

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

**Інститут інформатики та радіоелектроніки**  
**Факультет комп'ютерних наук та технологій**  
(повне найменування інституту, факультету)

**Кафедра комп'ютерних систем та мереж**  
(повне найменування кафедри)

## Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти (освітній ступінь))

на тему **РОЗРОБКА ТА ПРОСУВАННЯ В ІНТЕРНЕТ**  
**СИСТЕМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ**

Виконав: студент 4 курсу, групи КНТ-518сп  
спеціальності \_\_\_\_\_

123 «Комп'ютерна інженерія»

(код і найменування спеціальності)

Освітня програма (спеціалізація) \_\_\_\_\_

«Комп'ютерна інженерія»

Наконечний Олександр Дмитрович

(прізвище та ініціали)

Керівник Касьян М.М.

(прізвище та ініціали)

Рецензент Качан О.І.

(прізвище та ініціали)

м. Запоріжжя  
2021 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
**Національний університет «Запорізька політехніка»**  
(повне найменування вищого навчального закладу.)

Інститут, факультет інформатики та радіоелектроніки, комп'ютерних наук і технологій  
 Кафедра «Комп'ютерні системи та мережі»  
 Ступінь вищої освіти (освітній ступінь) бакалаврський  
 Спеціальність 123 Комп'ютерна інженерія  
(код і найменування)  
 Освітня програма (спеціалізація) Комп'ютерна інженерія  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри Кудерметов Р.К.

“ ” 2021 року

**ЗАВДАННЯ**  
**НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА**

Наконечного Олексія Дмитровича

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка та просування в Інтернет системи електронної комерції

керівник проекту (роботи) Касьян Микола Миколайович, к. т. н., доцент

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

затверджені наказом вищого навчального закладу від “17” березня 2021 року № 81

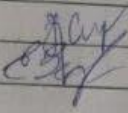
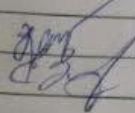
2. Строк подання студентом проекту (роботи) 10 травня 2021 року

3. Вихідні дані до проекту (роботи) необмежена кількість категорій і товарів в кожній категорії; зберігання товарів в базі даних; поділ на адміністраторську та клієнтську частину; кросбраузерність; платформа PHP, MySQL, Apache; просування розробленої системи в Інтернет

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області; аналіз методів і засобів розробки та просування в мережі Інтернет; проектування системи електронної комерції; розробка системи електронної комерції; тестування розробленої системи; SEO-оптимізація і просування розробленої системи електронної комерції

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) аналіз предметної області; структурна схема системи електронної комерції; алгоритм роботи системи; структура бази даних системи; інтерфейс системи електронної комерції



## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	Касьян М. М., доцент		
Нормоконтроль	Зелік О.В., асистент		

7. Дата видачі завдання 01.03.2021 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз технічного завдання	01.03.2021 р.	
2	Проектування системи електронної комерції	14.03.2021 р.	
3	Розробка бази даних	21.03.2021 р.	
4	Розробка інтерфейсу системи	01.04.2021 р.	
5	Розробка програмного коду системи	07.04.2021 р.	
6	Пошукова оптимізація розробленої системи	14.04.2021 р.	
7	Тестування розробленої системи	23.04.2021 р.	
8	Оформлення пояснювальної записки	01.05.2021 р.	
9	Оформлення графічного матеріалу роботи	06.05.2021 р.	
10	Оформлення презентації	10.05.2021 р.	

Студент  (підпис)О.Д. Наконечний  
(ініціали та прізвище)Керівник проекту (роботи)  (підпис)М.М. Касьян  
(ініціали та прізвище)

## РЕФЕРАТ

ПЗ: 75 с., 38 рис., 3 табл., 9 лістингів, 15 посилань.

АРАСНЕ, MVC, MYSQL, PHP, SEO, БАЗА ДАНИХ, ЕЛЕКТРОННА КОМЕРЦІЯ, ПОШУКОВА ОПТИМІЗАЦІЯ, СЕМАНТИЧНЕ ЯДРО, СИСТЕМА

Об'єкт проектування в дипломній роботі є система електронної комерції з SEO-оптимізованим контентом.

Метою дипломної роботи є автоматизація процесу замовлення та покупки товару через мережу Інтернет, скорочення часу на пошук товару, підвищення продажів за допомогою просування в пошукових системах. За допомогою розробленої системи здійснюється залучення потенційних покупців та ознайомлення їх з товарами та послугами компанії.

Для досягнення поставленої мети в роботі плануються та виконуються наступні задачі:

- аналіз предметної області та, аналіз методів і засобів розробки та просування в мережі Інтернет;
- проектування системи електронної комерції;
- розробка та тестування розробленої системи електронної комерції;
- SEO-оптимізація і просування розробленої системи електронної комерції.

В першому розділі роботи проведено аналіз сучасних систем електронної комерції. Обґрунтовано вибір мов програмування для розробки системи та архітектури MVC. В другому розділі було розроблено структурну схему та алгоритм функціонування системи, реалізовано базу даних, спроектовано кроки з SEO-оптимізації розробленої системи. В третьому розділі розроблено інтерфейс користувача та програмний код системи на мові PHP, контент системи оптимізовано згідно розробленого семантичного ядра.

## ЗМІСТ

Вступ.....	7
1 Аналіз технічного завдання та технологій розробки системи .....	9
Огляд аналогічних систем електронної комерції.....	9
Аналіз принципів побудови систем електронної комерції .....	12
Порівняльний аналіз технологій розробки .....	15
Порівняння PHP з Ruby та Python .....	17
Порівняння PHP з Java .....	19
Порівняння PHP з ASP.NET .....	19
Аналіз архітектурних шаблонів та фреймворків.....	21
Архітектурний шаблон MVC.....	23
Zend Framework .....	25
Фреймворк Symfony .....	26
Фреймворк Codeigniter.....	26
Фреймворк Laravel .....	27
2 Проектування системи електронної комерції.....	29
Розробка структурної схеми системи.....	29
Архітектура електронного магазину .....	29
Програмні засоби для розробки та роботи системи електронної комерції ...	31
Пошукова оптимізація контенту .....	33
Розробка семантичного ядра системи.....	36
Оптимізація сторінок .....	38
Розробка алгоритму роботи системи.....	42
Розробка бази даних системи .....	45
3 Програмна реалізація та тестування системи.....	51
Розробка інтерфейсу системи .....	51
Розробка програмного коду .....	57
Конфігурування та тестування системи.....	66

Висновки.....	73
Перелік джерел посилання .....	74
Перелік графічних матеріалів:	
Плакати 1 - Огляд існуючих систем електронної комерції	
Плакати 2 - Аналіз технологій, що використовуються	
Плакати 3 - Структура системи електронної комерції	
Плакати 4 - Пошукова оптимізація контенту	
Плакати 5 - Інтерфейс системи електронної комерції	

## ВСТУП

Стрімкий розвиток глобальної мережі Інтернет, використання новітніх технологій і засобів комунікацій, як в комерційній діяльності, так і в повсякденному житті зумовило появу таких нових економічних явищ, як електронна комерція. Електронна комерція - це доволі широка галузь економіки, що включає до себе багато видів діяльності, а саме інтернет-магазини, інтернет-маркетинг і реклама, різноманітні платіжні системи та інше. Сьогодні все більш магазинів від невеличких та вузькоспеціалізованих до великих торгівельних мереж прагнуть мати своє представництво в мережі Інтернет, а деякі взагалі здійснюють свою діяльність тільки в Інтернет. Це стає можливим завдяки тому, що Інтернет надає нові канали збуту продукції та відкриває широкі можливості з реклами і маркетингових досліджень.

Система електронної комерції - це віртуальний магазин з реальними товарами, інструмент з продажу товарів і послуг через мережу Інтернет.

Система електронної комерції має низку переваг над традиційним магазином. Доступ до неї мають потенційні покупці з усього світу, тому є можливість розширення географії продажів. Інтернет-магазин не потребує великої кількості обслуговуючого персоналу - є можливість обійтися лише одним консультантом там, де звичайні торгівельні заклади обслуговуються декількома. До того ж, нема обмежень щодо кількості представлених товарів. Інформація про клієнта може зберігатися в базі даних, тому у випадку з розкруткою і просуванням інтернет-магазину є можливість сфокусуватися тільки на потенційно зацікавлених користувачах. Ще одна перевага полягає у автоматизації таких процесів, як прийом замовлень, отримання заявок і т. і. Список переваг можна продовжувати ще довго.

Крім того, власник системи електронної комерції може аналізувати поведінку покупців на сайті, інтерес до товарів, і таким чином підлаштовувати

свій віртуальний магазин під актуальні потреби споживачів. Це також дозволить значно збільшити обсяг продажів.

До того ж, система електронної комерції зручна і для покупця. Великий перелік товарів, зручний вибір, можливість порівняти характеристики товарів, побачити відгуки, цілодобовий доступ до магазину і товарам, зручні варіанти оплати та доставки.

В дипломній роботі в якості об'єкту проектування було обрано сервіс з продажу та обслуговування комп'ютерної техніки.

Основною метою роботи було реклама товарів і послуг компанії в мережі Інтернет та підвищення продажів за допомогою просування в пошукових системах. Для досягнення поставленої мети в системі необхідно реалізувати такі можливості:

- засоби автоматизації електронної комерції;
- пошук в системі;
- оформлення замовлень;
- порівняння товарів;
- зв'язок з соціальними мережами;
- наявність мікроданих на сторінці товару;
- можливість надавати ціни з сайтів конкурентів;
- механізм відгуків;
- розробка та втілення семантичного ядра системи для пошукової оптимізації.



# 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ ТА ТЕХНОЛОГІЙ РОЗРОБКИ СИСТЕМИ

## Огляд аналогічних систем електронної комерції

Практично всі торгівельні мережі пов'язані з продажем комп'ютерної техніки поряд з реальними торгівельними точками мають й інтернет-магазини. Наприклад, «Фокстрот» (рис.1.1), «Алло» (рис.1.2), «Ельдорадо» (рис.1.3), «Comfy» (рис.1.4) та інші. Деякі, наприклад «Rozetka» (рис.1.5), спеціалізуються саме на продаж через Інтернет. Але всі вони мають дуже схожий функціонал та інтерфейс своїх інтернет-магазинів. Магазины займаються продажем товарів наступних категорій:

- мобільна техніка;
- комп'ютери та ноутбуки;
- телевізори та аудіо;
- фото і відео;
- велика і дрібна побутова техніка.

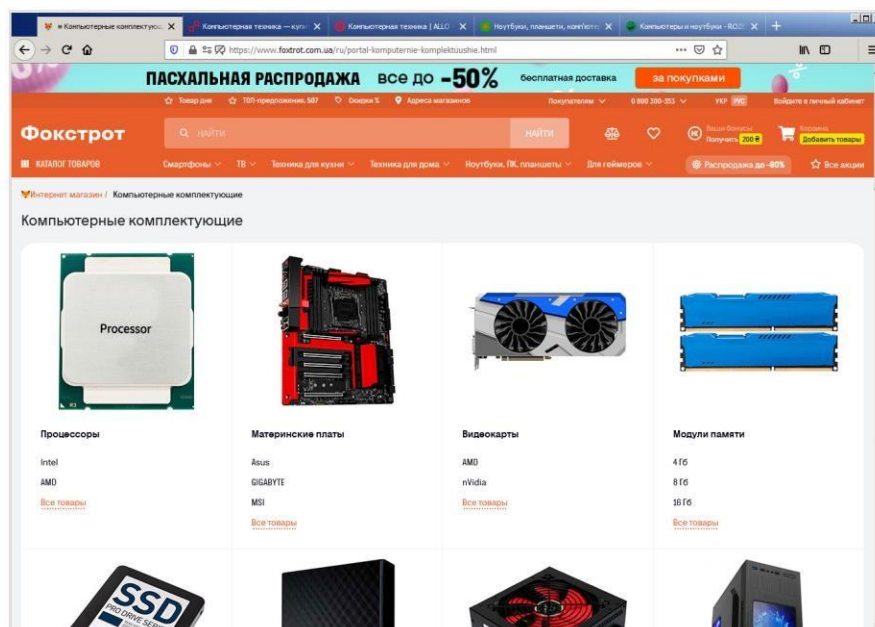


Рисунок 1.1 – Інтерфейс інтернет-магазину «Фокстрот»

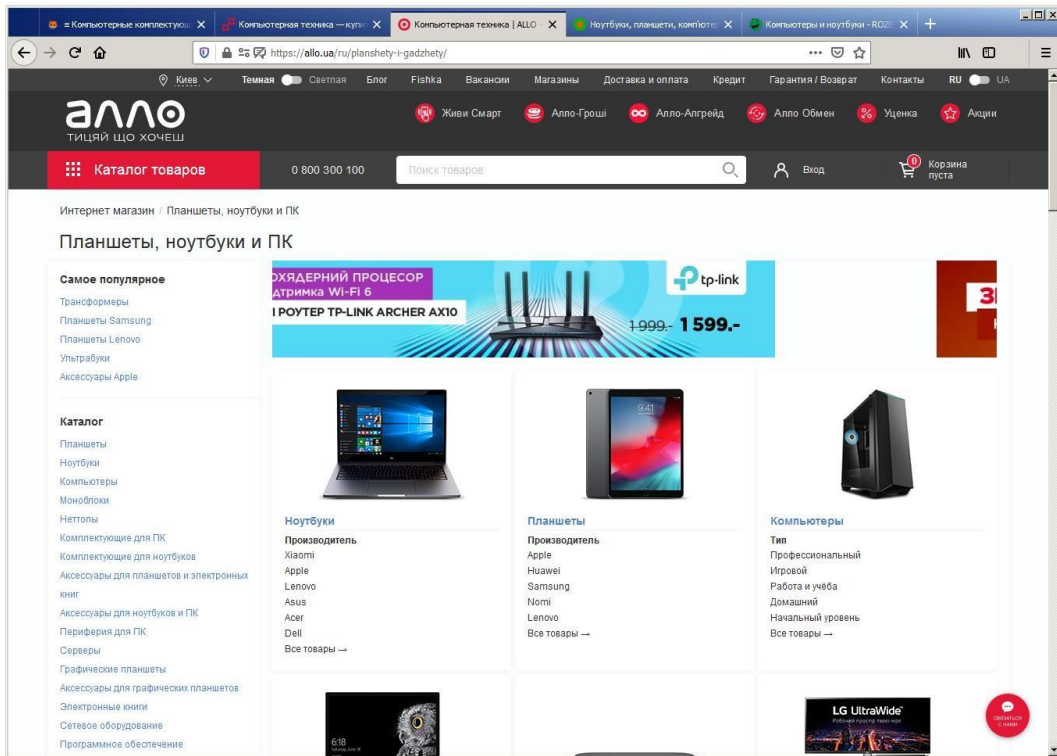


Рисунок 1.2 – Интерфейс интернет-магазину «Алло»

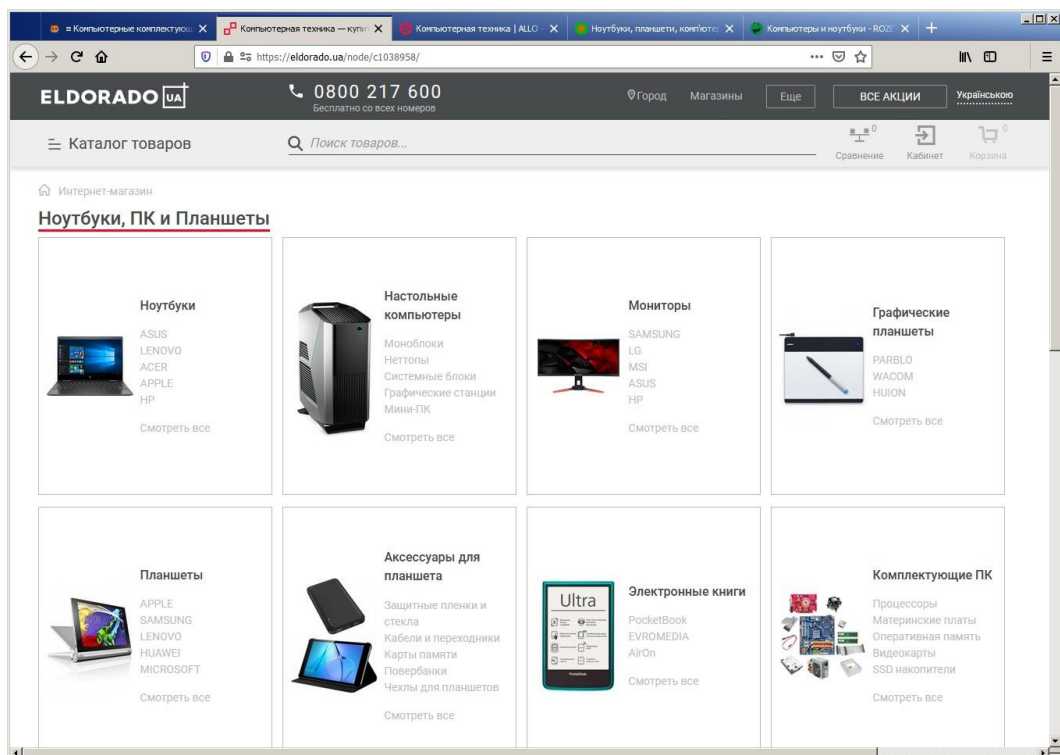


Рисунок 1.3 – Интерфейс интернет-магазину «Ельдорадо»

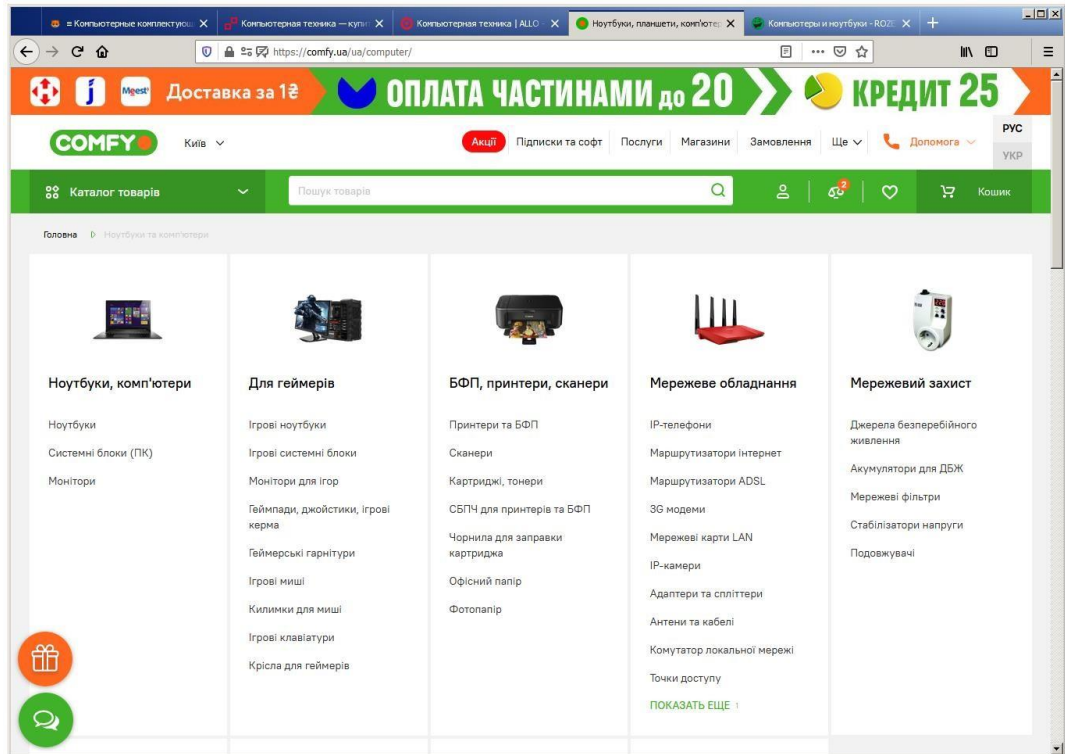


Рисунок 1.4 – Інтерфейс інтернет-магазину «Comfy»

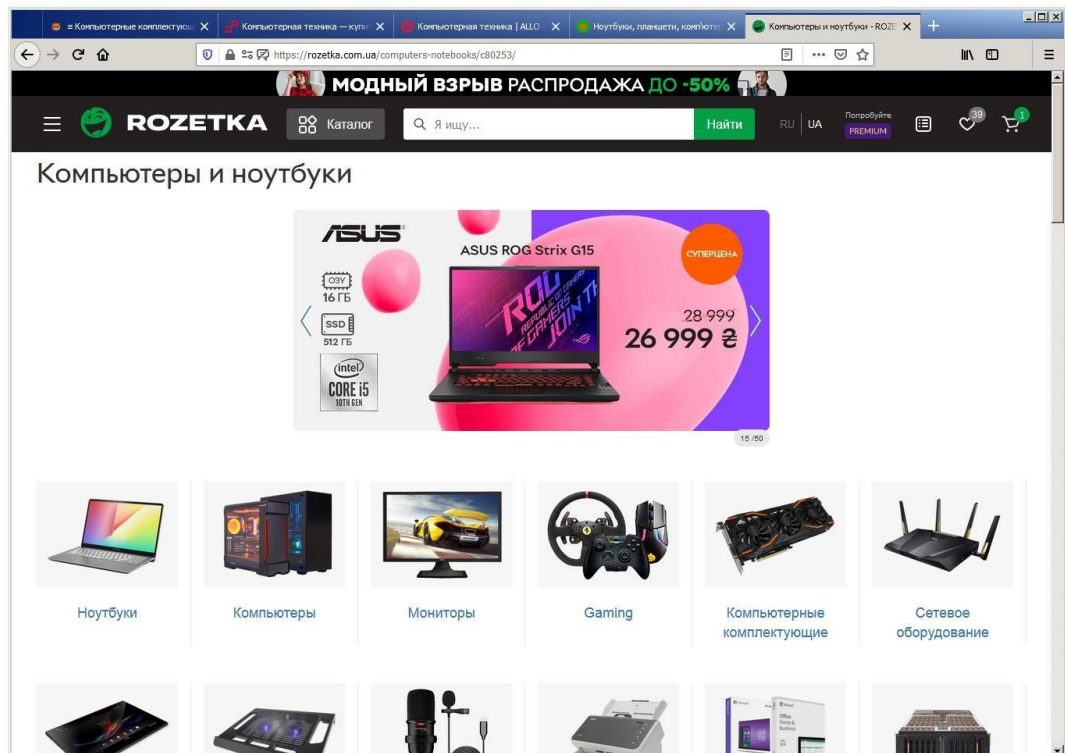


Рисунок 1.5 – Інтерфейс інтернет-магазину «Rozetka»

І в загальному переліку товарів в кожній позиції, і на сторінці кожного товару є кнопка «Замовити». Натиснувши її, клієнт додає обраний товар до свого кошику. Кошик - це список товарів, що покупець обрав в магазині, так само, як він кладе їх в свій кошик в реальному супермаркеті.

При цьому відразу ж розраховується загальна вартість обраних товарів та загальна вага замовлення, що важливе для їх доставки. Загальне число товарів висвічується поряд у верхньому куті сторінки поруч з зображення кошику.

Після переходу на сторінку кошика, можна оформити замовлення, натиснув відповідну кнопку, або повернутися до вибору товарів. Також можна видалити будь-яку позицію або очистити всю корзину, змінити кількість одиниць будь-якого товару, що перебувають у кошику. При цьому коригується загальна вартість та ваги покупки.

Це зручно, особливо, якщо розірвався зв'язок або покупець закрив сторінку, обрані товари будуть збережені в його кошику.

Як правило, для замовлення товарів необхідно пройти ідентифікацію - або зареєструватися, або зайти під своїм логіном.

Можна обирати варіанти оплати (банківською карткою, при доставці готівкою, інші) та способи доставки (Нова пошта, Укрпошта, кур'єром по місту та інші).

### **Аналіз принципів побудови систем електронної комерції**

Розглянемо, перелічені вище, деякі популярні в Україні інтернет-магазини з точки зору їх технічної реалізації. Визначити за допомогою яких технологій створено веб-сайт можна, наприклад, за допомогою спеціалізованого ресурсу [builtwith.com](http://builtwith.com) [1].

Інтернет-магазин «Фокстрот» (<http://www.foxtrot.com.ua>) виконаний на базі технології, що базується на Windows-системах з доступом до MS SQL. Веб-

застосунок написано за допомогою відповідного фреймворку як набір файлів Active Server Pages (ASP.NET MVC), що представляють собою розширення HTML з доданими інструкціями на мовах C# або VBScript. Ці інструкції виконуються на сервері при запиті даної сторінки і управляють формуванням відповіді користувачу. Доступ до баз даних здійснюється за допомогою спеціального об'єкта ADO (ActiveDataObject). ADO є надбудовою над OLE-DB, який дозволяє єдиним чином звертатися до даних різного виду (наприклад, до електронних таблиць, текстових файлів, баз у форматах Paradox, dBase, Access, MS-SQL і будь-яким іншим через інтерфейс ODBC). Обгорткою ADO є Entity Framework для зручної роботи з БД. Також використовує бібліотеки і функції JavaScript jQuery, Bootstrap.js, Angular JS та веб-сервер IIS.

Для порівняння технології, що використовуються в деяких інших популярних інтернет-магазинах, наведені в таблиці 1.1.

Таблиця 1.1 – Технології популярних інтернет-магазинів

	ALLO.UA	ELDORADO.UA	COMFY.UA	ROZETKA.COM.UA
Серверна мова	PHP	PHP JS	PHP	PHP JS
Фреймворк сервер	PHP 7	Express		Express
Бібліотеки JS	jQuery Core-js	jQuery Redux Lodash	jQuery Lodash	jQuery Core-js Lodash
Фреймворк клієнт	Handlebars Vue	React	Vue	Angular React
Веб-сервер	Nginx	Nginx	Nginx	Nginx

Магазин займається продажем товарів наступних категорій (рис.1.1):

- смартфони;
- телевізори;

- техніка для кухні;
- техніка для дому;
- ноутбуки, ПК, планшети;
- для геймерів.

Реєстрація проходить при оформленні замовлення лише за декілька кроків та не займає багато часу. Є зручна можливість відновлення забутого паролю - його просто висилають на вказану інтернет-адресу. Адреса електронної пошти використовуються в якості логіна.

Для здійснення покупки товару, його потрібно покласти до кошику для покупок. Товар поміщається в кошик натисканням кнопки «Купити» і далі буде запропоновано продовжити вибір товарів, або перейти в корзину для оформлення замовлення.

Натискання кнопки «Купити» ні до чого не зобов'язує. Завжди можна видалити товар з кошика, відкласти його для подальшої покупки або взагалі не робити замовлення.

Подивитися вміст кошика можна в будь-який момент натисканням кнопки «Переглянути корзину» в заголовку віконця корзини або піктограми із зображенням кошика в правому верхньому куті екрану.

Натисканням кнопки «Оформити замовлення» починається процес оформлення замовлення.

Оформлення складається з наступних кроків:

- ідентифікація користувача;
- вибір адреси доставки;
- вибір способу доставки;
- вибір способу оплати;
- підтвердження замовлення (замовлення отримує номер та приймається в обробку).

На будь-якому кроці можна повернутися назад та виправити параметри замовлення. На останньому кроці потрібно перевірити всі дані і підтвердити

замовлення, натиснувши кнопку «Підтвердити замовлення». Тільки після підтвердження замовлення вважається прийнятим.

Замовленню присвоюється номер та клієнту на e-mail відправляється повідомлення про прийняття замовлення.

Крім того, на останньому кроці можна зробити відмітку, щоб параметри цього замовлення використовувалися в подальшому для замовлення за 1 крок з метою скорочення та спрощення оформлень замовлень. Без такої відмітки в якості параметрів «заказ за 1 крок» буде використовуватись параметри попереднього замовлення.

### **Порівняльний аналіз технологій розробки**

В загальному випадку системи електронної комерції складається з наступних функціональних частин:

- каталог товарів;
- система пошуку;
- кошик користувача;
- реєстраційна форма;
- форма відправки замовлення.

Каталог являє собою складну і багаторівневу структуру даних, що повинна простим та зрозумілим способом робити 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. і. Таке упорядкування просто необхідно для зручного та швидкого пошуку та замовлення товарів.

Пошукова система є обов'язковим елементом динамічного каталогу та реалізується на боці сервера. Незважаючи на те, що каталог забезпечує упорядкування та групування даних, пошукова система дає користувачу

можливість швидкого пошуку інформації, що особливо важливо, коли каталог є досить розгалуженою структурою даних з великою кількістю розділів, підрозділів та товарів. Тому користувач погано уявляє в якому розділі може перебувати товар, що його цікавить та чи є він взагалі в каталозі. Пошукова система інколи дозволяє значно скоротити кількість переходів між сторінками каталогу для доступу до потрібної інформації.

Особливість реалізації пошуку в Інтернеті полягає в тому, що робиться вибірка всіх записів, що задовольняють умовам запиту. Цей механізм пошуку називають пошуком з надлишком. У разі великої вибірки даних виведення результатів пошуку робиться посторінково для того, щоб відвідувачам не доводилося довго чекати завантаження всієї вибірки, що може включати в себе сотні, тисячі або більше записів. До того ж, як правило, відвідувачі не переглядають всі сторінки вибірки, обмежуючись двома або трьома. Тому такий механізм пошуку часто працює дуже повільно та неефективно. Але він все ж дозволяє здійснювати вибірку однакових товарів від різних постачальників, порівнювати їх характеристики та обирати найкращий варіант.

Кошик користувача представляє собою деякий масив даних, що служить для зберігання товару, який замовив користувач.

Реєстраційна форма служить для введення персональних даних користувачів, що в подальшому використовуються для їх ідентифікації між сеансами роботи з системою електронної комерції. Ця інформація може зберігатися як на боці сервера, так і на боці клієнта.

Форма відправки замовлення служить для введення контактної інформації покупця та відправлення її разом з замовленням продавцю.

Систему електронної комерції можна реалізовувати як на боці сервера, так і на боці клієнта. В першому варіанті використовуються серверні мови програмування, наприклад, PHP, Python, ASP, JSP, Ruby, JavaScript або інші. В другому варіанті лише JavaScript. Вибір конкретної реалізації залежить від багатьох факторів, які впливають з завдання, що ставляться.



Щодо вибору серверної мови програмування, то був проведений порівняльний аналіз найбільш поширених мов програмування (рис.1.6) [2].

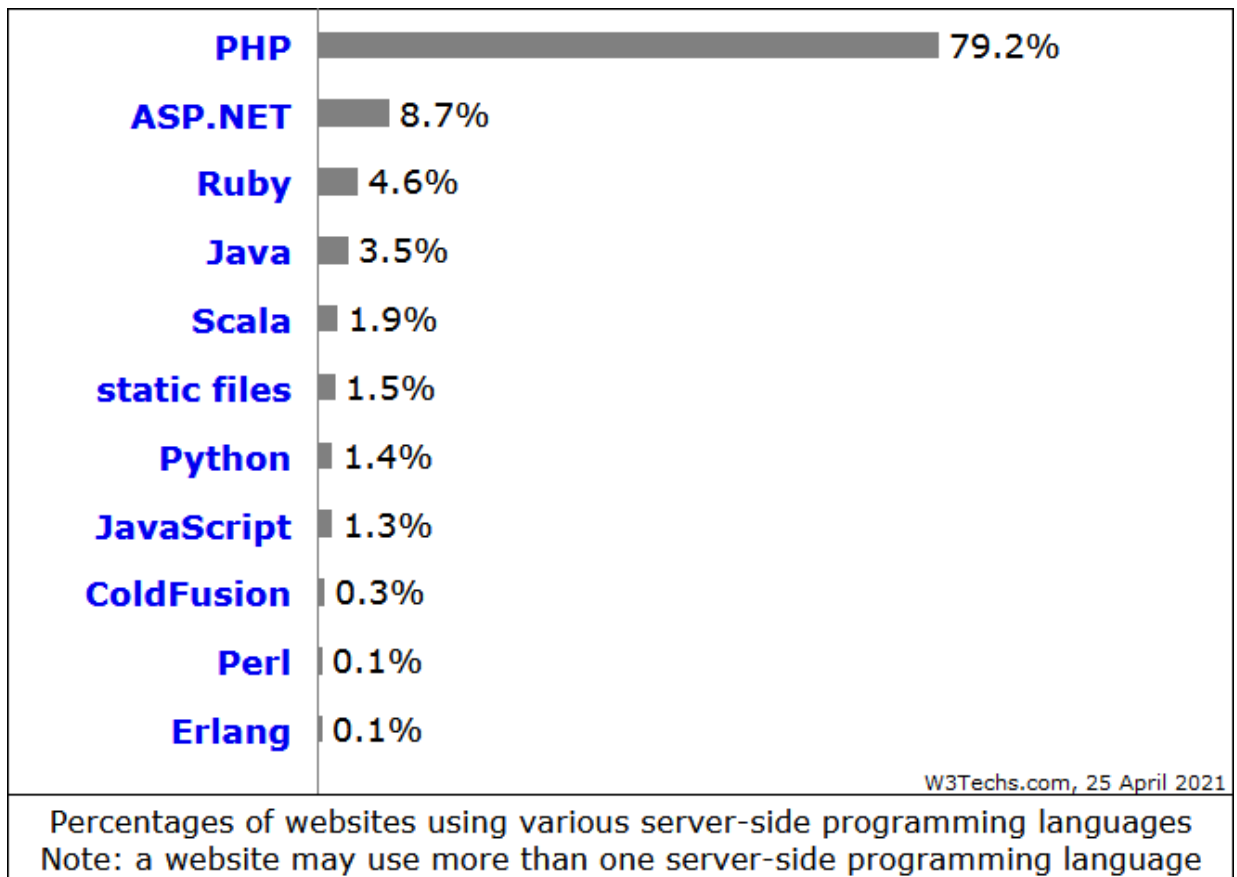


Рисунок 1.6 – Найбільш поширені серверні мови програмування

Порівнювалась мова PHP, як найпопулярніша на сьогодні мова, з іншими мовами програмування веб-застосунків.

### **Порівняння PHP з Ruby та Python**

Більшість завдань можна виконати на будь-якому з розглянутих серверних мов програмування. Але при виборі мови варто орієнтуватися на конкретну ситуацію.

Якщо ви тільки починаєте вивчати веб-програмування, то після освоєння основ мови доведеться зіткнутися з необхідністю вивчення будь-якого фреймворка на його основі. Веб-фреймворки значно полегшують створення додатків.

Більшість сайтів мають набір базових функцій: обробка сесій і авторизація, валідація запитів і т. і. Фреймворк звільняє від необхідності переписувати всі ці функції заново при створенні сайту. Веб-програмісту залишиться спроектувати і реалізувати лише функції взаємодії сайту з користувачем.

Розвинені фреймворки, що включають в себе функції для роботи з даними користувача (статтями, темами, постами, фотографіями і т. і.) називаються фреймворками управління контентом (Content Management Framework - CMF).

Якщо додати до цього елементи інтерфейсу для кінцевого користувача сайту, виходить так звана система управління контентом або CMS (Content Management System). CMS дозволяє отримати сайт з прототипом інтерфейсу і мінімальною функціональністю відразу після установки або після певної настройки, тобто взагалі без програмування.

При цьому більшість CMS надають програмісту як програмний інтерфейс CMF, так і інтерфейс для розширення своєї функціональності.

Для PHP розроблено багато фреймворків і CMS. Найбільш популярними CMS є WordPress, Joomla, Drupal.

Для цих систем розроблено безліч плагінів. Є також самовчителі і документація, вивчивши які, можна написати свій плагін до однієї з цих систем буквально через декілька днів.

Можна зайнятися вивченням і менш популярного фреймворку або CMS, але в цьому випадку шанси знайти готовий приклад коду або плагін для реалізації будь-якої ідеї будуть істотно нижче.

У випадку з Python, з безлічі фреймворків найбільш популярними є Django і CherryPy. Вони дають найбільш повний набір засобів для веб-розробки «з коробки».

Якщо ж використовувати Ruby, то вибір більш очевидний - Rails є основним засобом веб-розробки на цій мові, хоча існують і інші, значно менш популярні фреймворки.

Програмісту початківцю варто враховувати те, яка мова програмування забезпечить кращу підтримку у вигляді форумів, конференцій та документації для початківців - статей, самовчителів.

Резюмуючи сказане:

- PHP - краща мова для створення динамічних веб-сторінок;
- Python - універсальна мова програмування, за допомогою якої можна робити будь-які застосунки в діапазоні від інтернет-сайтів та десктопних застосунків до роботів і системних сервісів;
- Ruby - найбільш високорівнева мова, що дозволяє приділяти менше уваги деталям інтерфейсу і організації зберігання даних, щоб зосередитися на прикладній задачі.

### **Порівняння PHP з Java**

Мова програмування Java є C-подібною мовою, що розроблялась як покращений C++. Технологія Java включає в себе клієнтську та серверну частину, а також доступ до баз даних, тому найбільш коректно проводити порівняння технології Java зі зв'язкою Apache/PHP/MySQL. Технологія Java розроблялась як міжплатформна технологія, що дозволяє створювати веб-застосунки корпоративного масштабу. Головними перевагами цієї технології є міжплатформна переносимість та об'єктно-орієнтована мова, що дозволяє створювати складні та об'ємні програми. До числа недоліків можна віднести повільне виконання, споживання великої кількості пам'яті (плата за міжплатформну переносимість) та складність розробки веб-застосунків в порівнянні з PHP. Тим часом PHP майже не поступається Java в гнучкості і масштабованості створених застосунків. Крім того, використання Java для розробки досить простих додатків навряд чи виправдано.

### **Порівняння PHP з ASP.NET**

ASP (Active Server Pages) – технологія на базі мови сценаріїв від компанії Microsoft. Основна перевага ASP.NET полягає в можливості використання всієї потужності інтерфейсу Windows для розробки веб-застосунків. Це дозволяє розробляти сторінки зі складним інтерфейсом, не піклуючись про збереження,

шифруванні та передачі змінних, що використовуються на динамічних веб-сторінках.

ASP.NET тісно інтегрований в нову технологію NET компанії Microsoft. Головною особливістю розробки веб-застосунків за допомогою ASP.NET є можливість застосовувати будь-яку мову програмування середовища виконання NET. Це може бути як модифікований варіант VisualBasic - Visual Basic.NET, так і нова мова C# або клон Java - J#. Більш того, в межах одного веб-застосунку можуть бути використані кілька мов програмування. Все це дозволяє об'єднувати в рамках одного проекту програмістів, що володіють різними мовами програмування. Код веб-застосунку на ASP.NET компілюється в проміжну мову MSIL (Microsoft Intermediate Language), іноді просто звану IL. Апаратно-незалежний код на мові IL розміщується в кеш веб-серверу для підвищення продуктивності, за аналогією з байт-кодом Java. Але на відміну від технології Java, що забезпечує платформну незалежність, NET забезпечує незалежність від мови розробки. Виконання веб-застосунків ASP.NET на серверах, що працюють не під Windows, практично неможливо. Тому в сенсі переносимості PHP, безумовно, відрізняється від ASP.NET в кращу сторону. Інший 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. Якщо ж написати про помилку в Microsoft і попросити її виправити, це навряд чи вдасться. Ще одним плюсом PHP, особливо важливим для початківців, є те, що його легше застосовувати для побудови веб-застосунків.

Таким чином, при виборі між ASP.NET і PHP слід виходити з поставленого завдання. Якщо в мережі Інтранет великої корпорації більшість серверів і клієнтських машин працюють під управлінням Windows, використання ASP.NET дуже виправдане. В той же час для розробки веб-сайту, що розміщується в Інтернеті, доцільніше використовувати PHP.

Основні переваги PHP - це його простота, велика кількість простих функцій, велика кількість розширень для нього, а також зручність роботи з базами даних. На даний час підтримується переважною більшістю хостинг-провайдерів і є лідером серед мов програмування, що використовуються для створення динамічних веб-сайтів.

MySQL - дуже швидка та надійна система управління базами даних (СУБД). Ця СУБД дозволяє ефективно зберігати, шукати, сортувати та вибирати інформацію. MySQL управляє доступом до даних, дозволяючи працювати з ними одночасно декільком користувачам, забезпечує швидкий доступ до даних та гарантує надання доступу тільки тим користувачам, які мають на це право. Отже, MySQL є багатокористувацькою, багатопоточною СУБД. MySQL портована на всі основні платформи, сумісна з великою кількістю мов програмування в тому числі і з PHP.

### **Аналіз архітектурних шаблонів та фреймворків**

При розробці програмних систем застосовуються різні архітектурні шаблони (патерни). Кожен шаблон має свої завдання і свої недоліки.

Багаторівневий шаблон (Layered pattern). Система поділяється на рівні, які на діаграмі зображуються один над іншим. Кожен рівень може викликати тільки рівень на один нижче за нього. Таким чином розробку кожного рівня можна вести відносно незалежно, що підвищує модифікованість системи. Недоліками цього підходу є ускладнення системи та зниження продуктивності.

Шаблон посередника (Broker pattern). Якщо в системі присутня велика кількість модулів, їх пряма взаємодія один з одним стає занадто складною. Для вирішення проблеми додається посередник (наприклад, шина даних), по якій модулі спілкуються один з одним. Таким чином, підвищується функціональна сумісність модулів системи. Всі недоліки випливають з наявності посередника -

він знижує продуктивність, його недоступність може зробити недоступною всю систему, він може стати об'єктом атак і вузьким місцем системи.

Шаблон Модель-Представлення-Контролер (Model-View-Controller pattern). Оскільки вимоги до інтерфейсу змінюються найчастіше, то виникає потреба часто його модифікувати, при цьому зберігаючи коректну взаємодію з даними (читання, збереження). Для цього в шаблоні Model-View-Controller (MVC) інтерфейс відділений від даних. Це дозволяє змінювати інтерфейси, так само як і створювати їх різні варіанти. У MVC система розділена на:

- модель, що зберігає дані
- представлення, яке відображає частину даних і взаємодіє з користувачем;
- контролер, який є посередником між представленнями і моделлю.

Однак, концепція MVC має і свої недоліки. Зокрема, через ускладнення взаємодії падає швидкість роботи системи.

Клієнт-серверний шаблон (Client-Server pattern). Якщо є обмежене число ресурсів, до яких потрібен обмежений правами доступ великої кількості споживачів, то зручно реалізувати клієнт-серверну архітектуру. Такий підхід підвищує масштабованість і доступність системи. Але при цьому сервер може стати вузьким місцем системи, при його відсутності стає недоступна вся система.

Фреймворк (або «каркас») - це програмна платформа, набір бібліотек та інструментів, що дозволяє спростити і прискорити вирішення типових задач конкретного мови програмування.

Бібліотека - це набір підпрограм з корисними функціями загального призначення. Вона не визначає архітектуру програми, а просто надає засоби, за допомогою яких простіше вирішувати ті чи інші завдання.

Фреймворк ж створює деякі архітектурні обмеження - визначає загальну структуру, її поділ на класи і об'єкти, їх основні функції та методи взаємодії. Також у них різні напрямки викликів: фреймворк викликає код, а бібліотека викликається кодом.

Повертаючись до MVC, можна помітити, що дуже багато фреймворків використовують саме такий підхід - дають досить чітку структуру програми, при

якій відбувається ще й поділ файлів на види, моделі та контролери. Все це в сукупності здатне заощадити багато часу, якщо одного разу витратити його на те, щоб навчитися використовувати фреймворк і модель MVC.

Більшість розробників використовують PHP-фреймворки тому, що використання фреймворків:

- істотно скорочує терміни розробки;
- дозволяє писати добре структурований, добре документований і повторно використовуваний код;
- дозволяє створювати масштабовані додатки, що легко розгортаються;
- приховує від розробника необхідність піклуватися про низькорівневу безпеку сайту;
- стимулює слідувати шаблоном проектування MVC (Модель-Представлення-Контролер), що дозволяє розділити логіку програми і представлення даних;
- сприяє застосуванню сучасних методів програмування, в першу чергу об'єктно-орієнтованого.

### **Архітектурний шаблон MVC**

Концепція Model-View-Controller дозволяє розділити дані, представлення та обробку дій користувача на три окремих компоненти (рис.1.7):

- модель (Model). Модель надає дані і методи роботи з цими даними, реагує на запити, змінюючи свій стан. Не містить інформації, як ці знання можна візуалізувати;
- представлення, вид (View). Відповідає за відображення інформації (візуалізацію). Часто як представлення виступає форма (вікно) з графічними елементами;
- контролер (Controller). Забезпечує зв'язок між користувачем і системою - контролює введення даних користувачем та використовує модель і представлення для реалізації необхідної реакції.

Важливо відзначити, що як представлення, так і контролер залежать від моделі. Однак модель не залежить ні від представлення, ні від контролера. Тим

самим досягається мета такого поділу - він дозволяє будувати модель незалежно від візуального представлення, а також створювати кілька різних представлень для однієї моделі.

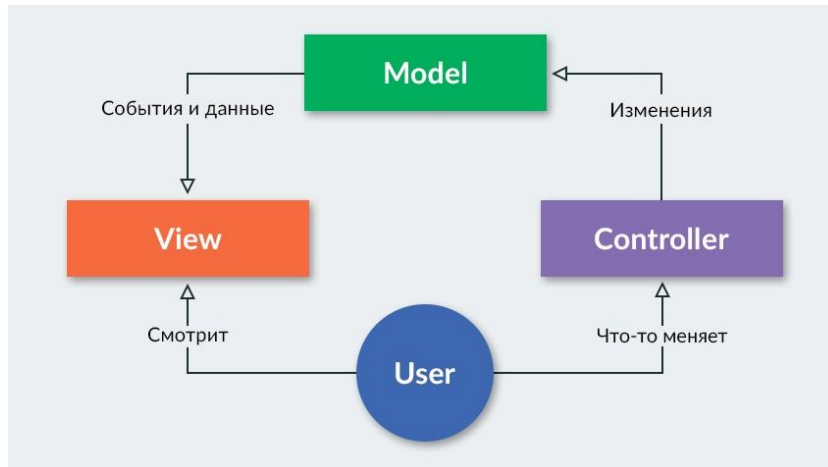


Рисунок 1.7 - Архітектурний шаблон MVC

Основна мета застосування цієї концепції полягає в відокремленні бізнес-логіки (моделі) від її візуалізації (представлення, виду). За рахунок такого поділу підвищується можливість повторного використання. Найбільш корисне застосування даної концепції в тих випадках, коли користувач повинен бачити ті ж самі дані одночасно в різних контекстах або з різних точок зору. Зокрема, виконуються наступні завдання:

- до однієї моделі можна приєднати кілька видів, при цьому не зачіпаючи реалізацію моделі. Наприклад, ті ж самі дані можуть бути одночасно представлені у вигляді електронної таблиці, гістограми та кругової діаграми;

- не зачіпаючи реалізацію видів, можна змінити реакції на дії користувача (натискання мишею на кнопки, введення даних), для цього досить використовувати інший контролер;

- ряд розробників спеціалізується тільки в одній з областей: або розробляють графічний інтерфейс, або розробляють бізнес-логіку. Тому можливо добитися того, що програмісти, які займаються розробкою бізнес-логіки (моделі), взагалі не будуть інформовані про те, яке представлення буде використовуватись.



## Zend Framework

Zend Framework - це PHP-фреймворк, створений і підтримуваний компанією Zend, співробітники якої є безпосередніми авторами мови PHP. Тому він наслідує традиції і дух PHP - базується на простоті, об'єктно-орієнтованих принципах, дружній ліцензії і коді, який ретельно тестується.

Основні особливості:

- всі компоненти повністю орієнтовані на PHP 7;
- вбудований генератор коду;
- архітектура «використовуй тільки те що необхідно» з мінімальними залежностями компонентів;
- використовує легко розширюваний шаблон проектування MVC, що за замовченням підтримує макети і PHP-шаблони;
- підтримує безліч різних баз даних включаючи MariaDB, MySQL, Oracle Database, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite та Informix;
- формування, відправлення та отримання поштових повідомлень за протоколами mbox, Maildir, POP3 і IMAP4;
- гнучка система кешування з підтримкою різних типів - в пам'яті або в файлової системі.

Але має і недоліки:

- дещо завеликий;
- занадто багат шаровий;
- дуже повільний без кешування;
- доволі складна архітектура, складний в засвоєнні без глибокого розуміння шаблонів проектування;
- відносно невелике співтовариство;
- має багато дрібних недоліків;
- дуже повільно розвивається;
- не має ORM (Object Role Model) для зв'язку з базами даних.

## **Фреймворк Symfony**

Symfony – це PHP фреймворк, що має велику бібліотеку класів. Архітектура має корисні компоненти та відмінні інструменти призначені для створення складних веб-застосунків. Symfony - вільний фреймворк, написаний на PHP, що використовує патерн Model-View-Controller (MVC). Symfony пропонує швидку розробку та управління веб-застосунками, дозволяє легко вирішувати рутинні завдання веб-програміста.

Його особливості:

- підтримує багато баз даних (MySQL, PostgreSQL, SQLite або будь-яка інша PDO-сумісна СУБД);
- вбудовані класи для роботи з email;
- гнучка система шаблонів;
- вбудований кодогенератор;
- дуже гнучкий;
- висока продуктивність.

Також є недоліки:

- високий поріг входу, треба знати ООП, розбиратися в шаблонах проектування і мати практичний досвід роботи з проектами на інших фреймворках;
- підходить тільки для великих проектів;
- недостатня українсько та російськомовної документації;
- дуже ресурсномісткий.

## **Фреймворк CodeIgniter**

CodeIgniter - популярний MVC фреймворк з відкритим вихідним кодом, написаний на мові програмування PHP, для розробки повноцінних веб-систем і застосунків. Заснований на моделі MVC, що дозволяє відокремити логіку застосунку від представлення.

Модель - клас моделі, набір функцій для роботи з даними, який допоможе оновити, отримати, додати інформацію. Представлення - шаблон для відображення даних, зазвичай простий HTML зі вставками PHP. Контролер -

виступає в ролі посередника між моделлю, представленням та будь-якими іншими ресурсами. У CodeIgniter моделі не є обов'язковими, можна використовувати тільки контролер та представлення.

Основні особливості:

- CodeIgniter відрізняє простота;
- хороша документація;
- розвинуте співтовариство;
- надає свободу програмісту, не створюючи будь-яких структурних обмежень;
- вже містить в собі більшість необхідних бібліотек для роботи з файлами, відправки електронних листів, перевірки форм, підтримки сесій, роботу із зображеннями та багато інших;
- легко розширювана система завдяки підключенню власних бібліотек та плагінів;
- працює практично на будь-якому хостинговому плані, який має підтримку PHP версії 5.1 і вище;
- вважається одним з найшвидших та не вимогливих до ресурсів фреймворків;
- підтримка баз даних MySQL, PostgreSQL, MSSQL, SQLite, Oracle.

Його недоліки:

- підтримка PHP4 тягне за собою зайвий код;
- не має вбудованої ORM;
- не має вбудованої системи поділу прав;
- повільно розвивається.

### **Фреймворк Laravel**

Laravel - це безкоштовний PHP фреймворк з відкритим вихідним кодом, створений Тейлором Отвеллом для розробки веб-додатків за архітектурним шаблоном MVC.

Можна сказати, що на появу Laravel вплинули інші PHP фреймворки. Він був створений як альтернатива фреймворку CodeIgniter, в якому було недостатньо

корисних функцій для розробки веб-застосунків. В якості основи Laravel виступають компоненти іншого фреймворка - Symfony.

Laravel дуже популярний серед західних розробників веб-застосунків.

Його плюси:

- досить непогана і зрозуміла документація;
- навколо фреймворка створена потужна екосистема. Різні курси, конференції, навчальні матеріали дозволяють зібрати навколо фреймворка велику кількість розробників і спонсорів, які зацікавлені в розвитку інструменту і приймають в цьому участь;
- гнучка система маршрутизації, що дозволяє скласти найрізноманітніші маршрути перевірки веб-застосунків;
- багато синтаксичного цукру (syntactic sugar - це синтаксичні можливості, застосування яких не впливає на поведінку програми, але робить використання мови більш зручним для людини). Синтаксис API фреймворка досить простий і зрозумілий. Тут немає довгих і складних конструкцій, а тільки короткі і продумані назви функцій;
- містить зручний механізм обробки помилок і виключень;
- включає в себе вбудовані механізми аутентифікації та авторизації користувачів, які можна переналаштувати під свої потреби;
- надає з коробки механізми для кешування веб-застосунків за допомогою Memcached і Redis. Крім цього є зручні функції для використання простого файлового кешування даних.

Але є й недоліки:

- без знання англійської мови або з його слабким знанням до мінусів фреймворка можна віднести досить невелику кількість статей, прикладів коду, перекладів офіційної документації;
- синтаксичний цукор в Laravel як плюс, так може бути і мінусом. Дуже легко звикнути до нього і забути, як пишуться чисті запити і функції;
- порушення зворотної сумісності між версіями фреймворка;
- не дуже логічне розташування каталогів і файлів.

## **2 ПРОЕКТУВАННЯ СИСТЕМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ**

### **Розробка структурної схеми системи**

Вхідними даними для системи є інформація, що вводиться адміністратором, менеджерами або користувачами в спеціальні форми для введення.

Інформація, що вводиться адміністратором бази даних:

- логін та пароль;
- дані про товари.

Інформація, що вводиться менеджерами:

- логін та пароль.

Інформація, що вводиться користувачем:

- а) критерії пошуку;
- б) відгуки, питання, коментарі у формі зворотного зв'язку;
- в) логін та пароль для входу в особистий кабінет;
- г) дані для оформлення замовлення:
  - 1) адреса доставки;
  - 2) контактний телефон для зв'язку;
  - 3) кількість обраного товару;
  - 4) примітки до замовлення.

Вихідними даними є інформація та результати запитів до бази даних, які відображаються на екрані монітора у вигляді HTML-сторінок:

- інформація про товар;
- результати пошуку;
- статті про товари;
- інформація про способи доставки та оплати.

### **Архітектура електронного магазину**

Архітектура веб-сайту електронного магазину обумовлюється необхідністю систематизації інформації та навігації по неї з метою допомоги

користувачам успішно знаходити потрібні їм товари та інформацію. Продумана архітектура веб-сайту гарантує, що користувач витратить на це менше часу.

Розробка архітектури має вестися з врахуванням найбільш важливої інформації щодо просування товарів та послуг в мережі Інтернет. При створенні структури нового вебсайту, або при оптимізації структури вже існуючого, необхідно зосередити увагу відвідувачів саме на цієї інформації та управляти відвідуваністю потенційними покупцями саме найбільш важливих розділів сайту відповідно до позиціонування на ринку та товарів і послуг, що просуваються.

Правильний розподіл пріоритетів між розділами і сторінками сайту, робить їх основними точками входу на сайт, що дозволить потенційному покупцю швидко знайти необхідну інформацію про потрібні товари і послуги та підвищить успішність бізнесу в Інтернет.

Архітектура інтернет-магазину проста та інтуїтивно зрозуміла. Вона складається з клієнтської частини, програмної частини і адміністрування (рис.2.1).



Рисунок 2.1 - Архітектура інтернет-магазину

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

Як операційна частина розглядається середовище розробки інтернет-магазину.

Серверна частина містить компоненти інтернет-магазину, що розміщені на сайті провайдера з підтримкою технології, що використані при створенні інтернет-магазину.

### **Програмні засоби для розробки та роботи системи електронної комерції**

Спроектвана система електронної комерції розроблювалась за допомогою таких обраних технологій розробки веб-застосунків, як HTML, CSS, JavaScript, JQuery, PHP, CodeIgniter, MySQL. Вона створена з використанням програмних шаблонів MVC, що дозволяє як збільшити ефективність функціонування системи, так і виключити помилки написання програмного коду.

HTML (Hyper Text Markup Language) - мова розмітки гіпертексту, служить для створення гіпертекстових веб-сторінок, розташованих в глобальній мережі Internet або у внутрішній мережі підприємства (intranet).

CSS (Cascading Style Sheets) - мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки. CSS використовується для завдання кольорів, шрифтів, розташування окремих блоків та інших аспектів представлення зовнішнього вигляду веб-сторінок. Метою CSS є відокремлення опису логічної структури веб-сторінки, що здійснюється за допомогою HTML) від опису зовнішнього вигляду цієї ж веб-сторінки (здійснюється за допомогою мови CSS).

JavaScript використовується для додавання інтерактивності веб-сторінкам системи та динамічної типізації. JavaScript застосовується в клієнтській частині веб-застосунків, що мають розподілену між сервером і клієнтом логіку. Обмін інформацією у веб-застосунку відбувається по мережі. Одною з переваг такого підходу є те, що клієнти не залежать від конкретної операційної системи користувача, тому веб-застосунок є кросплатформним сервісом.

jQuery - бібліотека JavaScript, яка фокусується на взаємодії JavaScript з HTML. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого

елементу об'єктної моделі документу DOM, звертатися до атрибутів та вмісту елементів DOM, маніпулювати ними. Також бібліотека jQuery надає зручний API для роботи з технологією AJAX.

CodeIgniter - MVC фреймворк з відкритим вихідним кодом, написаний на мові програмування PHP, для розробки повноцінних веб-систем та веб-застосунків. Це готовий каркас для програми, яка будується на його основі. Завдяки тому, що фреймворк заснований на шаблоні MVC, стає можливим відокремити логіку застосунку від представлення.

MySQL - вільна реляційна система управління базами даних. Розробка та підтримка MySQL здійснює корпорація Oracle, яка отримала права на торговельну марку разом з поглиненої Sun Microsystems, яка раніше придбала шведську компанію MySQL AB. Продукт поширюється як під GNU General Public License, так і під власною комерційною ліцензією. Крім цього, розробники створюють функціональність за замовленням ліцензійних користувачів.

MySQL є рішенням для малих і середніх додатків. Входить до складу серверів WAMP, AppServ, LAMP і в портативні збірки серверів Denver, XAMP, VertrigoServ. Зазвичай MySQL використовується як сервер, до якого звертаються локальні або видалені клієнти, проте в дистрибутив входить бібліотека внутрішнього сервера, що дозволяє включати MySQL в автономні програми.

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Більш того, СУБД MySQL поставляється із спеціальним типом таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі і GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць.

Спільнотою розробників MySQL створені різні відгалуження коду, такі як Drizzle (англ.), OurDelta, Percona Server і MariaDB.



## Пошукова оптимізація контенту

SEO-оптимізація [3] – сфера, що динамічна розвивається та в якій зміни відбуваються дуже швидко. Оперативна реагуючи та підлаштовуючись, можна зайняти суттєву долю ринку. На початку появи пошукової оптимізації в хід йшли всі засоби, щоб сайт виявився як можна вище за максимально короткий термін. Тоді, як з часом тактика все більше і більше ускладнюється. Це не тільки ускладнює роботу оптимізаторів, а й допомагає усунути нечесну конкуренцію з тими, хто використовує так звані «чорні» методи. Не кажучи про те, що пошукова видача стає якіснішою і релевантною.

«Чорні» методи пошукової оптимізації застосовувалися, в основному, на початковому етапі розвитку SEO-оптимізації та просування сайтів. Тоді ще алгоритми пошукових систем не були до кінця вивчені і мало хто думав про стабільний та довгостроковий ефект. Головною метою було просування сайту в топ за будь-яку ціну і в максимально короткий термін. В основі «чорного» SEO - всілякі види пошукового спаму і накрутки посилального оточення за рахунок безкоштовних каталогів.

На відміну від «чорних» методів, «сірі» офіційно не заборонені пошуковими системами і досить часто застосовуються до цих пір. Однак і в їх основі лежить спроба впливати на пошуковики з метою поліпшення позицій сайту у видачі.

Поступово «сірі» методи присікаються за допомогою зміни алгоритмів і втрачають свою ефективність. Деякі з них і зовсім забороняються, якщо виявляються занадто явні зловживання. Наприклад мало місце виключення з результатів пошуку багатьох сайтів, що просувалися великими SEO-компаніями, причиною чого стали виявлені накрутки за допомогою нагону трафіку користувачів через соціальні мережі. Тому найбільш авторитетні SEO-компанії і ресурси все частіше вдаються виключно до «білих» методів.

«Білі» методи пошукової оптимізації - офіційно дозволені і допустимі пошуковими системами. Вони полягають, перш за все, в грамотній роботі над сайтом, його зручністю і інформативністю для користувача, з дотриманням рекомендацій пошукових систем. Такі методи часто використовуються в рамках комплексного інтернет-маркетингу і виключають штучне вплив на позиції сайту в пошуковій видачі.

Оптимізація сайту включає в себе комплекс заходів щодо поліпшення якості контенту, навігації, вмісту HTML-коду. Зокрема, складається грамотне семантичне ядро, в повній відповідності з тематикою сайту. Використовується унікальний і змістовний контент, який повністю відповідає тематиці і не перенасичений ключовими запитами. Проводиться внутрішня перелінковка сторінок (проставляються посилання на розділи), підбираються якісні домени і піддомени, URL внутрішніх сторінок (що містять основні ключові запити, написані транслітом). Відстежується неякісні посилання і некоректні відповіді сервера на звернення до розділу (в разі помилки запиту, падіння хостинг-сервера, видалення сторінки та ін.). Особлива увага приділяється складанню файлу robots.txt, призначеного для повідомлення інформації про сайт пошуковим роботам, а також заповненню META-тегів в HTML: Title, h1, Description, Keywords.

Просування сайту – це розміщення інформації та посилань на сайт на тематичних ресурсах, проведення рекламних кампаній, публікація якісних статей і інші способи природного нарощування посилального оточення.

Такі методи дають результат не так швидко, але й ефект від них буде стабільним і тривалим, без ризику будь-яких санкцій з боку пошукових систем. Крім того, «біле» SEO забезпечить сайту унікальний і інформативний контент, зробить його більш привабливим для відвідувачів. Що, без сумніву, позначиться на конверсії.

Основи пошукової оптимізації - ключові слова. Оскільки користувачі пошукових систем знаходять потрібний сайт, вводячи в рядок пошуку потрібне слово або словосполучення, і пошукові системи, виконуючи замовлення

користувача, приймаються за пошук потрібних слів і пропозицій в проіндексованих ними сайтах. Чим більше текстової контент сайту, на думку пошукової системи відповідає запиту, тим вище в результатах пошуку система розмістить посилання на ресурс.

Пошукова оптимізація нового сайту починається ще при його створенні. Розробка системи управління, дизайну та інтерфейсу користувача, так само як і написання текстового контенту, повинні підкорятися строгим правилам SEO, тільки тоді сайт вважається оптимізованим для пошукових систем і конкурентоспроможним на ринку.

Робота над змістом сайту і приведення його у стан, найбільш відповідне вимогам пошукових систем називається впливом на внутрішні фактори.

Біла оптимізація - пошукова оптимізація (підстроювання коду, тексту та інших параметрів сайту під алгоритми пошукових систем з метою підняття його позицій у видачі), при якій не застосовуються заборонені і недобросовісні методи просування.

Біла оптимізація включає в себе сукупність ряду робіт:

- розробка семантичного ядра (підбір ключових слів, за допомогою яких буде просуватися сайт). Важливо правильно скласти семантичне ядро тому, що ключові слова - це ті запити, за якими користувач буде знаходити сайт в Інтернет;
- розробка структури сайту і робота з його юзабіліті;
- технічний аудит сайту;
- оптимізація HTML-коду сайту (додавання в мета-теги ключових слів, описи);
- створення нових веб-сторінок для смислового покриття тематики сайту;
- оптимізація текстів сайту;
- складання файлу robots.txt, що робить сторінки сайту недоступними для пошукових роботів.

Біла оптимізація вимагає великих трудовитрат і результат можна побачити не скоро, але зазначені вище заходи створюють базу для якісного просування сайту в подальшому.

Сучасні пошукові системи активно впроваджують нові алгоритми, що суттєво ускладнюють використання заборонених методів оптимізації, і тому біла оптимізація на сьогоднішній день набуває все більший сенс, хоча і є не найпростішим методом.

### **Розробка семантичного ядра системи**

Для проєктованої системи електронної комерції було відібрано 15 684 запитів за допомогою наступних програмних засобів:

- бази Пастухова;
- підказки пошукових систем;
- wordstat;
- solomono – вибірковий перегляд анкор-листів конкурентів;
- Key Collector.

Все було завантажено в Key Collector і через стоп-листи було прибрано багато «сміття». Потім всі фрази були пробиті через Key Collector на частотності wordstat. Були відсіяні фрази, з частотністю "WS" = 0. На виході вийшло 3000 + запитів, які все перебиралися вручну.

Незважаючи на велику кількість запитів – отримана структура семантичного ядра достатньо проста.

Сторінки категорій:

- ноутбуки;
- ультрабуки;
- аксесуари для ноутбуків;
- планшеты;
- електронные книги.

Групи тегів:

а) тип ноутбуку:

- 1) нетбуки, ультрабуки;
- 2) для бизнеса, мультимедийные, геймерские;
- 3) тонкие, легкие, трансформеры;

б) для кого:

1) бизнес;

2) дом, работа;

в) локація, розташування:

1) магазин, інтернет-магазин, сайт;

2) где, запорожье;

г) якісні характеристики:

1) экран, дисплей;

2) разрешение;

3) процессор;

4) видео карта;

5) память, hdd;

6) масса, цвет;

г) известные, популярные, знаменитые;

д) продаж:

1) продажа, продам, продавать;

2) куплю, покупать, покупка, опт, оптом;

е) заказ, заказать;

є) вартість:

1) цена, цены, стоит, стоимость, ценами;

2) дорогие, дорого, дорогих;

3) недорогие, недорого, дешевые, дешево, скидка, скидки;

ж) країна виробник:

1) производитель;

2) китайские, китай, китая;

з) марка, бренд, фирма;

и) каталог;

і) інформація: все, всё;

ї) ремонт:

1) ноутбука, ноутбуков;

2) компьютеров, компьютера;

3) електронной техники.

### **Оптимізація сторінок**

Мета-тег Description надає пошуковій системі коротке резюме, чому саме присвячена ця сторінка. Заголовок сторінки може бути складений з кількох слів, Description може містити від одного речення до одного короткого абзацу.

Вміст цього тегу не повинен повторювати заголовок. Цей тег розташовується в кодї сторінки, як і заголовок всередині тега Head. Мета-тег Description важливий тим, що пошукачі можуть використовувати його в якості короткого резюме вашої сторінки в результатах пошукової видачі.

Тег Title є найважливішою частиною сторінки будь-якого сайту, тому nht,f приділити велику увагу правильності його складання:

- тайтл відображає суть сторінки;
- тайтл включає в себе бренд сайту;
- використання різних тайтлов на різних станицях;
- тайтл містить в собі ключові слова;
- не повинно бути стоп-слів в тайтл;
- кожне слово не повторюється більше двох разів;
- заголовки сторінок оптимальної довжини;
- довжина title 8 - 12 слів або 45 – 140 символів.

Для пошукових систем має велике значення наявність посилань на сайт з інших, авторитетних і не дуже, ресурсів. Дані посилання називають зовнішніми посиланнями. Від посилальної популярності сайту безпосередньо залежить його положення у видачі сайтів при пошуку в пошуковій системі по певних запитах.

Велика увага приділяється тексту, який знаходиться на головній сторінці сайту. Текст складається спеціальним чином, і оптимізується під ті слова, за якими сайт повинен розкручуватися і знаходиться в пошукових системах, як показано в таблицях 2.1 та 2.2. Анкор - це текст посилання, який читач бачить на сторінці. Вони як би додаються до тексту сторінки, на яку веде посилання, і індексуються пошуковими системами. Сукупність усіх прив'язаних до сторінці анкорів називається анкор-листом.

Таблиця 2.1 - Аналіз шаблону типу сторінки головна

Анкор-лист vs. Семантичне ядро	Внутрішній анкор лист переспамлен анкором «Каталог ноубуків» (який до того ж не цільовий), потрібне розширення / розбавлення цільовими анкорами з семантичного ядра для даної сторінки: <ul style="list-style-type: none"> <li>– ноутбук</li> <li>– купити ноутбук</li> <li>– ноутбук asus</li> <li>– ноутбук lenovo</li> <li>– ноутбук hp</li> <li>– ноутбук acer</li> <li>– ноутбуки ціни</li> <li>– ремонт ноутбуків</li> <li>– ультрабук купити</li> </ul>
Вихідні посилання	Закрити в seohide посилання на службові сторінки: замовлення та оплата, доставка, гарантія, потрібна порада, контакти, карта сайту.
Валідація сторінки	Не проходить валідацію W3C, усунути критичні помилки верстки.
Граматика	Граматичних помилок немає
Входження ключового слова в теги	В title – так, в keywords – так, в description – так, в H – так, в Strong – так, в B - так
Обсяг тексту	3200 символів
Унікальність	Унікальна
Зовнішні посилання	немає
Вхідні посилання	3340

За статистикою користувачі не читають весь текст цілком, особливо якщо він не особливо привабливий для користувача. Для того, щоб користувачі не ігнорували сайт, слід уникати безперервного тексту, великих і довгих речень, розбиваючи їх на більш короткі, відмовитися від використання складних слів та словосполучень.

Анкор лист і семантичне ядро. Ядро системи ділилося по сторінках брендів, і за тегами. Маємо семантичне ядро з 3х типів: семантичне ядро головної сторінки, семантичне ядро по брендам, семантичне ядро за тегами.

Таблиця 2.2 - Аналіз шаблону типу сторінки картка товару

Анкор-лист vs. Семантичне ядро	Тільки один анкор "купити ноутбук Dell 3584". Є сенс розширити анкор-лист. Чи не відомо реальної назви моделі. На даний момент розширювати нічим.
Вихідні посилання	Закрити в seohide посилання на службові сторінки: замовлення та оплата, доставка, гарантія, потрібна порада, контакти, карта сайту.
Валідація сторінки	Не проходить валідацію W3C, усунути критичні помилки верстки.
Входження ключового слова в теги	В title – так, в keywords – так, в description – ні, в H – так, в Strong – так, в B - ні
Грамматика	Грамматичних помилок немає
Обсяг тексту	300-400
Унікальність	Унікальна
Зовнішні посилання	немає
Вхідні посилання	1 - при даній схемі перелинковки картки товару не отримують практично ніякої ваги. це також може викликати проблеми з індексацією.

У підсумку вийшло добре збалансоване семантичне ядро - на кожен сторінку в середньому від 5 до 10 запитів, максимум 30 запитів (по двох сторінкам брендів), на головну припадає 42 запиту, що нормально для головної. Також проведено аналіз програмою-аналізатором. Результати програми показано на рисунку 2.2

Состояние	
Статус проекта:	Анализ, завершён.
Дата запуска:	07.04.2021 23:59
Объём страниц проекта:	146656,988кб. (143,22мб.) средний:14,095кб.

Загрузка	
Количество страниц:	3471 при лимите в 10000 (не включая ошибки).
Количество пройденных ссылок:	290740 Углубление до: 5 уровня доступности от стартовой.
Время соединения с сервером:	Среднее: 0,335 сек.
Время загрузки страницы:	Среднее: 1,6 сек. Средняя скорость:30,709kbps
Ответы сервера:	2** = 3467 3** = 0 4** = 0 5** = 0 Без ответа: 4 Другие: 0
Блокировано:	Инструкции robots.txt и stop слова url:16

Рисунок 2.2 – Аналіз семантичного ядра



Пагінація (Pagination) - це порядкова нумерація сторінок, яка в основному розміщується вгорі або внизу сторінок сайту.

Більшість сторінок розташовані в трьох кліках від головної (четвертий рівень вкладеності), в той же час сторінок другого рівня вкладеності дуже мало, але це зумовлено структурою систем електронної комерції. Другий рівень вкладеності - бренди товарів, третій рівень вкладеності (2 кліка від головної сторінки) - товари першої сторінки бренду, четвертий рівень вкладеності - товари видимої частини пагінатора, п'ятий рівень вкладеності - сторінки невидимій частині пагінатора.

На перший погляд може здатися, що на сайті занадто мало контенту. Але це система електронної комерції і основну частину складають картки товарів: характеристика і фото. Решта - категорії товарів (від 1000 до 1500 знаків) і статті (2000+ знаків), як показано на рисунку 2.3.

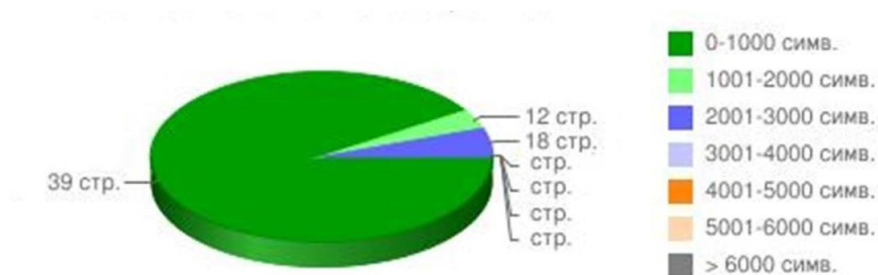


Рисунок 2.3 – Кількість контенту на сторінках

Мікродані - це спосіб семантично розмічати відомості про картку товару (назва, фото, ціни, опис, рейтинг) на веб-сторінках пошукової системи, використовуючи стандартні елементи мови HTML. Користувач може сприймати розмічену сторінку як звичайну веб-сторінку (через браузер), а програмісти пошукової системи здатні витягти з такої сторінки структуровану інформацію завдяки атрибутам  `itemprop`, що використовуються в HTML-розмітці:

- `Itemprop= "name"` – назва товару;
- `Itemprop= "image"` – фото товару;
- `Itemprop= "rating"` – рейтинг товару;

- Itemprop= "price" – ціна товару;
- Itemprop= "description" – опис товару.

Наприклад:

```
<h1 itemprop="name">Ноутбук DELL Inspiron G5 5500 купити</h1>
```

## **Розробка алгоритму роботи системи**

Відвідувач набирає в рядку пошуку URL магазину і потрапляє на головну сторінку сайту. На головній сторінці знаходиться головне меню, за допомогою якого відбувається навігація по сайту. До складу головного меню входять пункти:

- товари;
- зворотній зв'язок;
- пошук;
- про магазин;
- послуги;
- статті;
- вихід.

Вибравши пункт «Товари», користувачу надається можливість обрати товар за такими критеріями:

а) за призначенням:

- 1) ноутбуки: нетбуки, ультрабуки, для бізнесу;
- 2) електронні книги;
- 3) планшети: бюджетні, продуктивні, трансформери;
- 4) аксесуари: для ноутбуків, планшетів, електронних книг;
- 5) бренди;

б) за фірмою виробником.

Після вибору певного виду товару, відвідувач може перейти до його опису або оформити замовлення на обраний товар:

- «Описание» - наводиться опис товарів;
- «Прайс-лист» - відвідувач має змогу ознайомитись з вартістю товарів інтернет-магазину;
- «Заказ» - обравши даний пункт, користувач може оформити замовлення на обраний товар, заповнивши форму. У формі потрібно вказати адресу доставки, контактний телефон, кількість обраного товару;
- «Возврат на главную страницу» - дозволяє повернутися на головну сторінку сайту.

Пункти «Описание» і «Заказ» можна обрати тільки після вибору товару.

Повернутися на головну сторінку можна вибравши пункт «Возврат на главную страницу».

Обравши пункт «Обратная связь» користувач може відправити повідомлення, побажання чи задати питання менеджеру магазину. Для цього необхідно заповнити форму - обрати пункт «Заполнение формы». Після заповнення форми, її потрібно відправити - пункт «Отправка заполненной формы».

Повернутися на головну сторінку можна обравши пункт «Возврат на главную страницу».

«Поиск» дозволяє відвідувачу здійснити пошук товару. Для початку пошуку необхідно ввести критерії пошуку (будь-які, на вибір) - пункт «Ввод критериев поиска»:

- назва товару;
- модель товару;
- виробник товару.

Якщо пошук здійснювати не потрібно, на головну сторінку можна повернутися обравши пункт «Возврат на главную страницу».

Після введення критеріїв пошуку з'являються результати пошуку - «Просмотр результатов поиска». Якщо необхідно здійснити новий пошук, досить вибрати пункт «Возврат на страницу поиска». Щоб повернутися на головну сторінку, необхідно вибрати «Возврат на главную страницу».

Пункт «Услуги» дозволяє відвідувачу дізнатися про послуги, що надає компанія, а саме: ремонт ноутбуків та іншої електронної техніки, налаштування, прошивка, забезпечення ПЗ.

Пункт «Про магазин» надає відвідувачу інформацію про магазин. Контакти магазину, адресу та телефон можна дізнатися обравши пункт «Просмотр контактов». Переглянути інформацію про доставку товару можна обравши пункт «Просмотр информации о доставке». Інформацію про магазин в цілому можна отримати обравши пункт «Информация о магазине». Пункт «Возврат на главную страницу» повертає відвідувача на головну сторінку сайту.

При виборі пункту «Статьи», відвідувачу пропонується обрати тему статті із запропонованого списку. Відвідувач може повернутися на сторінку списку статей для вибору нової статті - пункт «Возврат к выбору статей» або повернутися на головну сторінку сайту - пункт «Возврат на главную страницу сайта».

До вибору статті, користувач може повернутися на головну сторінку. Для цього необхідно вибрати пункт «Возврат на главную страницу сайта».

Для входу в якості адміністратора або менеджера необхідно ввести ім'я та пароль в поля для входу - пункт «Вход». Після входу в якості адміністратора, можна редагувати каталог продукції та управляти роботою сайту.

Якщо зайти в якості менеджера, можна переглядати замовлення, що надійшли, повідомлення від користувачів (побажання в роботі інтернет-магазину або питання). Також менеджер має можливість переглядати статистику покупок, вносити зміни до інформації про магазин, до статей про товари та створювати або редагувати модулі на сайті.

Адміністрування містить інструменти управління системою електронної комерції і включає в себе як загальні налаштування магазину, так і спеціальні налаштування.

В адмініструванні містяться основні налаштування системи електронної комерції, а саме:

- загальні налаштування магазину: назва магазину, адреса, телефон, e-mail магазину і т. і .;

- налаштування форми оформлення замовлення та реєстрації клієнта в системі електронної комерції;
  - загальні налаштування доставки;
  - налаштування складу;
  - налаштування логів (файлів реєстрації), файлів, куди записується службова інформація;
  - налаштування формату виводу товару в системі електронної комерції.
- Можна налаштувати формат виводу товару за своїм бажанням;
- різні налаштування каталогу, тобто додавання, видалення, редагування товару і категорій товарів, робота з постачальниками, Excel імпорт/експорт товарів і т. і.;
  - налаштування різних модулів доставки, оплати, модулі знижок і т. і. Встановлення нових модулів, видалення існуючих модулів, налаштування способів оплати та доставки замовлень інтернет-магазину;
  - управління оформленими замовленнями, управління зареєстрованими клієнтами;
  - додавання, видалення змін курсів валют;
  - статистичні звіти про роботу інтернет-магазину;
  - важливі інструменти для роботи інтернет-магазину. Такі як резервне копіювання бази даних, незавершені замовлення, пошукові запити і т. і.

### **Розробка бази даних системи**

Вибір системи управління базою даних (СУБД) для системи електронної комерції загалом являє собою складну багатопараметричну задачу і є одним з важливих етапів при розробці застосунків баз даних.

В загальному випадку перелік вимог до СУБД, що використовуються при аналізі тієї чи іншої інформаційної системи, може змінюватися в залежності від поставлених цілей, але можна виділити кілька груп критеріїв:

- моделювання даних;
- особливості архітектури та функціональні можливості;
- контроль роботи системи;
- особливості розробки застосунків;
- продуктивність;
- надійність;
- вимоги до робочого середовища.

Існує багато моделей даних найпоширеніші з яких ієрархічна, мережна, реляційна, об'єктно-реляційна та об'єктна. Питання про використання тієї чи іншої моделі має вирішуватися на початковому етапі проектування інформаційної системи.

Проектування реляційної БД включає наступні три етапи.

Перший етап (інфологічне проектування) - збір, аналіз і редагування вимог до даних в предметній області. Для цього проводять інформаційне дослідження предметної області, вивчення її інформаційної структури, виявлення всіх фрагментів, кожен з яких характеризується інформаційними об'єктами і зв'язками між ними, процесами над інформаційними об'єктами, моделювання та інтеграція всіх представлень. Результатом цього етапу є інфологічна модель БД;

Предметна область: магазин комп'ютерної техніки.

Сутності:

- покупець (id\_рок, ФІБ, Нік, email);
- категорії покупців (cat\_id, Ім'я\_Категорії);
- категорія товару (id\_cat\_tov, Назва\_Категорії);
- замовлення (id\_zak, Номер\_Замовлення);
- товар (id\_tov, Опис, Ціна).

В результаті отримуємо логічну модель (рис.2.4).

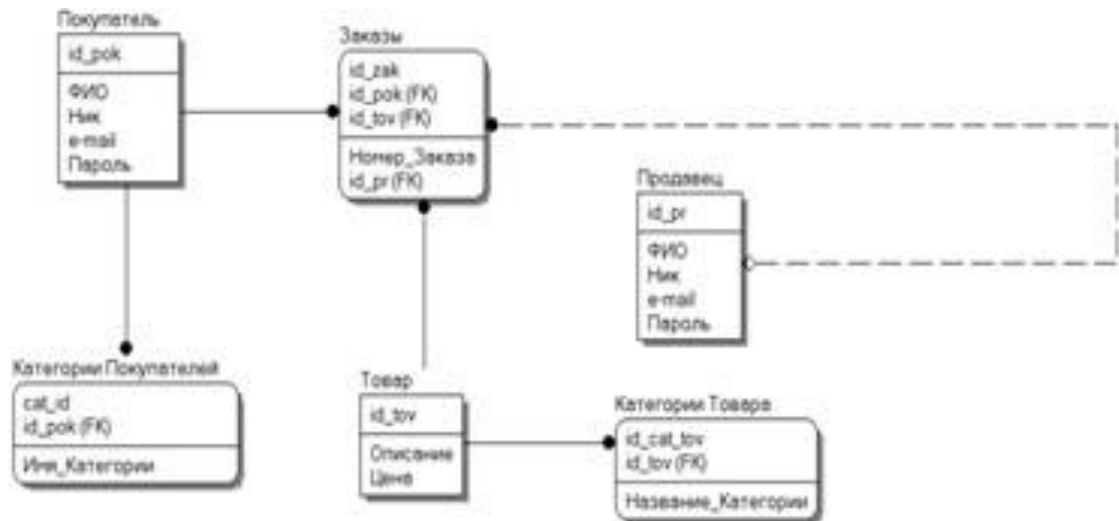


Рисунок 2.4 - Логічна модель БД

Другий етап (логічне проектування) - перетворення інфологічної моделі БД в структури даних. Результатом цього етапу є логічна модель БД представлена на рисунку 2.4.

Визначимо первинні та зовнішні ключі відносин:

Покупець:

- первинний - id\_pok.

Категорії покупців:

- первинний - cat\_id;
- вторинний - id\_pok.

Категорія товару:

- первинний - id\_cat\_tov;
- вторинний - id\_tov.

Замовлення:

- первинний - id\_zak;
- вторинний - id\_pok, id\_pr, id\_tov.

Товар:

- первинний - id\_tov.

Визначаємо функціональні залежності для кожної відносини:

Покупець:

- id\_рок – ФІБ;
- id\_рок – Нік;
- id\_рок – email.

Категорії покупців:

- cat\_id – Ім'я\_Категорії.

Категорія товару:

- id\_cat\_tov - Назва\_Категорії.

Замовлення:

- id\_zak - Номер\_Замовлення.

Товар:

- id\_tov – Опис;
- id\_tov – Ціна.

Третій етап (фізичне проектування) - визначення особливостей зберігання даних, методів доступу і т. і.

Тому, що основний каркас системи взято з CMS Opencart, то таблиці в базі даних створюються автоматично.

Нижче наведені приклади таблиць з логічної моделі, створених CMS.

У базі даних немає конкретного поділу покупців, вони знаходяться в одній таблиці (рис.2.5).

customer_id	firstname	lastname	address_id	telephor	email	password	salt
1	Віталій		1	0508027	svinovej@	4dcf96c262	5e93
2	Світлана	Перекопная	2	0955179	closed@x:	af7e044fce!	fd4c
3	alex		4	alex	sasha.u.1!	53c8b6a80c	3b8e
4	Николай	Соколов	5	0634589	kolyan007_	3f1582df36!	1b2c
5	Кирилл	Бабошин	6	0990384	karimanik@	360cecb46!	bb0e
6	Сергей	Морозов	7	0990268	Exy52@b!	09b7c5982!	0a8e
7	Артем	Целик	8	3806363	celik_arter	65021d275!	aac4

Рисунок 2.5 – Скриншот таблиці користувачів



Є можливість створювати групи для управління сайтом, а також групи користувачів в магазині та надавати їм привілеї доступу на розсуд адміністратора.

Для товарів також створюється кілька таблиць. Основну таблицю показано на рисунку 2.6.

product_id	quantit	stock_status_id	manufacturer_id	shipping	price	date_added	date_modified	viewed	date_available	image
91	9	7	12	1	7500.000	2021-04-15 21:58:1	2021-04-17 10:57:11	376	2021-04-14	data/Sol
87	0	5	12	1	7500.000	2021-04-15 21:08:4	2021-04-18 23:23:37	241	2021-04-14	data/Sol
88	0	8	12	1	7500.000	2021-04-15 21:08:4	2021-04-18 23:24:42	236	2021-04-14	data/Sol
89	0	6	12	1	7500.000	2021-04-15 21:08:4	2021-04-18 00:31:18	260	2021-04-14	data/Sol
90	9	7	12	1	7500.000	2021-04-15 21:08:4	2021-04-17 10:57:20	309	2021-04-14	data/Sol
83	8	7	12	1	7500.000	2021-04-15 20:53:5	2021-04-17 10:57:11	315	2021-04-14	data/Sol
110	0	8	12	1	1200.0000	2021-04-15 19:45:0	2021-04-18 23:23:37	210	2021-04-18	data/Pill
84	0	6	12	1	7500.000	2021-04-15 21:08:3	2021-04-18 00:31:25	236	2021-04-14	data/Sol
67	10	7	12	1	1350.0000	2021-04-10 19:55:4	2021-04-18 15:52:18	413	2021-04-17	data/Mix
70	0	8	12	1	1000.0000	2021-04-18 17:28:3	2021-04-20 23:23:37	460	2021-04-17	data/Bea Red.jpg

Рисунок 2.6 - Скриншот таблиці товарів

Для кожного замовлення створюємо запис у таблиці замовлень, що наведено на рисунку 2.7.

order_id	store_id	customer_id	firstname	lastname	email	telephone	payment_firstname	payment_lastname	payment_method	payment_city
24	0	17	Павел	Демин	pasha_demin@gmail.com	0952298140	Павел	Демин	Банковский перевод или платежная система	
26	0	18	Василь	Кліщ	Vasya-k111@gmail.com	0979034918	Василь	Кліщ	Оплата при доставке	Івано-Франківськ
27	0	0	Василь	Кліщ	vasya-k111@gmail.com	+380979034918	Василь	Кліщ	Оплата при доставке	
28	0	0	Наталія	Мозиль	mozilnv@gmail.com	0958689655	Наталія	Мозиль	Оплата при доставке	
29	0	0	shargil	ahmed	shargil.pk.98@gmail.com	015211	shargil	ahmed	Банковский перевод или платежная система	
30	0	0	shargil	ahmed	empty@localhost	015211278624	shargil	ahmed	Банковский перевод или платежная система	

Рисунок 2.7 - Скриншот таблиці замовлень

Таблиця, де можна подивитися інформацію щодо сплати замовлень наведено на рисунку 2.8.

payment_code	comment	payment_address_1	payment_postcode	payment_code	total
bank_transfer	Вариант оплаты: GooglePay  			490.0000	USD
cod	Вариант оплаты: Приват24  	Макарова 52	49000	490.0000	USD
bank_transfer	Отчество: Сергеевич   Вариант оплаты: Наложенный...	Мелитополь, Вакуленчука, 104	72300	450.0000	USD
bank_transfer	Отчество: Дмитриевич   Вариант оплаты: Не опреде...			1000.0000	USD
cod	Вариант оплаты: Приват24  	Чумаченко	69104	370.0000	USD

Рисунок 2.8 - Скриншот таблиці інформації щодо сплати замовлень

Крім того в базі даних є таблиця з описом варіантів доставки та їх вартості, що наведена на рисунку 2.9.

ess_1	shipping_city	shipping_postcode	shipping_country	shipping_country_id	shipping_method	shipping_code	shi
Макарова 52	Днепр	49000	Украина		220 Доставка Новой Почтой	citylink.citylink	
Мелитополь, Вакуленчука, 104	Запорожская обл.	72300	Украина		176 Доставка Новой Почтой	free.free	
Каменское, пр. Свободы, 64, кв.11	Днепропетровск область	51909			0 Доставка Новой Почтой	free.free	
Чумаченко	Запорожье	69104	Украина		220 Самовывоз в Запорожье	pickup.pickup	

Рисунок 2.9 - Скриншот таблиці способів доставки

Для сплати замовлення є декілька стандартних способів. Ці способи показані на рисунку 2.10.

payment_id	payment_method	payment_c
1		
2	Оплата при доставке	
7	Банковский перевод или платежная система	
9	GooglePay	

Рисунок 2.10 - Скриншот таблиці способів оплати

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

#### Розробка інтерфейсу системи

Інтерфейс взаємодії користувача з системою повинен бути інтуїтивно зрозумілим для користувача, бути естетично привабливим та відповідати ергономічним вимогам. Введення інформації користувачем здійснюється за допомогою клавіатури, миші, тачскріну на портативних пристроях. Інформація відображається на дисплеї персонального комп'ютера і мобільних пристроїв.

На головній сторінці сайту розташований пошук товарів, каталог товарів інтернет-магазину, вказані засоби зв'язку з менеджерами, кошик товарів. Також на головній сторінці розташована форма введення логіну та паролю для входу в особистий кабінет, слайдер банерів спеціальних пропозицій та блок рекомендованих товарів (рис. 3.1).

Під шапкою сайту розташоване меню сайту з випадючими при наведенні списками, що відображають підкатегорії товарів (рис.3.2).

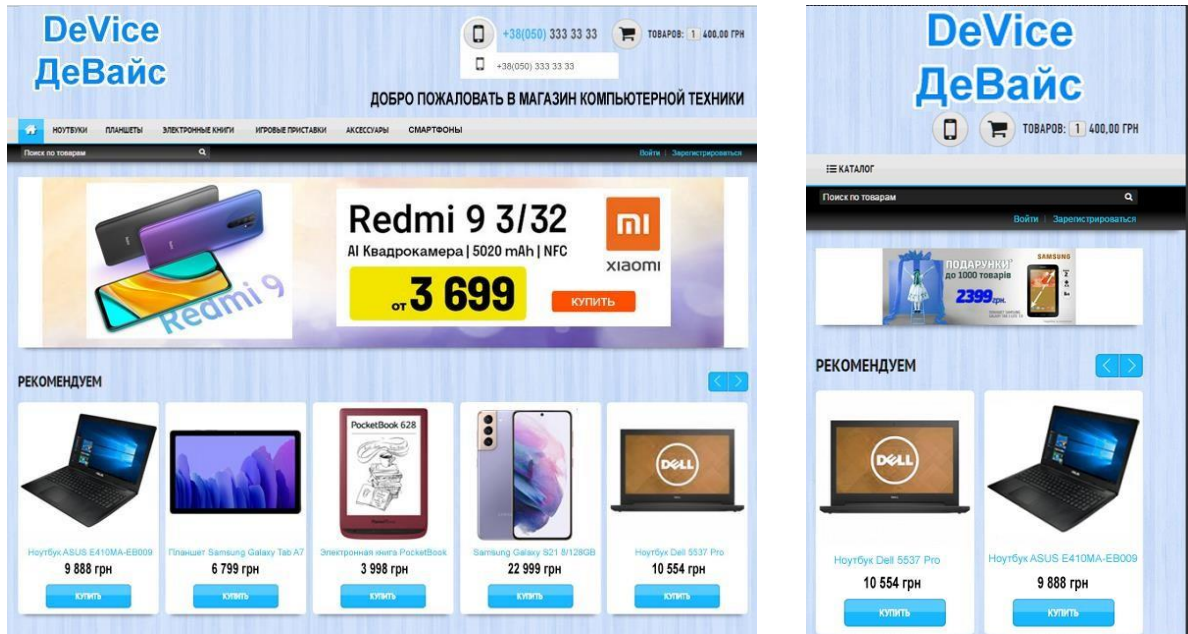


Рисунок 3.1 - Головна сторінка сайту на ПК та мобільному пристрої

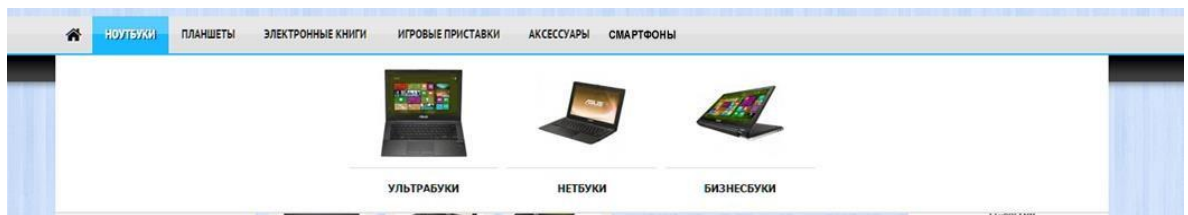


Рисунок 3.2 - Верхнє випадаюче меню

Зліва розташоване навігаційне меню по товарах, серед яких можна вибрати категорію товарів. В даному прикладі, це ноутбуки. Після вибору категорії товару, відвідувач може обрати підкатегорію товару. В цьому випадку це ультрабуки (рис.3.3). В результаті, на сторінці виводяться товари обраної категорії та обраної підкатегорії з короткою інформацією, ціною та технічними характеристиками. До того у цьому ж вікні можна або купити товар, що сподобався, або швидко подивитися інформацію про нього.

На рисунку 3.4 наведена сторінка докладного опису товару. На цій сторінці для порівняння наведені ціни з аналогічних систем електронної комерції,

шляхом парсингу контенту ресурсів конкурентів і відображення цін в окремому блоці. Характеристики товару можна переглянути на відповідній вкладці.



Рисунок 3.3 – Категорії та підкатегорії каталогу товарів

Детальний опис товару можна прочитати, обравши вкладку «Опис» на сторінці товару, що цікавить (рис. 3.5). Можна обрати інші відповідні вкладки, щоб подивитися характеристики товару або відгуки користувачів.

На вкладці «Відгуки» є можливість крім перегляду відгуків інших користувачів і залишити свої.

Для зв'язку з менеджерами магазину, щодо питань про товар або його роботу, можна скористатися «Контактами» (рис.3.6).

Для перегляду вмісту кошика покупок або оформлення замовлення, доступна область «Перегляд кошика/Оформлення замовлення». Області контактів та кошики замовлень доступні на будь-якій сторінці сайту.



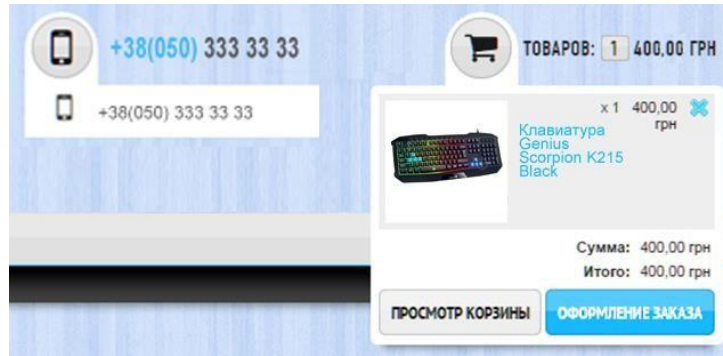


Рисунок 3.6 - Области «Контакты» та «Кошик замовлень»

**ОФОРМЛЕНИЕ ЗАКАЗА**


Способ доставки		Покупатель			
<b>Доставка по городу Запорожье</b> <input type="radio"/> Доставка по городу Запорожье 50 грн <b>Самовывоз</b> <input checked="" type="radio"/> Самовывоз в Запорожье 0 грн <b>Доставка Новой Почтой</b> <input type="radio"/> Доставка Новой Почтой 40 грн		Зарегистрироваться на сайте: <input type="radio"/> Да <input checked="" type="radio"/> Нет E-mail: <input type="text"/> Имя: <input type="text"/> Телефон: <input type="text"/> Вариант оплаты: <input type="text" value="Приват24"/> Комментарий: <input type="text"/>			
Способ оплаты		Адрес доставки			
<input checked="" type="radio"/> Оплата при доставке <input type="radio"/> Банковский перевод или платежная система		<b>Место встречи для самовывоза</b> Адрес: <input type="text"/>			
Фото	Наименование товара	Модель	Кол-во	Цена	Итого
	<b>Клавиатура Genius Scorpion K215 Black</b>	31310474105	1	400,00 грн	400,00 грн
Сумма: 400,00 грн					
Самовывоз в Запорожье: 0 грн					
<b>Итого: 400,00 грн</b>					
<input checked="" type="checkbox"/> Я прочитал и согласен с правилами <a href="#">Условия соглашения</a>					
<b>ОФОРМИТЬ ЗАКАЗ</b>					

Рисунок 3.7 - Оформлення замовлення

Для пошуку товарів потрібно написати в полі для пошуку або назву товару, або виробника, або модель товару. У даному прикладі пошук здійснюється за виробником. Результати пошуку представляються у вигляді списку товарів, знайдених відповідно до заданого критерію пошуку (рис. 3.8). Також є можливість порівняння товарів за їх характеристиками, ціною, зовнішнім виглядом. Для цього потрібно додати товар до порівняння натисканням кнопки «До порівняння» (рис.3.9).

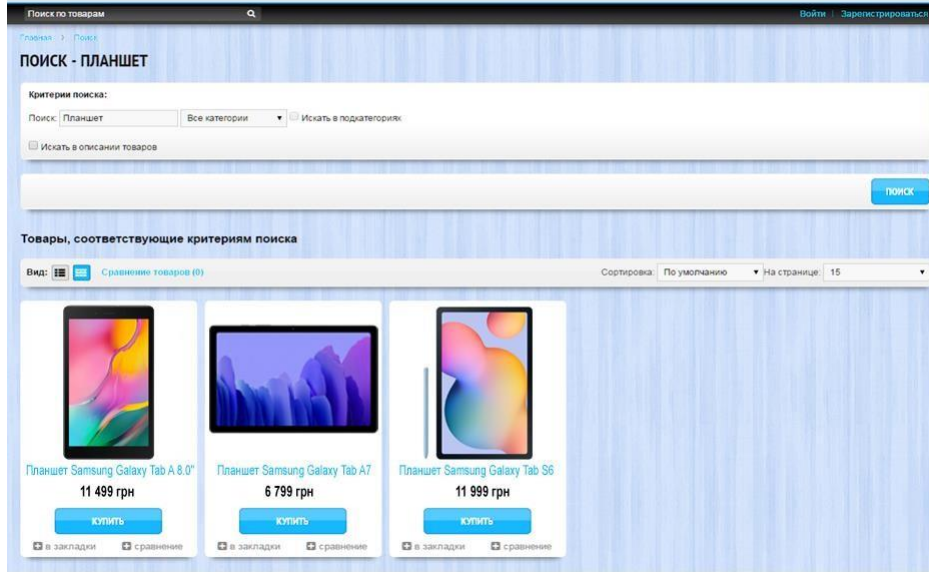


Рисунок 3.8- Пошук товару на фірмі виробників й результати

СРАВНЕНИЕ ТОВАРОВ

Наименование	Ноутбук Asus ZenBook 14 UX435EA-A5006T	Ноутбук Huawei MateBook D 14	Ноутбук Lenovo Yoga Slim 7 14IIL05
Изображение			
Цена	28 999 грн	21 999 грн	26 999 грн
Модель	90NB0RS1-M00600	53010XJD	82A100HQRA
Производитель	Asus	Huawei	Lenovo
Рейтинг	★★★★ На основе 1 отзыва	★★★★ На основе 0 отзывов	★★★★ На основе 0 отзывов
Краткое описание	Экран 14" IPS (1920x1080) Full HD, матовый / Intel Core i5-1135G7 (4.2 ГГц) / RAM 8 ГБ / SSD 512 ГБ / Intel Iris Xe Graphics / без ОД / Wi-Fi / Bluetooth / веб-камера / Windows 10 Home 64bit / 1.19 кг / серый	Экран 14" IPS (1920x1080) Full HD, матовый / AMD Ryzen 7 3700U (2.3 - 4.0 ГГц) / RAM 8 ГБ / SSD 512 ГБ / AMD Radeon RX Vega 10 Graphics / без ОД / Wi-Fi / Bluetooth / веб-камера / Windows 10 Home 64bit / 1.38 кг / серый	Экран 14" IPS (1920x1080) Full HD, матовый / Intel Core i5-1035G1 (1.0 - 3.6 ГГц) / RAM 8 ГБ / SSD 512 ГБ / nVidia GeForce MX350, 2 ГБ / без ОД / Wi-Fi / Bluetooth / веб-камера / Windows 10 Home 64bit / 1.5 кг / серый
<b>Характеристики ноутбука</b>			
Бренд	Asus	Huawei	Lenovo
Платформа	Intel	AMD	Intel
Тип	Ноутбук	Ноутбук	Ноутбук
Тип процессора	11-ое Tiger Lake		10-е Ice Lake
Модель процессора	Intel Core i5-1135G7	AMD Ryzen 7 3700U	Intel Core i5-1035G1
Частота процессора	4.2 ГГц	2.3 - 4.0 ГГц	1.0 - 3.6 ГГц
Количество ядер	4	4	4
Диагональ дисплея	14"	14"	14"
Разрешение	1920x1080	1920x1080	1920x1080
Оперативная память	8 ГБ	8 ГБ	8 ГБ
Жесткий диск	512 ГБ SSD	512 ГБ SSD	512 ГБ SSD
Видеокарта	Интегрированная	Интегрированная	NVIDIA GeForce MX350
Програмное обеспечение	Windows 10 Home 64bit	Windows 10 Home 64bit	Windows 10 Home 64bit
	<a href="#">Купить</a>	<a href="#">Купить</a>	<a href="#">Купить</a>
	<a href="#">Удалить</a>	<a href="#">Удалить</a>	<a href="#">Удалить</a>

Рисунок 3.9- Порівняння обраних товарів



## Розробка програмного коду

Для роботи системи електронної комерції також був розроблений її програмний код.

Якщо користувач вже авторизований, виконується алгоритм входу до магазину та редірект на сторінку профілю. В іншому випадку викликається діалогове вікно для введення даних для авторизації.

Програмний код реалізації кабінету користувача знаходиться в файлі `account.php` та наведений в лістингу 3.1.

### Лістинг 3.1 – Програмний код кабінету користувача

```

        if (!$this->customer->isLoggedIn()) {
            $this->session->data['redirect'] = $this->url-
>link('account/account', '', 'SSL');
            $this->redirect($this->url->link('account/login', '',
'SSL'));
        }
        $this->language->load('account/account');
        $this->document->setTitle($this->language-
>get('heading_title'));

        if (isset($this->session->data['success'])) {
            $this->data['success'] = $this->session-
>data['success'];
            unset($this->session->data['success']);
        } else { $this->data['success'] = ''; }
        $this->data['heading_title'] = $this->language-
>get('heading_title');
        .....
        $this->data['order'] = $this->url-
>link('account/order', '', 'SSL');

        if (file_exists(DIR_TEMPLATE . $this->config-
>get('config_template') . '/template/account/account.tpl')) {
            $this->template = $this->config-
>get('config_template') . '/template/account/account.tpl';
        } else {$this->template =
'default/template/account/account.tpl';}

        $this->children = array(
            'common/column_left',
            'common/column_right',
            'common/content_top',
            'common/content_bottom',

```

```

        'common/footer',
        'common/header'
    );
    $this->response->setOutput($this->render());

```

При здійсненні пошуку, пошуковий запит вводиться в спеціальний тег `input` з атрибутом `name="search"`. Система шукає співпадіння у назві та описі товару, перебираючи всі товари. Результати пошуку виводяться в представлення `search`. Програмний код розміщений в файлі `search.php` та наведений в лістингу 3.2.

### Лістинг 3.2 – Програмний код здійснення пошуку

```

    $this->language->load('product/search');
    $this->load->model('catalog/category');
    $this->load->model('catalog/product');
    $this->load->model('tool/image');
    if (isset($this->request->get['search'])) {
        $search = $this->request->get['search'];
    } else {
        $search = '';
    }
    $this->data['categories'] = array();
    $categories_1 = $this->model_catalog_category->getCategories(0);
    foreach ($categories_1 as $category_1) {$level_2_data = array();
    $categories_2 = $this->model_catalog_category->getCategories($category_1['category_id']);
    foreach ($categories_2 as $category_2) { $level_3_data = array();
    $categories_3 = $this->model_catalog_category->getCategories($category_2['category_id']);
    foreach ($categories_3 as $category_3) {
    $level_3_data[] = array(
        'category_id' => $category_3['category_id'],
        'name' => $category_3['name']);
    }
    $level_2_data[] = array(
        'category_id' => $category_2['category_id'],
        'name' => $category_2['name'],
        'children' => $level_3_data);
    }
    $this->data['categories'][] = array('category_id' => $category_1['category_id'],
        'name' => $category_1['name'],
        'children' => $level_2_data); }
    $this->data['products'] = array();

```

```

$this->data['sorts'] = array();
$this->data['sorts'][] = array('text' => $this->language-
>get('text_default'),
                             'value' => 'p.sort_order-ASC',
                             'href' => $this->url->link('product/search',
                             'sort=p.sort_order&order=ASC' . $url) );
$this->data['sorts'][] = array('text' => $this->language-
>get('text_name_asc'),
                             'value' => 'pd.name-ASC', 'href' => $this->url-
>link('product/search', 'sort=pd.name&order=ASC' . $url));

$this->response->setOutput($this->render());

```

Якщо відвідувач бажає залишає відгук про товар, йому треба додатково пройти перевірку на спам бота. Для цього необхідно ввести сарча, ввести код, що зображений на рисунку. Після цього відгук повинен узгодити адміністратор та підтвердити його валідність. Механізм відгуків реалізований в файлі `product.php` та наведений в лістингу 3.3.

### Лістинг 3.3 – Програмний код здійснення відгуків

```

$this->data['tab_review'] = sprintf($this->language-
>get('tab_review'), $product_info['reviews']);
$this->data['md_review_count'] = $product_info['reviews'];
$this->data['reviews'] = sprintf($this->language-
>get('text_reviews'), (int)$product_info['reviews']);
if ($this->config->get('config_review_status')) {
    $rating = (int)$result['rating'];
} else {
    $rating = false;
}

$this->data['products'][] = array(
    'product_id' => $result['product_id'],
    'thumb'      => $image,
    'name'       => $result['name'],
    'price'     => $price,
    'special'   => $special,
    'rating'    => $rating,
    'reviews'   => sprintf($this->language->get('text_reviews'),
(int)$result['reviews']),
    'href'      => $this->url->link('product/product',
'product_id=' . $result['product_id']));
$review_total = $this->model_catalog_review-
>getTotalReviewsByProductId($product_id);
$results = $this->model_catalog_review-
>getReviewsByProductId($product_id, ($page - 1) * 5, 5);
foreach ($results as $result) {
$this->data['reviews_array'][] = array(

```

```
'author' => $result['author'],
'text' => strip_tags($result['text']),
'rating' => (int)$result['rating'],
'reviews' => sprintf($this->language->get('text_reviews'),
(int)$review_total),
'date_added' => date($this->language-
>get('date_format_short'),
strtotime($result['date_added']));}
```

При оформленні купівлі товару користувач має заповнити форму замовлення, де йому потрібно вказати свої данні (ім'я, телефон, електронну пошту, адресу доставки), спосіб доставки та спосіб оплати. Програмний код оформлення замовлення реалізований в файлі `simplecheckout.php` та наведений в лістингу 3.4.

#### Лістинг 3.4 – Програмний код здійснення оформлення замовлення

```
if (!$this->customer->isLoggedIn() && $this->config-
>get('simple_disable_guest_checkout')) {
    $this->session->data['redirect'] = $this->url-
>link('checkout/simplecheckout', '', 'SSL');
    $this->redirect($this->url-
>link('account/login', '', 'SSL'));
    $this->language->load('checkout/simplecheckout');
    $this->document->setTitle($this->language-
>get('heading_title'));
    $this->data['simplecheckout_shipping'] = $this-
>getChild('checkout/simplecheckout_shipping');
    $this->data['simplecheckout_payment'] = $this-
>getChild('checkout/simplecheckout_payment');
    $this->data['simplecheckout_cart'] = $this-
>getChild('checkout/simplecheckout_cart');
    $this->data['simplecheckout_customer'] = $this-
>getChild('checkout/simplecheckout_customer');
    $this->data['has_shipping'] = $this->cart->hasShipping();
    $shipping_address_same = $this->simple->shipping_address_same;
    if ($this->validate() && ($asap || (!$asap && $this-
>data['simple_create_order']))) {
        $order_id = $this->order();
        $this->simple->save_custom_data(Simple::SET_CHECKOUT_CUSTOMER,
Simple::OBJECT_TYPE_ORDER, $order_id);
        if (!$shipping_address_same) {
            $this->simple->save_custom_data(Simple::SET_CHECKOUT_ADDRESS,
Simple::OBJECT_TYPE_ORDER, $order_id);
        }
        $payment_method = $this->simple->payment_method;
        $this->request->server['REQUEST_METHOD'] =
'GET';
    }
}
```

```

        $this->data['payment_form'] = $this-
>getChilд('payment/' . $payment_method['code']);

```

Якщо при оформленні замовлення, змінювати спосіб доставки, то динамічна підзавантажуються відповідні поля. Наприклад, якщо обрати спосіб доставки самовивіз, то поле адреси доставки зникне.

Код асинхронного перезавантаження форми при зміні значення полів реалізовано в файлах checkout.html та simplecheckout.js, наведений в лістингу 3.5.

### Лістинг 3.5 – Програмний код перезавантаження форми замовлення

```

<input type="radio" name="shipping_method"
value="citylink.citylink" id="citylink.citylink"
checked="checked"
onchange="simplecheckout_reload('shipping_changed');" />
function simplecheckout_reload(from) {
    var data =
jQuery('#simplecheckout_form').find('input,select,textarea').s
erialize();

    if (!data.length) {
        return;
    }
    set_location_hash(from);
    jQuery.ajax({
        url:
'index.php?' + simple_route + 'route=checkout/simplecheckout',
        data: data,
        type: 'POST',
        dataType: 'text',
        beforeSend: function() {
            block_form();
            overlay_simplecheckout();
            overlay_button();
        }
    },
    jQuery('#payment_form_reload').text(jQuery('#saving_changes').
text());
    },
    success: function(data) {
        jQuery('#simplecheckout_form').replaceWith(data);
        simplecheckout_init();
        make_tab();
        scroll_to_error();
        if (from == 'create_order' &&
can_submit_payment_form() && !payment_form_visible() &&
simple_asap) { payment_form_submit(); }
    },
    error: function(xhr, ajaxOptions, errorThrown) {

```

```

        console.log(throwError + "\r\n" + xhr.statusText
+ "\r\n" + xhr.responseText); unblock_form();
overlay_remove();      }    });}

```

При додаванні товару до списку порівняння, товари висвічуються поруч один з одним, та порівняння виконується за атрибутами товарів.

Код порівняння товарів реалізований в файлі `compare.php` та наведений в лістингу 3.6.

### Лістинг 3.6 – Програмний код здійснення порівняння товарів

```

if (!isset($this->session->data['compare']))
$this->session->data['compare'] = array();
foreach ($this->session->data['compare'] as $key =>
$product_id) {$product_info = $this->model_catalog_product-
>getProduct($product_id);
if ($product_info) {if ($product_info['image']) {
    $image = $this->model_tool_image-
>resize($product_info['image'], $this->config-
>get('config_image_compare_width'), $this->config-
>get('config_image_compare_height'));}
else {    $image = false;}
if (($this->config->get('config_customer_price') && $this-
>customer->isLoggedIn()) || !$this->config-
>get('config_customer_price')) {
    $price = $this->currency->format($this->tax-
>calculate($product_info['price'],
$product_info['tax_class_id'], $this->config-
>get('config_tax')));}
else {    $price = false;}    $attribute_data = array();
$attribute_groups = $this->model_catalog_product-
>getProductAttributes($product_id);
foreach ($attribute_groups as $attribute_group) {
    foreach ($attribute_group['attribute'] as $attribute) {
        $attribute_data[$attribute['attribute_id']] =
$attribute['text'];    }}
    $this->data['products'][$product_id] =
array(
        'product_id'    =>
$product_info['product_id'],
        'name'          =>
$product_info['name'],
        'thumb'        => $image,
        'price'        => $price,
        'description'  =>
utf8_substr(strip_tags(html_entity_decode($product_info['descr
iption'], ENT_QUOTES, 'UTF-8')), 0, 200) . '...',
        'model'        =>
$product_info['model'],

```

```

                'manufacturer' =>
$product_info['manufacturer'],
                'availability' => $availability,
                'rating'      =>
(int)$product_info['rating'],
                'attribute'    => $attribute_data,
                'href'        => $this->url->link('product/product',
'product_id=' . $product_id),
                'remove'      => $this->url->link('product/compare',
'remove=' . $product_id)    );
foreach ($attribute_groups as $attribute_group) {$this-
>data['attribute_groups'][$attribute_group['attribute_group_id
']]['name'] = $attribute_group['name'];
foreach ($attribute_group['attribute'] as $attribute) {$this-
>data['attribute_groups'][$attribute_group['attribute_group_id
']]['attribute'][$attribute['attribute_id']]['name'] =
$attribute['name'];    }} else {unset($this->session-
>data['compare'][$key]); }    }

```

При кліках по кнопках соціальних мереж, сторінка поститься до відповідної соціальної мережі.

Код для зв'язку з соціальними мережами розташований в файлі `product.php` та наведений в лістингу 3.7.

### Лістинг 3.7 – Програмний код для зв'язку з соціальними мережами

```

<div class="addthis_default_style addthis_32x32_style">
    <a class="addthis_button_vk at300b"
title="VKontakte" target="_blank" href="#"><span class="at4-
icon aticon-vk" style="background-color: rgb(50, 80,
120);"><span class="at_ally">Share on vk</span></span></a>
    <a class="addthis_button_facebook at300b"
title="Facebook" href="#"><span class="at4-icon aticon-
facebook" style="background-color: rgb(48, 88, 145);"><span
class="at_ally">Share on facebook</span></span></a>
    <a class="addthis_button_twitter at300b"
title="Twitter" href="#"><span class="at4-icon aticon-twitter"
style="background-color: rgb(44, 168, 210);"><span
class="at_ally">Share on twitter</span></span></a>
    <a class="addthis_button_google at300b"
title="Google" target="_blank" href="#"><span class="at4-icon
aticon-google" style="background-color: rgb(8, 104,
185);"><span class="at_ally">Share on google</span></span></a>
    <a class="addthis_button_email at300b" title="e-
mail" target="_blank" href="#"><span class="at4-icon aticon-
email" style="background-color: rgb(115, 138, 141);"><span
class="at_ally">Share on email</span></span></a>
</div>

```

Показ актуальних цін обраних товарів з сайтів існуючих інтернет-магазинів реалізована парсингом сайтів Алло, Stylus, MOYO, Citrus. Для цього використовується бібліотека Simple Html Dom. Рядок запиту формується з відповідної сигнатури запиту сайту конкурента та назви товару. З відповідних тегів забирається ціна та виводиться на сторінці товару проектованої системи електронної комерції. Код парсингу сайтів-конкурентів розміщується в файлі parsermodel.php та наведений в лістингу 3.8.

### Лістинг 3.8 – Програмний код парсингу цін з сайтів-конкурентів

```

        $this->load->model("parsermodel");    }
public function index()    {
        require_once APPPATH."libraries/simple_html_dom.php";
    $content = $this->LoadFromCURL($WordKey, 0);
    if(isset($content['Error'])) return;
    //Parsing OTHER SITES
        $pos = 1;
        foreach($content['html']->find
    $necessary_tag_expression) as $cite) {
            $cite = strip_tags($cite);
            $cite = strtolower($cite);
            if(strpos($cite, $DomainName) !== FALSE)
    {
                //Insert To DB
            $this->parsermodel->InsertInToDB($DomainName, $WordKey,
    $pos);

                $data["site"] = $DomainName;
                $data["WordKey"] = $WordKey;
                $data["Position"] = $pos;
                $this->load->view("Result", $data);
                return;
            }
    $pos++;
        }
        $data["site"] = $DomainName;
        $data["WordKey"] = $WordKey;
    }
private function LoadFromCURL($WordKey, $UseProxy)    {
        $url_allo =
    "http://allo.ua/catalogsearch/result/?q".$WordKey;
        $url_stylus =
    "http://stylus.com.ua/search?q=".$WordKey;
        $url_moyo =
    "http://www.moyo.ua/search.html?posted=1&s%5Btext%5D=".$WordKey
    y."&x=0&y=0";
        $url_citrus =
    "http://www.citrus.ua/search/?q=".$WordKey;

```



```

$curl = curl_init();
        //Options
        curl_setopt($curl, CURLOPT_URL, $url);
$html = new simple_html_dom();
        $html->load(curl_exec($curl));
        $Data = array();
if($html->find($necessary_selection)) {
        curl_close($curl);
        $Data['html'] = $html;
        return $Data;
}
else{
        curl_close($curl);
        $Data['Error'] = "Error";
        return $Data;
}
}

```

Код каталогу товарів наведений в лістингу 3.9.

### Лістинг 3.9 – Програмний код реалізації каталогу товарів

```

<?php echo $header; ?>
<div id="container-center">
<?php echo $column_left; ?><?php echo $column_right; ?>
<div id="content"><?php echo $content_top; ?>
        <h1><?php echo $heading_title; ?></h1>
        <?php if ($categories) { ?>
        <div class="category-list">        <ul>
                <?php foreach ($categories as $category) { ?>
                <li>
                        <a href="<?php echo $category['href']; ?>">
" style="width:100px;height:100px" />
<div style="font-size:14px;">
                <?php echo $category['name']; ?> </div></a></li>
                <?php } ?>        </ul>        </div>        <?php } ?>
        <?php if ($products) { ?>
        <div class="product-filter">
                <div class="display"><b><?php echo $text_display; ?></b>
<i class="icon-th-list" title="<?php echo $text_list; ?>"></i>
<a onclick="display('grid');" class="icon-th" title="<?php
echo $text_grid; ?>"></a></div>
                <div class="product-compare"><a href="<?php echo $compare;
?>" id="compare-total"><i class="icon-tasks"></i> <?php echo
$text_compare; ?></a></div>
                <div class="limit"><b><?php echo $text_limit; ?></b>
                <select onchange="location = this.value;">
                <?php foreach ($limits as $limits) { ?>
                <?php if ($limits['value'] == $limit) { ?>
                <option value="<?php echo $limits['href']; ?>"
selected="selected"><?php echo $limits['text']; ?></option>
                <?php } else { ?>

```

```

        <option value="<?php echo $limits['href']; ?>"><?php
echo $limits['text']; ?></option>          <?php } ?>
<?php } ?>      </select>      </div>
    <div class="sort"><b><?php echo $text_sort; ?></b>
    <select onchange="location = this.value;">
        <?php foreach ($sorts as $sort) { ?>
        <?php if ($sorts['value'] == $sort . '-' . $order) {
?>
            <option value="<?php echo $sorts['href']; ?>"
selected="selected"><?php echo $sorts['text']; ?></option>
            <?php } else { ?>
            <option value="<?php echo $sorts['href']; ?>"><?php
echo $sorts['text']; ?></option>          <?php } ?>
<?php } ?>      </select>      </div> </div>

```

## Конфігурування та тестування системи

Додавання нового товару в адміністративній частині складається з заповнення декількох вкладок з параметрами, як показано на рисунку 3.10. Для додавання товару необхідно обрати Каталог/Товар/Додати. На вкладці «Загальне» потрібно заповнити текстову частину опису товару. Ні - це назва, що відображається на картці товару. Title - мета-тег, що відображається в пошуковій видачі та в назві вкладки браузеру. Поле «Опис» заповнюється текстом про відповідний товар.

На вкладці «Данні» вказується модель, ціна, кількість, дата та основне зображення товару (рис.3.11).

На вкладці «Зв'язки» встановлюються зв'язки з категоріями і виробником (рис.3.12).

Встановлення зв'язку з категоріями та виробником потрібно для того, щоб товар відображався на відповідних йому категоріях. Технічно ці зв'язки є додаванням запису в таблицю бази даних, що відображає відповідність товару категорії.

**:: Console**

Главная | Каталог | Дополнения | Продажи | Система

### Ноутбук Lenovo Yoga Slim 7 14IIL05

Общие | **Данные** | Связи | Атрибуты | Изображения

Название товара:

Н1:

Title:

Мета-тег Description:

Описание: 

Экран 14" IPS (1920x1080) Full HD, матовый / Intel Core i5-1035G1 (1.0 - 3.6 ГГц) / RAM 8 ГБ / SSD 512 ГБ / nVidia GeForce MX350, 2 ГБ / без ОД / Wi-Fi / Bluetooth / веб-камера / Windows 10 Home 64bit / 1.5 кг / серый

Рисунок 3.10 – Заполнения опису товару

### Ноутбук Lenovo Yoga Slim 7 14IIL05


Общие | **Данные** | Связи | Атрибуты | Изображения

\* Модель:

Цена:

Количество:

Вычитать со склада:

Изображение товара:  [Обзор](#) | [Очистить](#)

Дата поступления:

Статус:

Порядок сортировки:

Рисунок 3.11 – Заполнения даних про товар

Ноутбук Lenovo Yoga Slim 7 14IIL05

Общие Данные Связи **Атрибуты** Изображения

Производитель: LENOVO

Главная категория: Ноутбуки

Показывать в категориях:

- Ноутбуки
- Ноутбуки > Бизнесбуки
- Ноутбуки > Нетбуки
- Ноутбуки > Ультрабуки

[Выделить всё](#) / [Снять выделение](#)

Рисунок 3.12 – Встановлення зав'язків з категоріями товару

На вкладці «Атрибути» додаються технічні характеристики товару (рис.3.13), а на вкладці «Зображення» додаються додаткові зображення товару (рис.3.14).

Ноутбук Lenovo Yoga Slim 7 14IIL05

Общие Данные Связи **Атрибуты** Изображения

Атрибут:	Текст:
Платформа	Intel
Тип процессора	10-e Ice Lake
Модель процессора	Intel Core i5-1035G1
Частота процессора	1.0 - 3.6 ГГц
Количество ядер	4
Диагональ дисплея	14"
Разрешение	1920x1080
Оперативная память	8 ГБ

Рисунок 3.13 – Заповнення технічних характеристик

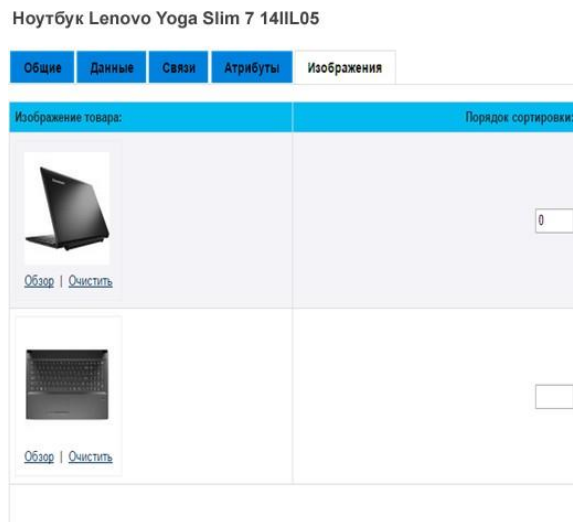


Рисунок 3.14 – Додавання додаткових зображень товару

Результат додавання товару видно на сторінці відповідної категорії (рис.3.15).

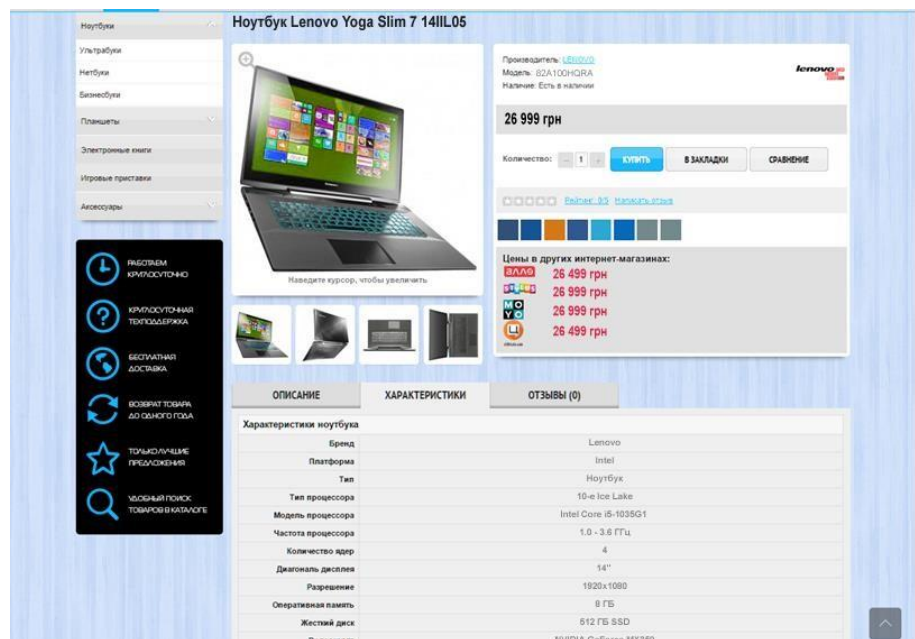


Рисунок 3.15 – Результат додавання товару

Операція порівняння товарів тестується додаванням обраних товарів до списку порівняння натисканням на посилання «Порівняння» в модулі товару (рис.3.16).

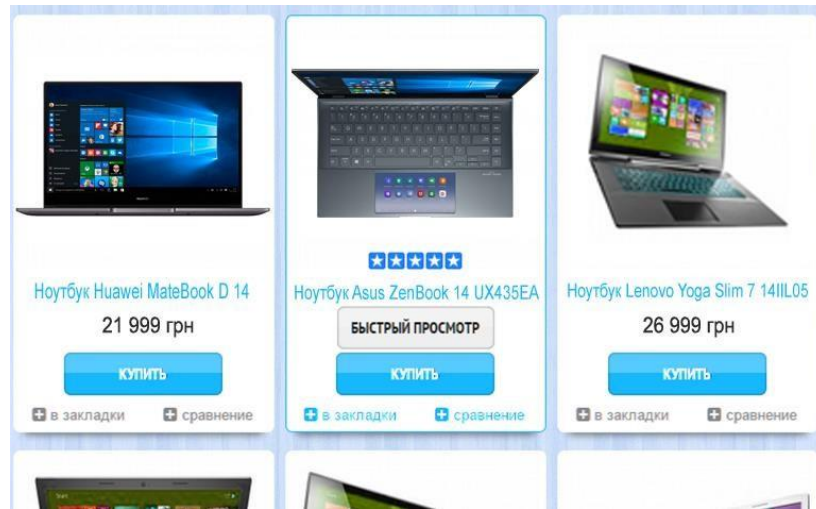


Рисунок 3.16 – Додавання товарів до порівняння

Перегляд обраних для порівняння товарів доступний за посиланням "Порівняння товарів» на сторінці категорії (рис.3.17).

СРАВНЕНИЕ ТОВАРОВ			
Описание	Ноутбук Asus ZenBook 14 UX435EA-A5006T	Ноутбук Huawei MateBook D 14	Ноутбук Lenovo Yoga Slim 7 14IIL05
Наименование	Ноутбук Asus ZenBook 14 UX435EA-A5006T	Ноутбук Huawei MateBook D 14	Ноутбук Lenovo Yoga Slim 7 14IIL05
Изображение			
Цена	28 999 грн	21 999 грн	26 999 грн
Модель	90NB0RS1-M00600	53010XJD	82A100HQRA
Производитель	Asus	Huawei	Lenovo
Рейтинг	★★★★★ На основе 1 отзыва	☆☆☆☆☆ На основе 0 отзывов	☆☆☆☆☆ На основе 0 отзывов
Краткое описание	Экран 14" IPS (1920x1080) Full HD, матовый / Intel Core i5-1135G7 (4.2 ГГц) / RAM 8 ГБ / SSD 512 ГБ / Intel Iris Xe Graphics / Без ОД / Wi-Fi / Bluetooth / веб-камера / Windows 10 Home 64bit / 1.18 кг / серый	Экран 14" IPS (1920x1080) Full HD, матовый / AMD Ryzen 7 3700U (2.3 - 4.0 ГГц) / RAM 8 ГБ / SSD 512 ГБ / AMD Radeon RX Vega 10 Graphic / Без ОД / Wi-Fi / Bluetooth / веб-камера / Windows 10 Home 64bit / 1.38 кг / серый	Экран 14" IPS (1920x1080) Full HD, матовый / Intel Core i5-1035G1 (1.0 - 3.6 ГГц) / RAM 8 ГБ / SSD 512 ГБ / nVidia GeForce MX350, 2 ГБ / Без ОД / Wi-Fi / Bluetooth / веб-камера / Windows 10 Home 64bit / 1.5 кг / серый
<b>Характеристики ноутбука</b>			
Бренд	Asus	Huawei	Lenovo
Платформа	Intel	AMD	Intel
Тип	Ноутбук	Ноутбук	Ноутбук
Тип процессора	11-ое Tiger Lake		10-е Ice Lake
Модель процессора	Intel Core i5-1135G7	AMD Ryzen 7 3700U	Intel Core i5-1035G1
Частота процессора	4.2 ГГц	2.3 - 4.0 ГГц	1.0 - 3.6 ГГц
Количество ядер	4	4	4
Диагональ дисплея	14"	14"	14"
Разрешение	1920x1080	1920x1080	1920x1080
Оперативная память	8 ГБ	8 ГБ	8 ГБ
Жесткий диск	512 ГБ SSD	512 ГБ SSD	512 ГБ SSD
Видеокарта	Интегрированная	Интегрированная	NVIDIA GeForce MX350
Програмное обеспечение	Windows 10 Home 64bit	Windows 10 Home 64bit	Windows 10 Home 64bit
	<a href="#">КУПИТЬ</a>	<a href="#">КУПИТЬ</a>	<a href="#">КУПИТЬ</a>
	<a href="#">УДАЛИТЬ</a>	<a href="#">УДАЛИТЬ</a>	<a href="#">УДАЛИТЬ</a>

Рисунок 3.17 – Сторінка порівняння товарів

Функція додавання відгуків тестується додаванням нового відгуку на сторінці товару (рис.3.18). Для відправки відгуку, з метою уникнення спаму, необхідно пройти процедуру CAPTCHA вводючи символи, що зображені на картинці у відповідне поле.

Рисунок 3.18 – Додавання відгуку

Для того, щоб відгук було опубліковано, він має пройти модерацию адміністратором, який повинен підтвердити валідність відгуку (рис. 3.19).

Рисунок 3.19 – Схвалення відгуку адміністратором

Для пошуку товару достатньо ввести запит в поле для пошуку і потрібні товари з'являються на одній сторінці (рис.3.20). Для оформлення замовлення необхідно перейти до «Кошику» і оформити замовлення (рис.3.21).

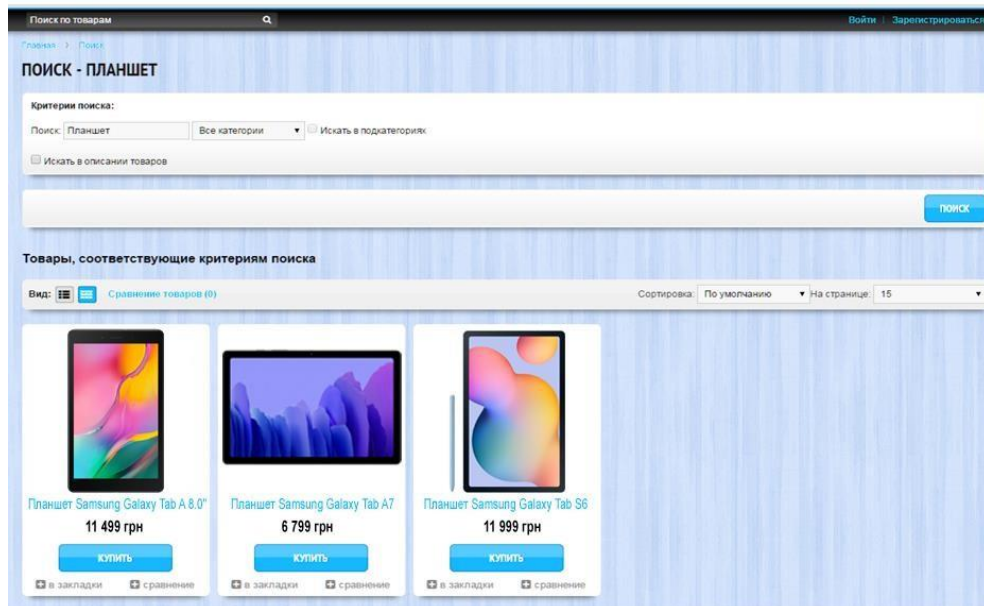


Рисунок 3.20 – Тестування пошуку товару

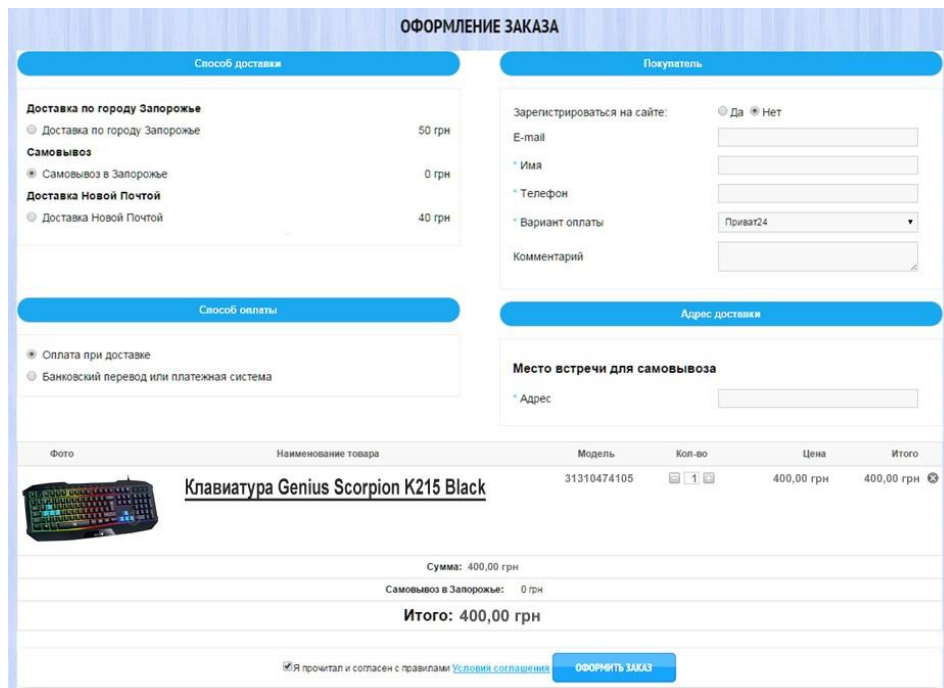


Рисунок 3.21 – Тестування оформлення замовлення



## ВИСНОВКИ

В дипломній роботі була розроблена система електронної комерції з SEO-оптимізованим контентом. Система базується на архітектурі MVC, що дозволяє відокремити бізнес-логіку (модель) від її візуалізації (представлення). Завдяки такому поділу підвищується можливість повторного використання та надається можливість користувачу бачити ті ж самі дані одночасно в різних контекстах та з різних точок зору.

Був проведений аналіз принципів побудови систем електронної комерції та досліджено структуру електронних магазинів. Також здійснено порівняльний аналіз технологій та програмних засобів для розробки веб-застосунків.

На підставі цього аналізу для розробки дипломного проекту була використана мова програмування PHP на архітектурі MVC з базою даних MySQL та веб-сервером Apache. Ці технології є найпопулярнішими і максимально підійшли для вирішення поставлених завдань.

Розроблювана система електронної комерції автоматизує процес замовлення та покупки товару через мережу Інтернет, скорочує час на пошук товару, підвищує продажі за рахунок просування в пошукових системах. За допомогою розробленої системи здійснюється залучення потенційних покупців та ознайомлення їх з товарами та послугами компанії.

Для досягнення поставленої мети в роботі було розроблено структурну схему та алгоритм функціонування системи, спроектовано та реалізовано базу даних системи, розроблено програмний код системи та дружній інтерфейс користувача. Для просування в мережі Інтернет контент системи було оптимізовано згідно розробленого семантичного ядра.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Find out what websites are Built With [Електронний ресурс]. – Режим доступу: <https://builtwith.com/> - Заголовок з екрану.
2. Usage statistics of server-side programming languages for websites [Електронний ресурс]. – [https://w3techs.com/technologies/overview/programming\\_language](https://w3techs.com/technologies/overview/programming_language) - Заголовок з екрану.
3. Белые, серые и черные методы оптимизации [Електронний ресурс]. – Режим доступу: <https://pr-cy.ru/lib/seo/Belye-serye-i-chernye-metody-optimizatsii> - Заголовок з екрану.
4. Ашманов, И. Оптимизация и продвижение сайтов в поисковых системах 4-е издание/ И. Ашманов, А. Иванов. - СПб.: Питер, 2019. – 513 с.
5. Бабаев, А. Раскрутка. Секреты эффективного продвижения сайтов/ А. Бабаев, Н. Евдокимов, М. Боде, Е. Костин.- СПб.: Питер, 2013. - 215 с.
6. Севостьянов, И.О. Продвижение порталов и интернет-магазинов /, Д.В. Иванов, Ф.А. Фиронов, Л.О. Гроховский. - СПб.: Питер, 2014. - 312 с.
7. Поиск и подбор ключевых слов с помощью планировщика ключевых слов Google Ads [Електронний ресурс]. – Режим доступу: [https://ads.google.com/intl/ru\\_ru/home/tools/keyword-planner](https://ads.google.com/intl/ru_ru/home/tools/keyword-planner) - Заголовок з екрану.
8. Дари, К. PHP и MySQL: создание интернет-магазина. 2-е издание / К. Дари, Э.Баланеску.- М.:Вильямс, 2014.- 640 с.
9. Шаблон Bootstrap [Електронний ресурс]. – Режим доступу: <http://bootstrap-ru.com/scaffolding.php> - Заголовок з екрану.
10. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Робин Никсон. Пер. с англ. – СПб.: Питер, 2016. – 767 с.
11. Хопкинс, К. PHP. Быстрый старт / Каллум Хопкинс. Пер. с англ. – М.: Эксмо, 2014. – 159 с.

12. Анализ сайта онлайн [Электронный ресурс]. – Режим доступа: <https://a.pr-cy.ru/> - Заголовок з екрану.
13. Янк, К. PHP и MySQL. От новичка к профессионалу / К. Янк. – М.: Эксмо, 2013. – 384с.
14. Основы проектирования реляционных БД [Электронный ресурс]. - Режим доступа: <http://citforum.ru/database/sql1g/index.shtml> - Заголовок з екрану.
15. Методичні вказівки до виконання дипломної роботи бакалавра для студентів спеціальності 123 «Комп'ютерна інженерія» всіх форм навчання / укл.: Р. К. Кудерметов, О. В. Польська, Н. В. Луценко, Н. В. Щербак, О. В. Зелік. Запоріжжя : НУ «Запорізька політехніка», 2020. 54 с.

## Огляд існуючих систем електронної комерції



Рисунок 1 – Інтерфейс інтернет-магазину «Фокстрот»

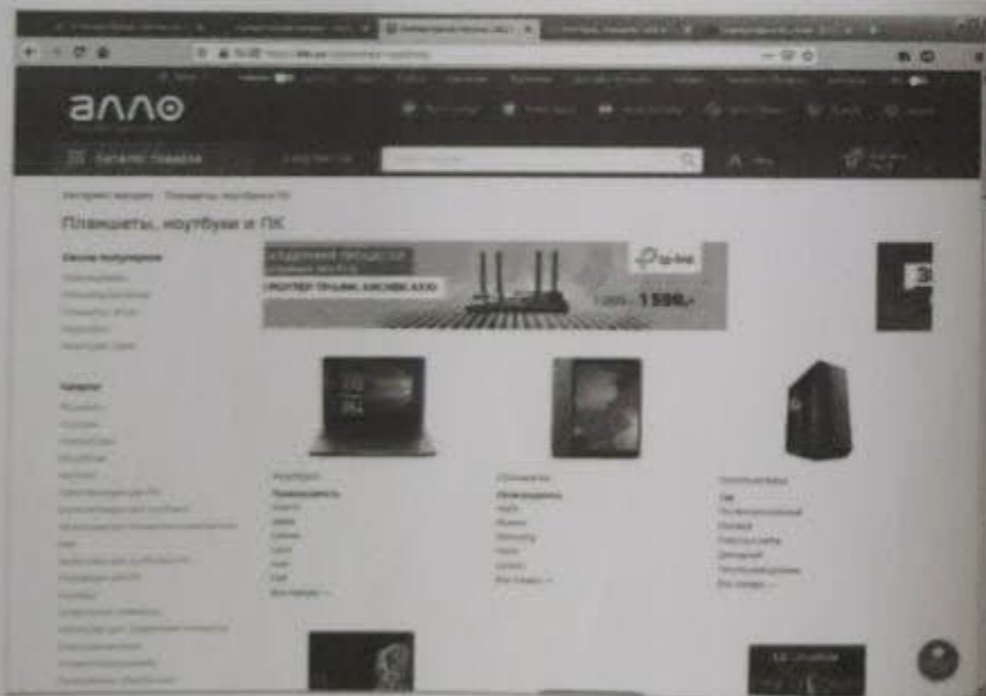


Рисунок 2 – Інтерфейс інтернет-магазину «Алло»



Рисунок 3 – Інтерфейс інтернет-магазину «Comfy»



Рисунок 4 – Інтерфейс інтернет-магазину «Rozetka»

13.02070849.00057 Пл. 1								
						Дня	Міся	Місяць
Ім.	Лист	№ Докум.	Підп.	Дата	Розробка та просування в інтернет системи електронної комерції			
Розроб.	Перев.	Еконтр.				Огляд існуючих систем електронної комерції	Артем І	Артем І
Н.контр.	Затв.					НУЗП КНТ-518сп		

### Аналіз технологій, що використовуються

Таблиця 1 – Технології популярних інтернет-магазинів

	ALLO.UA	ELDORADO.UA	COMFY.UA	ROZETKA.COM.UA
Серверна мова	PHP	PHP JS	PHP	PHP JS
Фреймворк сервер	PHP 7	Express		Express
Бібліотеки JS	jQuery Core-js	jQuery Redux Lodash	jQuery Lodash	jQuery Core-js Lodash
Фреймворк клієнт	Handlebars Vue	React	Vue	Angular React
Веб-сервер	Nginx	Nginx	Nginx	Nginx

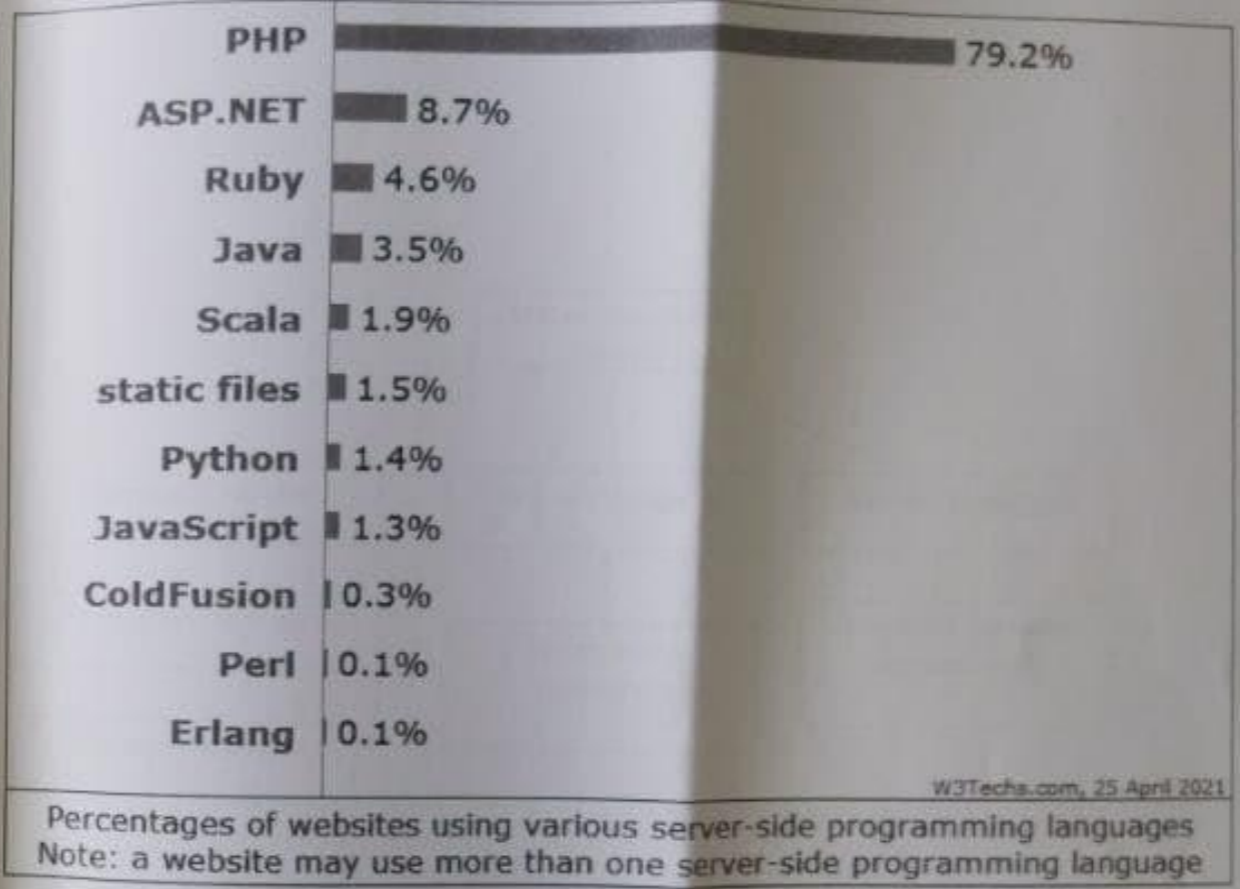


Рисунок 1 – Найбільш поширені серверні мови програмування

					<b>13.02070849.00057 Пл.2</b>			
№	Лист	№ Докум.	Підп.	Дата	Розробка та просування в інтернет системи електронної комерції	Лист	Місяц	Рік
Розроб.		Наковичий О.Д.	<i>[Signature]</i>					
Перев.		Касян М.М.	<i>[Signature]</i>		Аналіз технологій, що використовуються	Листы 2	Листы 3	
Г.контр.						НУЗП КНТ-518сн		
Н.контр.		Зелік О.В.	<i>[Signature]</i>					
Затв.		Кудеристов Р.К.	<i>[Signature]</i>					

## Структура системи електронної комерції

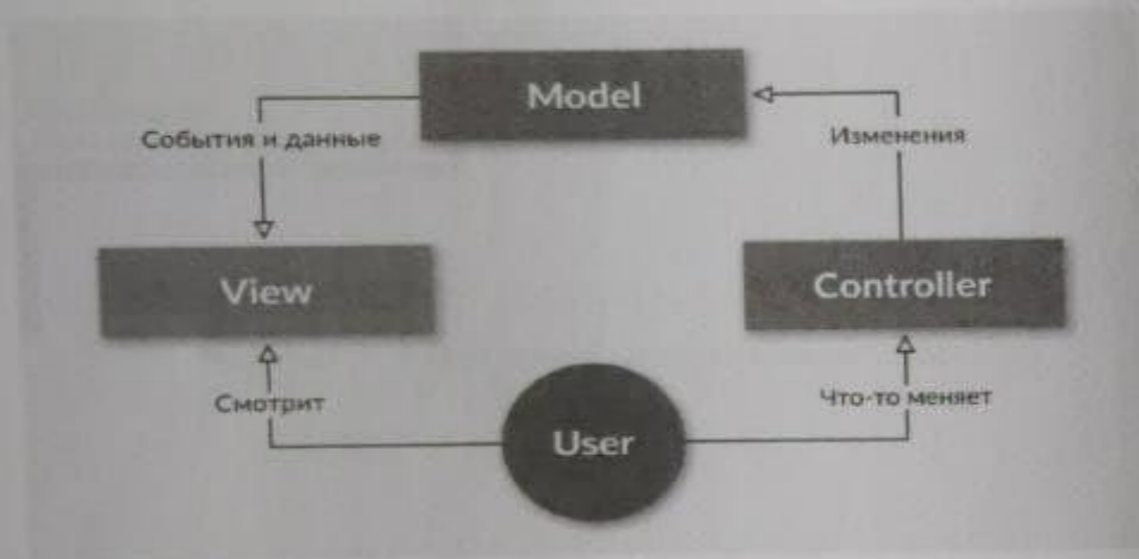


Рисунок 1 - Архітектурний шаблон MVC

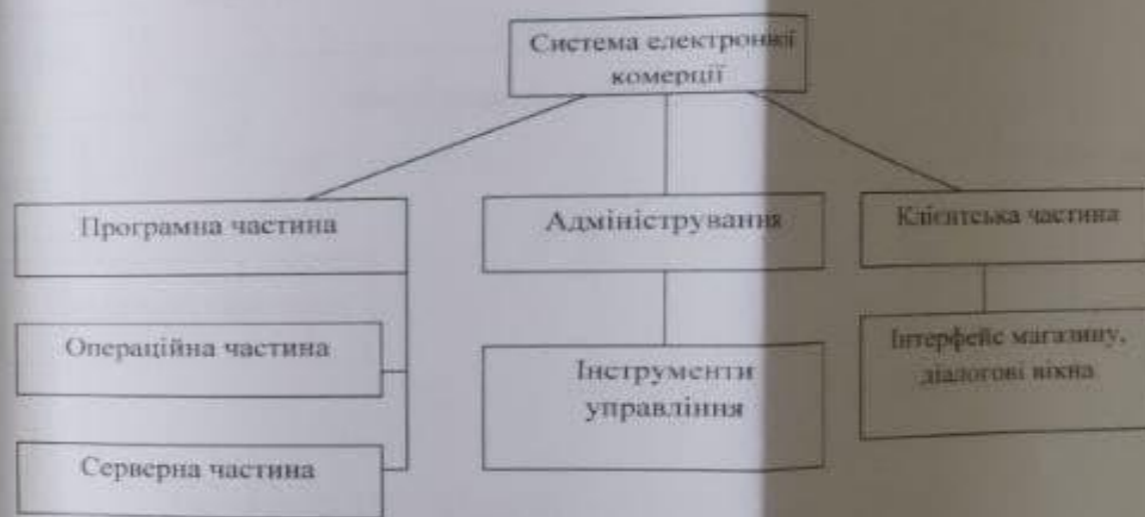


Рисунок 2 - Структура системи електронної комерції

					13.02070849.00057 Пл.3								
№	Лист	№ Докум.	Підп.	Дата	Розробка та просування в Інтернет систем електронної комерції								
Розроб.	Накомечний О.Д.												
Перев.	Кисель М.М.				Структура системи електронної комерції								
І.контр.													
І.контр.	Зелік О.В.				<table border="1" style="width: 100%;"> <tr> <td>Тем</td> <td>Масл</td> <td>Масл</td> </tr> <tr> <td>Листы 1</td> <td>Листы 1</td> <td></td> </tr> </table>			Тем	Масл	Масл	Листы 1	Листы 1	
Тем	Масл	Масл											
Листы 1	Листы 1												
Затв.	Кудерметов Р.К.				НУЗП КНТ-518сп								

# Пошукова оптимізація контенту



Рисунок 1 – Етапи розробки семантичного ядра сайту

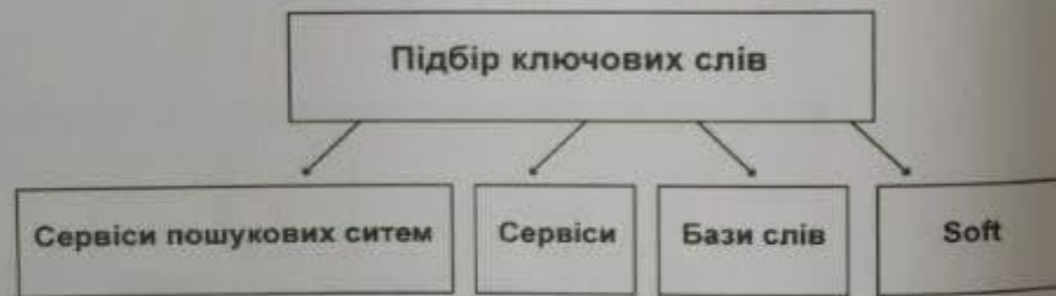


Рисунок 2 – Шляхи підбору ключових слів для SEO

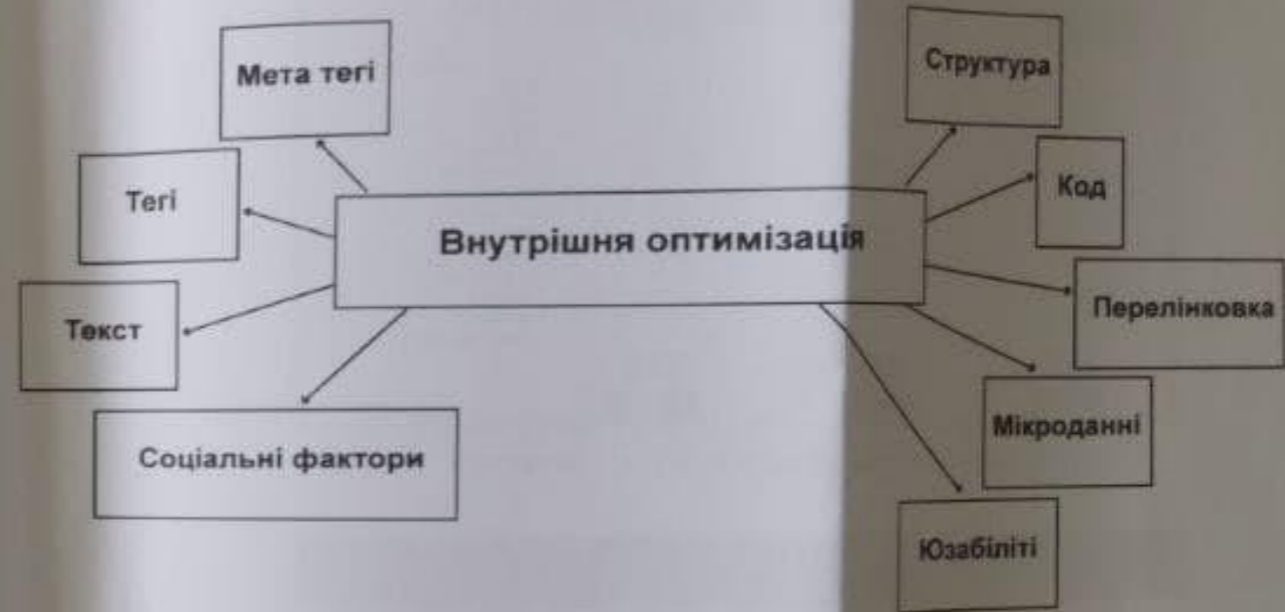


Рисунок 3 – Складові внутрішньої оптимізації сайту

					13.02070849.00057 Пл.4			
№	Лист	№ Докум.	Підп.	Дата	Розробка та просування в інтернет системи електронної комерції	Лист	Вибір	Відомості
Розроб.		Наконечний О.Д.	<i>[Signature]</i>			Лист 2	Лист 2	
Перев.		Костян М.М.	<i>[Signature]</i>		Пошукова оптимізація контенту			
Л.контр.								
Н.контр.		Зеліс О.В.	<i>[Signature]</i>					
Затв.		Кудерметов Р.К.	<i>[Signature]</i>					
						НУЗП КНТ-518сп		

# Інтерфейс системи електронної комерції



Рисунок 1 – Головна сторінка сайту



Рисунок 2 – Категорії та підкатегорії каталогу товарів

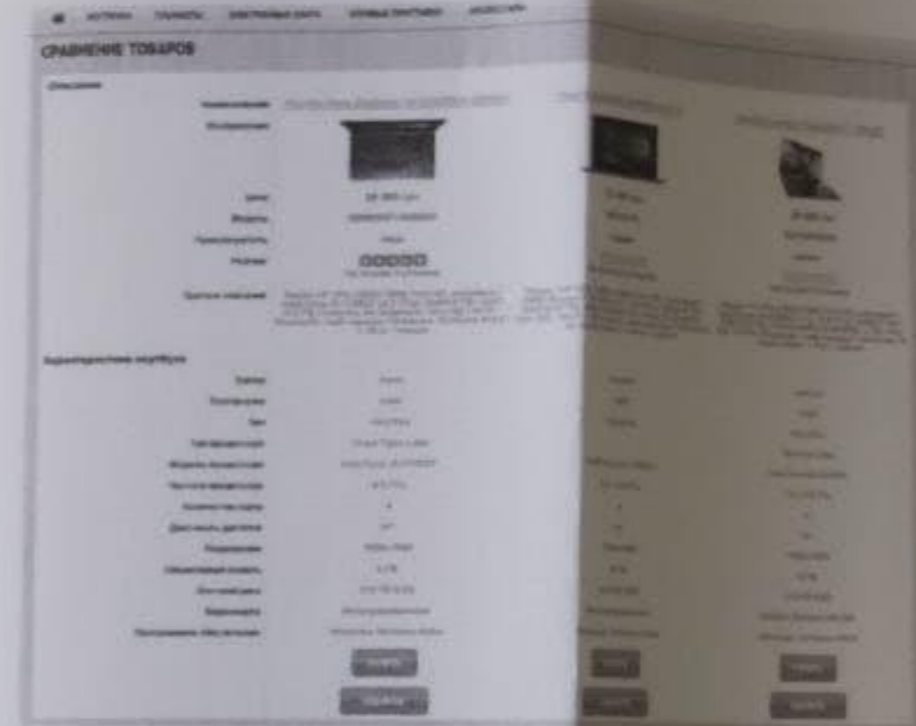


Рисунок 3 – Порівняння обраних товарів

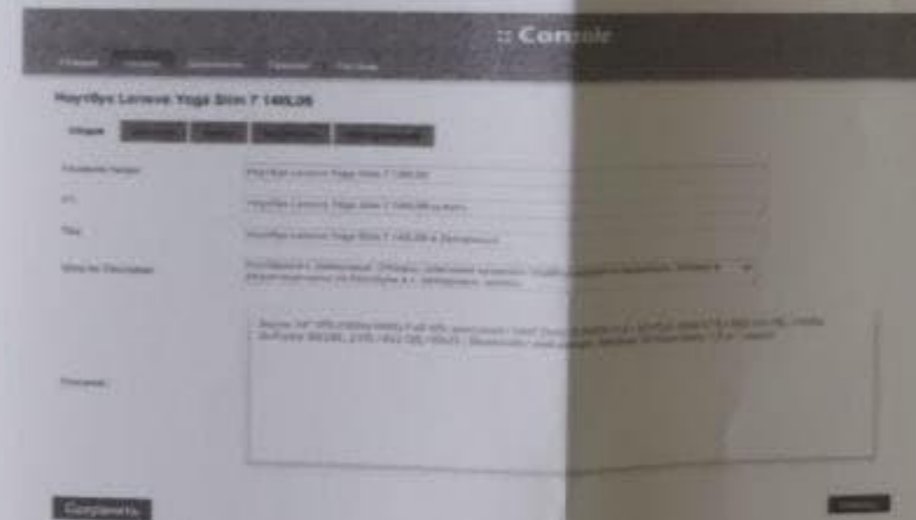


Рисунок 4 – Заповнення опису товару адміністратором

					13.02070849.00057 Пл.5			
№	Лист	№ Докум.	Підп.	Дата	Розробка та просування в інтернет систем електронної комерції	Лист	Місяц	Рік
Розроб.		Наконечний О.Д.	<i>[Signature]</i>			Листы 3		Листы 3
Перев.		Касян М.М.	<i>[Signature]</i>		Інтерфейс сайта электронной комерції			
Т.контр.								
Н.контр.		Зелік О.В.	<i>[Signature]</i>					
Зато.		Кудерніштов Р.К.	<i>[Signature]</i>					
						НУЗП КНТ-518сп		