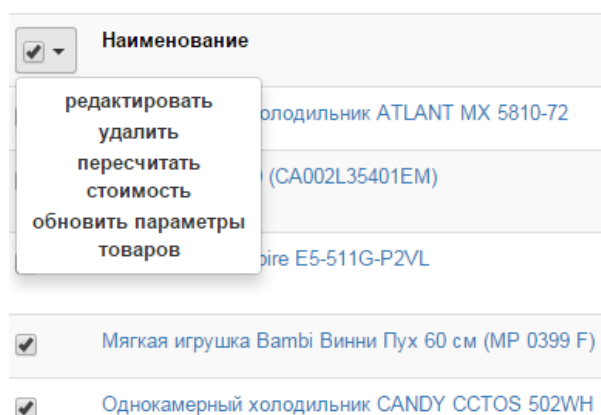


Рисунок 5.7 Функціонал групових дій

При масовому редагуванні товарів ми потрапляємо на наступну форму:

The image shows a web form for mass editing products. It includes sections for 'Укажите поля для редактирования:' with a list of fields, 'Тип товара:' with a dropdown menu, a 'Переопределить влияние на цену' button, 'Редактируемые товары:' with a list of selected items, 'Категория:' with a dropdown menu, and 'Фотографии:' with a file selection area and a 'загрузить' button. There is also a checkbox for 'Удалить ранее загруженные фото'.

Рисунок 5.8 Масове реагування товарів

Ми можемо вказати поля, які будуть оновлені при редагуванні та вибрати ті товари, які ми хочемо відредагувати.

Якщо ми хочемо відредагувати конкретний товар, то ім'я товару на головному вікні – це лінк на вікно редагування товару.

5.1.3 Карточка товару

Карточка товару виглядає наступним чином:

Админ. центр / Товары / Редактирование 🔍 Посмотреть товар на сайте ✔️ сохранить 📄 применить ← назад

Основное | Параметры публикации | SEO | Отчет импорта

Наименование:
Однокамерный холодильник ATLANT MX 5810-72

Артикул:

Приоритет:

Тип товара:
Однокамерный ▾

[Переопределить влияние на цену](#)

Категория:
Холодильники ▾

Цена:
9890

Валюта:
По умолчанию ▾

Старая цена:

Наценка (формат: +10, +2.5%, -10, -2.5%):

Наличие:
16 ИЛИ Выберите статус ▾

Краткое описание: (включить редактор)

Описание: (включить редактор)

Доступные способы доставки товара:
Курьерская
По области ▾

Фотографии:
atlant.png 🗑️ ✖️

Документы:

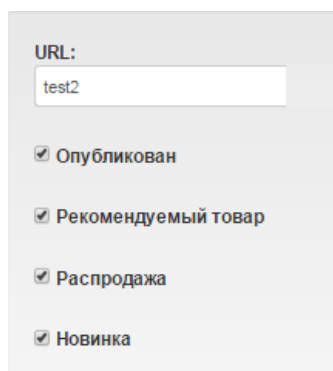
Расположение морозильной камеры:
нижнее
otsustsvuet
sboku ▾

Рисунок 5.9 Форма додавання/редагування карточки товару

Форма додавання та редагування товару мають одні і ті ж самі поля. Після збереження, маємо нагоду переглянути товар на сайті. Товар має чотири вкладки:

«Основное» – базові поля товару. Ми можемо вказати для нього ім'я, артикул, пріоритет відображення, тип до якого належить товар, категорія, ціна, валюту в якій буде відображатися ціна цього товару, стару ціну товару, його скидку, наявність товару, перев'ю, повний опис товару, спосіб доставки. Ми можемо додати технічну документацію до товару, додати фотографії до товару та вибрати фільтри, які належать до цього товару. Фільтри визначаються за допомогою типу товару. Перев'ю та опис товару можна редагувати з html–тегами.

«Параметры публикации» – урл товару, та його вид.



URL:
test2

Опубликован

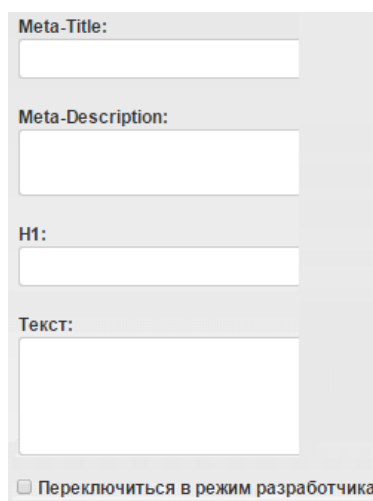
Рекомендуемый товар

Распродажа

Новинка

Рисунок 5.10 Параметри публікації

«SEO» – поля для seo–даних, які можна налаштувати для кожної карточки окремо. Режим розробника описан нижче у розділі про генерацію метаданих.



Meta-Title:
[input field]

Meta-Description:
[input field]

H1:
[input field]

Текст:
[input field]

Переключиться в режим разработчика

Рисунок 5.11 SEO-форма

«Отчет импорта» – звіт, який генерується після імпорту товару.

5.1.4 «Категории»

Категорії містить перелік всіх категорій конкретного домену.



Рисунок 5.12 Лістинг категорій

html5 дає змогу «перетаскувати» категорії, таким чином сортувати їх відображення на сайті.

The image shows a web form for adding or editing a category. The form is organized into several sections:

- Наименование:** Text input field containing "Холодильники".
- URL:** Text input field containing "kholodilniki".
- Категория:** Dropdown menu with "Бытовая техника" selected.
- Meta-Title:** Text input field.
- Meta-Description:** Text area.
- H1:** Text input field.
- Текст:** Large text area.
- Checkboxes:**
 - Переключиться в режим разработчика
 - Показывать товары в листинге
 - Показывать подкатегории в листинге
- Описание:** Text area with a link "(включить редактор)".
- Изображение категории:** File upload section with "Выбрать файлы", "Файл не выбран", and "загрузить" buttons.
- Опубликована:** checkbox.

Рисунок 5.13 Форма додавання/редагування категорій

Сторінка категорії містить перелік стандартних полів: імя, урл, метадані та здатність відображати на сайті не товари в подкатегорії, а список подкатегорій.

При переході на кожен з наступних пунктів, ви потраплятимете на список конкретних типів сутностей. Наприклад, при переході в розділ «товари», ви потраплятимете на список товарів, категорій – на список категорій.

5.1.5 «Типи товарів»

Опишемо сутність «тип товару».

Название: Геймерские ноутбуки

URL: gejmerskieoutbuki

Категория: Ноутбуки

Загрузка изображений для фильтров: Файл не выбран

Дополнительные параметры товаров этого типа

Название поля	Параметры		
Разрешение	<input type="button" value="скрыть опции"/>	0 <input type="button" value="x"/>	
	<input type="text" value="1600kh900"/>	<input type="text" value="Имя на русском"/>	<input type="text" value="Влияние на цену"/> - <input type="button" value="x"/> <input type="button" value="↓"/>
	<input type="text" value="full-hd"/>	<input type="text" value="Имя на русском"/>	<input type="text" value="Влияние на цену"/> - <input type="button" value="x"/> <input type="button" value="↓"/>
	<input type="button" value="Добавить опцию"/>		

множественный выбор

Заменять родительский тип товаров
 Общий тип (применяется ко всем товарам)

Рисунок 5.14 Форма редагування/додавання типу товару

Окрім базових полів, для кожного типу характерно мати ряд фільтрів, які мають свої опції. Кожна з опцій – це конкретна характеристика товару.

На сайті вона може слугувати характеристикою пошуку товару, або впливати на його цінову категорію. Для кожної опції можна задати своє ім'я, завантажити фотографію, вказати пріоритет відображення.

Фільтри мають свої характеристики, а головне – типи.

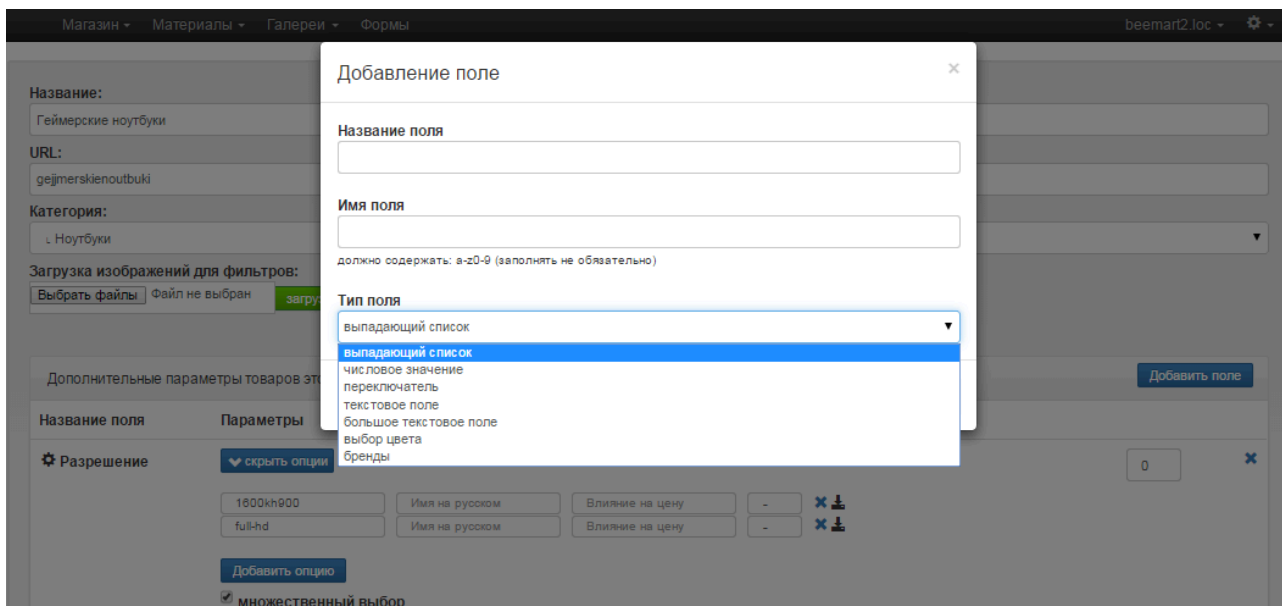


Рисунок 5.15 Форма додавання фільтра

Форма додавання поля зроблена в модальному окні для зручності повернення до основного функціоналу додавання типу. Тип поля має список вже встроєних в систему типів.

Наприклад, якщо ми створимо фільтр з типом «выбор цвета», збережемо тип і підв'яжемо його до товару, то у списку фільтрів товару з'явиться наступна панель :



Рисунок 5.16 Опція фільтра типа "кольор"

5.1.6 «Производители»

При створені виробників ми маємо змогу завантажити та опублікувати статтю про товари конкретного виробника на сайті.



The screenshot shows a web form for creating a brand. It includes the following fields and elements:

- Название бренда:** Text input field containing "asus".
- URL:** Text input field containing "asus".
- Описание: (включить редактор)**: A large empty text area for a description.
- Опубликована**: A checked checkbox for publishing.
- Логотип:**: A file upload field showing a thumbnail of "asus.jpg" with a red 'x' icon for removal.
- Buttons: "Выбрать файлы" (Choose files), "Файл не выбран" (File not selected), and "загрузить" (Upload).

Рисунок 5.17 Форма додавання бренда

Можна завантажити логотип виробника. При створенні фільтра з типом поля «бренд» будуть завантажуватися ті бренди, які ми створили у розділі «Производители».

5.1.7 «Коллекции»

Коллекції створюються для того, щоб об'єднати в собі товари з різних категорій.

Название:
Літній розпродаж

URL:

Опубликовано

Список товаров в коллекции

1. Однокамерный холодильник ATLANT MX 5810-72 ✕
2. Dell Latitude 3540 (CA002L35401EM) ✕

Навигатор категорий и товаров

Нет дочерних категорий назад

Dell Latitude 3540 (CA002L35401EM) 1
Ноутбук Acer Aspire E5-511G-P2VL

Рисунок 5.18 Форма додавання колекції

5.1.8 «Заказы»

Замовлення існують для того, щоб вести облік товарів та розраховувати клієнтів. Кожне замовлення має свій власний статус, список товарів та їх кількість. При редагуванні замовлення, ми можемо додати певні товари до замовлення, вибрати спосіб доставки і адресу доставки. Змінити ім'я та адресу замовника. Переглянути історію замовлення.

Статус: Ждет рассмотрения фильтровать параметры фильтра ⚙

Номер заказа	Статус	Дата создания	Сумма	E-mail	ФИО	Телефон	Адресс	Город	Товары
2	Ждет рассмотрения	16.05.2015 20:28	0		Александр	09312341234	ул. Просвещения 7		1. Однокамерный холодильник CANDY CCTOS 502WH ☰

параметры таблицы ⚙ | скачать таблицу ⬇

Рисунок 5.19 Лістинг замовлень

5.1.9 «Скидки»

Розділ скидок існує для того, щоб вказати скидку на певний період для категорії. Розмір знижки можна задавати як відсоток від ціни товару, так і вказати фіксовану ціну, яка буде відніматися від всіх товарів.

Название:	Літня скидка
Размер скидки:	15%
Тип скидки:	Скидка в процентах
Дата начала:	31.05.2015
Дата окончания:	17.06.2015
Категории в которой скидка будет действовать:	<ul style="list-style-type: none"> -- корневая категория -- Бытовая техника <ul style="list-style-type: none"> Холодильники Стиральные машины Газовые плиты Пылесосы Электроника

Рисунок 5.20 Форма додавання/редагування скидок

5.1.10 «Валюты»

Сrm-система здатна створювати валюти з замороженим курсом або з плаваючим.

Название на русском:	гривна
Код:	UA
Шаблон:	0.2f
Курс:	22
Замороженный курс:	21.5
<input checked="" type="checkbox"/> Опубликовано <input type="checkbox"/> Это валюта товаров по умолчанию	

Рисунок 5.21 Форма додавання/редагування валюти

5.1.11 «Настройка импорта»

Будь-якому користувачеві необхідно імпортувати товари в базу даних. Налаштування імпорту допоможе йому завантажити файл з усіма товарами. Якщо у власника сrm-системи є вже написаний скрипт, який допоможе йому

імпортувати товари, він може вказати шлях до нього і скрипт буде виконуватися. Ми можемо пропустити n-рядків вказавши їх число. Можемо вказати категорію товарів і параметри публікації.

Название:

Альтернативный обработчик:

путь от корня сайта, например public/alternative_handler.php

Пропустить первые N строк:

Поместить все товары в категорию:

Параметры публикации при добавлении новых товаров :

Добавлять товары которые отсутствуют в базе

Добавить "нет в наличии" товарам, которые отсутствуют в файле импорта

Автоматически добавлять отсутствующие параметры фильтров

Рисунок 5.22 Форма налаштування імпорту

5.1.12 «Импорт товаров»

Імпорт товарів – це одна з найважливіших речей в інтернет-магазині.

Ми маємо змогу додати .xls файл та вибрати той парсер, який ми створили заздалегідь в розділі «*Настройка импорта*». «*Загрузчик*» існує для того, щоб ми могли під'єднатися до по фтп до серверу.

Импортируемые файлы

Файл	Статус	Отчет	Дата добавления
Шины.xls	на очереди		01.06.2015 12:33

Загрузки

FTP	Статус
ftp://dev.beemart.com.ua:**@dev.beemart.com.ua	–

Рисунок 5.22 Лістинг імпорту

5.1.13 «Импорт картинок для товаров»

Функціонал розроблений для імпорту картинок до товарів. На вході йому необхідно завантажити архів з фотографіями. Назва кожної фотографії повинна співпадати з кодом товару.

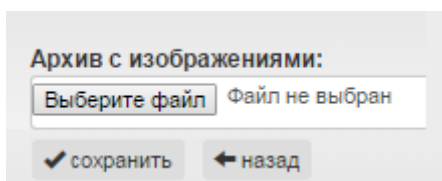


Рисунок 5.23 Поле завантаження архіву

5.1.14 «Импорт типов товаров»

Аналогічний функціонал з імпортом типів товару. Обираємо файл та зберігаємо його:

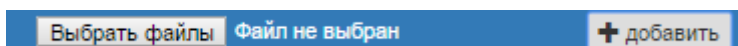


Рисунок 5.24 Поле імпорту типу товару

5.2 Розрахунок контрольного прикладу та отримання вихідних даних

Сайт простий у своєму використанні і не потребує яких-небудь специфічних пояснень.

Рисунок 5.26 - відображена карточка товару. В адмінці ми завантажили для неї фотографію та вказали всі характеристики.



Ноутбук Acer Aspire E5-511G-P2VL

Артикул: не указан **Нет в наличии**

Магазин: [fotos.ua](#)

Оплата

Наличными

8399 грн [В корзину](#)

В других **11** магазинах по цене **16749...27490 грн**

[← Возврат товара](#)

[🕒 Быстрый заказ](#)

Общее число пикселей - 16.4 млн.
 Тип видеосистемы - отсутствует
 Тип матрицы - CCD
 Максимальное разрешение - 4608x3456 px
 Размеры - 114x79x86 мм
 Вес - 403 г

Добавить к сравнению

[Характеристики](#)
[Отзывы](#)
[Видео](#)

Общее число пикселей	27.7 млн
Тип видеосистемы	зеркальный
Тип фотоаппарата	Зеркальный
Тип матрицы	CMOS
Максимальное разрешение по X	6016 pix
Максимальное разрешение по Y	4000 pix
Кроп-фактор	1.5
Сенсорный экран	Нет
Ручная фокусировка	Да
Ручные настройки выдержки и диафрагмы	Да
Режим серийной съемки	Нет
Поддержка RAW	Да
Поддержка TIFF	Нет
Поддержка RAW	Да
Поддержка TIFF	Нет

Общее число пикселей	27.7 млн
Тип видеосистемы	зеркальный
Тип фотоаппарата	Зеркальный
Тип матрицы	CMOS
Максимальное разрешение по X	6016 pix
Максимальное разрешение по Y	4000 pix
Кроп-фактор	1.5
Сенсорный экран	Нет
Ручная фокусировка	Да
Ручные настройки выдержки и диафрагмы	Да
Режим серийной съемки	Нет
Поддержка RAW	Да
Поддержка TIFF	Нет
Поддержка RAW	Да
Поддержка TIFF	Нет

Описание

Описание отсутствует.

☀️ Похожие предложения

[Все товары](#)



Рисунок 5.26 Карточка товару

На рис. 5.27 відображена сторінка категорій.

Можемо побачити блок фільтрів та блок порівняння товарів в лівій частині рисунка. Відсортувати товари за ціною, популярністю або за розміром скидок.

Якщо вам не зручно переглядати товари у вигляді лістингу, то ви завжди зможете увімкнути табличний вигляд.

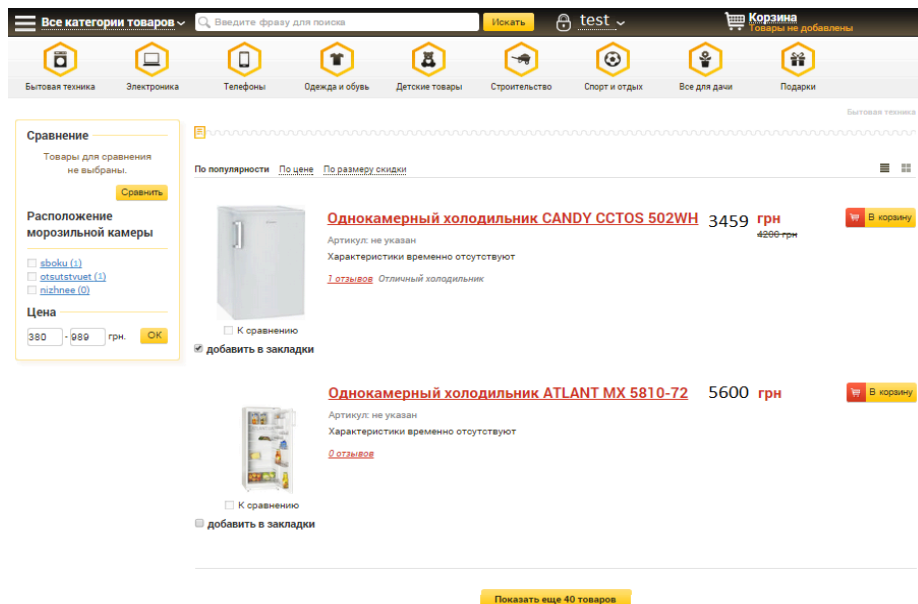


Рисунок 5.27 Сторінка категорій

При додаванні товару до корзини з'являється рор-уп віконце з товарами.

Ми можемо видалити товар, або змінити кількість товару. Якщо ви бажаєте продовжити покупки, то ви можете зачинити цей рор-уп.

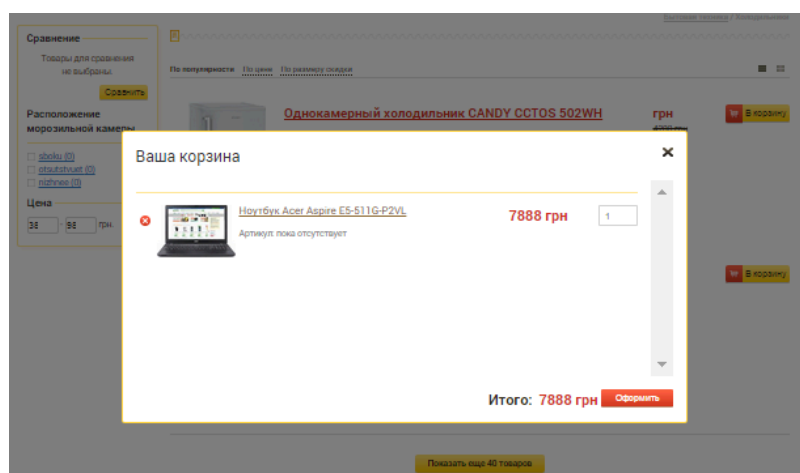


Рисунок 5.28 Корзина. Рор-уп

На Рис. 5.29 відображена головна сторінка сайту.

BEEMART Торговий центр

Киев Доставка и оплата

Info@site.com Контактные данные

+38044 777-55-33 Работаем с 9:00 до 19:00

Все категории товаров

Введите фразу для поиска

Искать

valleon

Корзина

Бытовая техника Электроника Телефоны Одежда и обувь Детские товары Строительство Спорт и отдых Все для дачи Подарки

Суперцена 30 апреля – 21 мая

Ноутбук Acer Aspire ES1

Купить

Велосипеды для детей и взрослых

Велосипеды для детей и взрослых

★ **Лучшие цены** Все товары →

Мелкая игрушка Ватви Виван Пук 60 см (MP 0399 F)	Однокomorный холодильник SANDY CCTOS 502WH	Ноутбук Acer Aspire ES-511G P2V	Однокomorный холодильник ATLANT МК 9810-72	Dell Latitude 3540 (CA002L35401EM)
665 грн	3800 грн	7888 грн	9890 грн	14000 грн
В корзину	В корзину	В корзину	В корзину	В корзину

☀ **Скидки дня** Все товары →

Однокomorный холодильник ATLANT МК 9810-72	Dell Latitude 3540 (CA002L35401EM)	Ноутбук Acer Aspire ES-511G P2 V	Мелкая игрушка Ватви Виван Пук 60 см (MP 0399 F)	Однокomorный холодильник SANDY CCTOS 502WH
9890 грн	14000 грн	7888 грн	665 грн	3800 грн
В корзину	В корзину	В корзину	В корзину	В корзину

♥ **Распродажа** Все товары →

Горка Ching Ching Каскад с баскетбольным кольцом SL-11	Горка Ching Ching Каскад с баскетбольным кольцом SL-11	Горка Ching Ching Каскад с баскетбольным кольцом SL-11	Горка Ching Ching Каскад с баскетбольным кольцом SL-11	Горка Ching Ching Каскад с баскетбольным кольцом SL-11
7890 грн	7890 грн	7890 грн	7890 грн	7890 грн
В корзину	В корзину	В корзину	В корзину	В корзину

♦ **У нас покупают**

Холодильники Стиральные машины Газовые плиты Пылесосы Моноблоки Акустические системы МВУ СВЧ печи Чайники Утюги

Покупателям

- Купить на Виванмерте
- Доставка и оплата товаров
- Магазины
- Приложения для Windows Phone

Магазинам

- Виван. История успеха
- Торговля на Виванмерте
- Как шар думать?
- Добавить магазин
- Платные продвижения

О компании

- Что такое Виванмерте
- Работа на Виванмерте
- Отзывы
- Контакты

Полезная информация

- Пользовательское соглашение
- Соглашение о конфиденциальности и персональных данных
- Договор оферты
- Реклама на сайте

Мы в социальных сетях

© Beemart, 2014 info@site.com

Рисунок 5.29 Головна сторінка сайту

ВИСНОВКИ

Для розробки інтернет-магазину був проведений аналіз предметної області, було складено технічне завдання, на підставі якого був створений дизайн майбутнього сайту. Створений сайт повністю відповідає вимогам, пред'явлених до проекту в технічному завданні.

У рамках роботи були отримані наступні теоретичні та практичні результати:

В першому розділі був проведений аналіз предметної області, та аналіз існуючих систем керування контентом. Також розглянуто організаційну структуру Інтернет-магазину та наприкінці розділу виконана постановка задачі.

В другому розділі було побудовано систему дерева цілей, у якій зображено усі пункти, які необхідно досягнути для створення підсистеми.

Третій розділ присвячений побудові структури та визначенню функціональних можливостей підсистеми. А саме у ньому розглянуто загальне представлення підсистеми, архітектури сайту.

В четвертому розділі проводиться визначення основних сутностей. Опис програмного забезпечення та опис інтерфейсу програми.

В п'ятому розділі приводиться контрольний приклад використання програмного продукту. В ньому можна побачити увесь функціонал інтернет-магазину, а також його інтерфейс.

При створенні сайту були використані найпопулярніші технології, такі як PHP, Zend Framework та MySQL. Отриманий сайт є адаптивним і має чітку логічну структуру, що без особливих зусиль дозволить прив'язати даний веб-сайт до будь-якої системи управління сайтами, використовувати і розвивати його в подальшому.

Розроблений магазин має зрозумілий і простий користувальницький інтерфейс, що забезпечить схвалення сайту з боку клієнтів Інтернет-магазину.

Реалізований сайт дозволяє відобразити інформацію про товари, про категорії товарів, дозволяє клієнтам додавати товари в кошик, оформляти замовлення.

Даний веб-сайт створений для російськомовної аудиторії. Сайт можна використовувати для продажу товарів Інтернет-магазину на території країн СНД.

Головними позитивними ознаками дипломної роботи є:

- Простота у використанні програмного продукту, як для користувачів, так і для адміністратора;
- Широкий функціонал;
- Великий потенціал, для подальшого розвитку.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Боуэн Р., Лиска А. Apache. Настольная книга администратора: Пер. с англ. - М.: ДИАСофт ЮП, 2002. - 384 с.
2. Гешвинде Э., Шенинг Г. Разработка WEB- приложений на PHP и PostgreSQL: Руководство разработчика и администратора: Пер. с англ. - М.: ДИАСофт ЮП, 2002. - 608 с.
3. Апопій В. В. Інтернет-торгівля: проблеми і перспективи розвитку / Апопій В. В. // Регіональна економіка. - 2003. - № 1. - с. 25.
4. Скотт Хокинс. Администрирование веб-сервера Apache и руководство по электронной коммерции = Apache Web Server Administration and e-Commerce Handbook. — М.: Вильямс, 2001. — 336 с.
5. Гайна Г.А. Основы проектирования баз данных: Навчальний посібник. — К.: КНУБА, 2005. — 204 с.
6. PHP и MySQL. Создание интернет-магазина: Кристиан Дари, Эмилиан Баланеску — Санкт-Петербург, Вильямс, 2010 г.- 640 с.
7. Гончаров А. Ю. Web-дизайн: HTML, JavaScript и CSS. Карманный справочник.. — "КУДИЦ-ПРЕСС", 2007. — 320 с.
8. Мартін Фаулер "Шаблони корпоративних додатків".
9. Гарсиа-Молина Г., Ульман Дж., Уидом Дж. Системы баз данных. Полный курс. — М.: Вильямс, 2003. — 1088 с.
10. Томас А. Пауелл. Ajax. Настільна книга програміста / Томас А. Пауелл. - М: Ексмо, 2009. - 720 стор.
11. Граді Буч "Об'єктно-орієнтований аналіз та проектування з прикладами додатків".
12. Кришна Шасанкар «Zend Framework 2.0. Разработка веб-приложений» Питер, 2014 год, 208 стр.
13. Яковлев Алексей Александрович. Раскрутка и продвижение сайтов: основы, секреты, трюки.. — СПб.: БХВ-Петербург, 2007. — 336 с.

14. Колісниченко Д. Н. PHP 5/6 і MySQL 6. Розробка Web-додатків / Денис Колісниченко. - СПб: Вільямс, 2009. - 607 с. : Ил., Табл.
15. Прохоренок Н. HTML, JavaScript, PHP і MySQL. Джентльменський набір Web-майстра / Микола Прохоренок. - СПб: БХВ-Петербург, 2010. - 900 с.
16. Томсон Л., Веллінг Л. Розробка Web-додатків на PHP і MySQL. - К. : "ДіаСофт", 2001. - 672 с.
17. Денні Гудман, Майкл Моррісон. JavaScript. Біблія користувача. - СПб: Вільямс, 2006. - 1184 с.
18. Крістіан Даруй, Богдан Брінзаре, Філіп Черчез-Тоза, Міхай бусика. AJAX і PHP. Розробка динамічних веб-додатків. - СПб: Символ-Плюс, 2009. - 336 с.
19. Е. Фрімен, Е. Фрімен "Паттерни проектування".
20. Е. Гамма, Р. Хелм, Р. Джонсон, Дж. Вліссідес "Прийоми об'єктно-орієнтованого проектування. Патерни проектування".
21. Голдовский И. Безопасность платежей в Интернете. Санкт Петербург: 2002.
22. Леонов С. Тестирование производительности веб-приложений // Открытые системы.СУБД. 2014. URL: Открытые системы.СУБД. (дата звернення: 7.10.2017).
23. IBM Design. Design in motion [Електроний ресурс] // IBM Design: [сайт]. [2015]. URL: <http://www-01.ibm.com/software/ucd/designconcepts/whatisUXD.html> (дата звернення: 11.10.2017).
24. ГОСТ 34. 601-90. Автоматизовані системи. Стадії створення. / М.: Видавництво стандартів, 1991.
25. The 9 th International scientific and practical conference "World science: problems, prospects and innovations" (May 19-21, 2021) Perfect Publishing, Toronto, Canada. 2021. 678 p.