

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Інститут електроенергетики  
(інститут)

Факультет інформаційних технологій  
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем  
(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**  
**кваліфікаційної роботи ступеня**  
**бакалавра**

(назва освітньо-кваліфікаційного рівня)

студента Половинки Олександра Олеговича  
(ПІБ)

академічної групи 121-18-1  
(шифр)

спеціальності 121 Інженерія програмного забезпечення  
(код і назва спеціальності)

освітньої програми Інженерія програмного забезпечення  
(назва освітньої програми)

на тему: Розробка веб-додатку для Інтернет магазину з продажу персональних комп'ютерів на базі CMS OpenCart

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Спирінцев В.В.			
<b>розділів:</b>				
спеціальний	доц. Спирінцев В.В.			
економічний	доц. Касьяненко Л.В.			
<b>Рецензент</b>				
<b>Нормоконтролер</b>	доц. Гуліна І.Г.			

Дніпро  
2022

Міністерство освіти і науки України  
НТУ «Дніпровська політехніка»

**ЗАТВЕРДЖЕНО:**

завідувач кафедри  
програмного забезпечення комп'ютерних систем

(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

«    »                      2022 року

**ЗАВДАННЯ**

**на кваліфікаційну роботу**

**бакалавра**

(назва освітньо-кваліфікаційного рівня)

студента 121-18-1  
(група)

Половинки Олександра Олександровича  
(прізвище та ініціали)

тема кваліфікаційної роботи

Розробка веб-додатку для Інтернет  
магазину з продажу персональних комп'ютерів на базі CMS OpenCart

затверджена наказом ректора НТУ «ДП» від

№

Розділ	Зміст виконання	Термін виконання
Спеціальний	На основі матеріалів проєктно-технологічної та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми	13.05.2022 р.
Економічний	Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки	27.05.2022 р.

Завдання видав

(підпис)

доц. Спирінцев В.В

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Половинка О.О.

(прізвище, ініціали)

Дата видачі завдання: 14.01.2022 р.

Термін подання кваліфікаційної роботи до ЕК: 13.06.2022 р.

## РЕФЕРАТ

Пояснювальна записка: 93 с., 34 рис., 3 дод., 21 джерел.

Об'єкт розробки: веб-додаток для інтернет магазину з продажу персональних комп'ютерів на базі CMS OpenCart.

Мета кваліфікаційної роботи: створення зручного веб-сайту магазину для реалізації торгівлі через інтернет, і досягнення таким чином певних переваг перед конкурентами через такі фактори як економія часу та грошей, залучення нових клієнтів, оптимізація існуючих торгових операцій та процесів.

У вступі уточняється галузь та мета кваліфікаційної роботи, аналізується сучасна проблематика, конкретизується постановка завдання, обґрунтовується актуальність теми.

Практичне значення розробки: створення веб-сайту, який буде дозволяти формувати замовлення, покупати та продавати комп'ютери, оптимізувати та покращувати торгівлю в мережі інтернет.

Сфера веб-розробки постійно змінюється, аналіз цієї галузі говорить про те що, актуальність розробки веб-сайтів постійно залишається високою через високий попит з боку великого та малого бізнесу.

Список ключових слів: ІНТЕРНЕТ МАГАЗИН, ВЕБ-САЙТ, CMS, ІНТЕРНЕТ, ВЕБ-РОЗРОБКА.

## **ABSTRACT**

Explanatory note: 93 pp., 34 fig., 3 extra, 21 sources.

The object of development: web application for an online store for the sale of personal computers based on OpenCart CMS.

The purpose of the qualification work: to create a convenient website for the store to sell online, and thus achieve certain advantages over competitors through factors such as saving time and money, attracting new customers, optimizing existing trading operations and processes.

The introduction specifies the field and purpose of the qualification work, analyzes the current issues, the statement of the task is specified, the relevance of the topic is substantiated.

The practical value of development: creating a website that will allow you to place orders, choose the method of delivery and payment, buy and sell computers, optimize and improve online commerce.

The field of web development is constantly changing, the analysis of this industry shows that the relevance of website development remains high due to high demand from large and small businesses.

List of keywords: ONLINE STORE, WEBSITE, CMS, INTERNET, WEB DEVELOPMENT.

## ЗМІСТ

РЕФЕРАТ .....	3
ABSTRACT .....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ...	10
1.1. Загальні відомості з предметної галузі .....	10
1.1.1 Веб-розробка та її використання для електронної комерції.....	10
1.1.2 Розробка інтернет-магазину.....	34
1.2. Призначення розробки та галузь застосування.....	36
1.3. Підстава для розробки .....	37
1.4. Постановка завдання.....	38
1.5. Вимоги до програми або програмного виробу .....	39
1.5.1. Вимоги до функціональних характеристик.....	40
1.5.2. Вимоги до інформаційної безпеки .....	41
1.5.3. Вимоги до складу та параметрів технічних засобів .....	43
1.5.4. Вимоги до інформаційної та програмної сумісності.....	44
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	45
2.1. Функціональне призначення програми.....	45
2.2. Опис застосованих математичних методів.....	45
2.3. Опис використаної архітектури та шаблонів проектування.....	45
2.4. Опис використаних технологій та мов програмування.....	48
2.5. Опис структури програми та алгоритмів її функціонування.....	51
2.6. Обґрунтування та організація вхідних та вихідних даних програми .....	58
2.7. Опис розробленого програмного продукту .....	59
2.7.1. Використані технічні засоби .....	59
2.7.2. Використані програмні засоби.....	60
2.7.3. Виклик та завантаження програми .....	61
2.7.4. Опис інтерфейсу користувача.....	61

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....	70
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту .....	70
3.2. Рахунок витрат на створення програми.....	73
ВИСНОВКИ.....	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	78
Додаток А. Код програми.....	82
Додаток Б. Відгук керівника економічного розділу.....	92
Додаток В. Перелік файлів на диску .....	93

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

HTML - Hyper Text Markup Language  
CSS - Cascading Style Sheets  
CMS - Система управління контентом  
СУБД - Система управління базами даних  
БД - База даних  
ПЗ - Програмне забезпечення  
SQL - Structured Query Language  
IDE - Integrated Development Environment  
CRM - customer relationship management  
ІТ - Information Technology  
ПК - Персональний комп'ютер  
WCM - Web Content Management  
HTTP - HyperText Transfer Protocol  
JS - JavaScript  
MVC - Model–view–controller  
DNS - Domain name server

## ВСТУП

Тематика цієї кваліфікаційної роботи присвячена розробці веб-додатку для інтернет магазину з продажу персональних комп'ютерів на базі CMS OpenCart.

Метою даної кваліфікаційної роботи є розробка спеціалізованого інтернет магазину для продажу комп'ютерів у мережі інтернет, реалізація певного складу необхідного функціоналу на веб-сайті.

Останнім часом сфера веб-розробки інтернет магазинів набуває все більшого поширення серед підприємців різного рівня, тому що такий спосіб ведення бізнесу є вигідним для них. Стабільний попит зі сторони ринку на розробку інтернет магазинів породжує пропозицію з боку веб-розробників різного плану. Кількість бажаючих володіти бізнесом в інтернеті зростає щороку. Сучасні технологічні рішення дають їм таку можливість, зараз поріг входу в інтернет-торгівлю став значно менше ніж був раніше. Цей фактор зв'язаний з розповсюдженням систем управління контенту, які дають можливість зменшити витрати на розробку сайту. Цим обумовлений вибір даної тематики кваліфікаційної роботи.

Попри велику кількість існуючих рішень та технологій, які спрощують як розробку, так і підтримку інтернет-магазинів, завжди потребується веб-розробник який буде розробляти сайт під потреби конкретного підприємця або компанії.

Веб-розробник, праця котрого виконується в даній кваліфікаційній роботі, як фахівець в області веб-програмування повинен володіти всіма необхідними знаннями та навичками, серед котрих є і мови програмування, і стандартизованою мовою розмітки документів для перегляду веб-сторінок у браузері (HTML), і формальною мовою опису зовнішнього вигляду документа (CSS). Окрім всього веб-розробник повинен мати досвід роботи з інструментами веб-розробників.

Дана кваліфікаційна робота повністю підпадає під вимоги які ставляться до веб-розробників, як фахівців своєї спеціальності.



У цьому проекті потребують вирішення стандартні для таких робіт проблеми: переписування під проект програмного коду системи, дороблення необхідного функціоналу, розробка клієнтської частини сайту (фронтенду), встановлення необхідних модулів, та інше.

В процесі розробки інтернет магазину треба визначити те, який саме функціонал потрібен цьому сайту, а від якого треба відмовитись, де і які елементи системи потребують редагування та корекції, як саме повинна виглядати клієнтська частина, що треба доопрацювати. Розроблений сайт повинен мати зручний та інтуїтивно зрозумілий інтерфейс.

Потенційними користувачами даного веб-додатку є підприємці, які займаються роздрібною торгівлею персональними комп'ютерами не використовуючи інтернет, та хочуть покращити та розширити свій бізнес. Або ж люди які тільки планують стати підприємцями та відкрити свій магазин в інтернеті, та спробувати себе в цій галузі, і саме для цього вони і потребують сайт.

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

#### 1.1. Загальні відомості з предметної галузі

##### 1.1.1 Веб-розробка та її використання для електронної комерції

«Веб-розробка – це кодування або програмування, яке дозволяє розробляти програми для публікації у веб-браузері відповідно до вимог проблеми. Існують дві важливі частини веб-розробки, а саме розробка фронтенду (також називається розробкою клієнта) і розробка бекенду (також відома як серверна розробка). [1]»

Веб-розробка зародилася ще наприкінці 1980-х, коли Тім Бернерс-Лі, який був одним із співробітників лабораторії CERN, запропонував ідею деякої комп'ютерної системи, яка, на його думку, повинна була допомогти вченим та дослідникам у різних частинах світу комунікувати та співпрацювати між собою.

Це призвело до того, що вже в 1993 році на світ з'явився всім відомий зараз HTML, він став і залишається основою в плані будівництва та роботи всесвітньої мережі.

До наших днів веб-розробка пройшла довгий шлях еволюціонує та розвиваючись, зараз це багатогранна та складна сфера, яка не стоїть на місці.

З прикладної точки зору веб-розробка займається розробкою веб-сайтів.

«Web-сайт — це група пов'язаних веб-сторінок, яку розташовано на сервері HTTP в Інтернеті. У більшості Web-сайтів початком є домашня, або головна, сторінка. Домашня сторінка пов'язана з іншими сторінками за допомогою гіперпосилань.

Можна використовувати сайти верхнього рівня (вузол верхнього рівня. Web-сайт на початку ієрархії родини вузлів, з якого можна керувати всіма можливостями родини вузлів. Вузол верхнього рівня може мати кілька підвузлів) та дочірні сайти (підвузол. Повнофункціональний Web-сайт, який зберігається в

іменованому підкаталогу Web-сайту верхнього рівня. Права адміністрування, редагування та перегляду підвузла можуть не залежати від вузла верхнього рівня та інших підвузлів), щоб поділити вміст на чіткі, окремо керовані сайти.

Web-сайти верхнього рівня можуть мати кілька підсайтів, які у свою чергу можуть мати власні підсайти. Всю структуру Web-сайту верхнього рівня та його підсайти називають колекцією сайтів. [2]»

Якщо як небудь категоризувати сайти, що розробляються, по типу формування наповнення, то можна сказати що існують два види сайтів, це статичні та динамічні.

Статичний сайт – це одна або певна група незмінних HTML сторінок, велика або ні, які мають увесь потрібний код, що необхідний для того, щоб браузері їх нормально відобразили.

Такий вид сайту має ряд позитивних рис:

- простий процес створення сторінок;
- швидкість завантаження сайту;
- невелике навантаження на сервер;
- економність використання;
- легкість перенесення на інший сервер.

Значний мінус цьому випадку – це певні труднощі з оновленням статичних сайтів. Великий сайт буде потребувати дуже багато роботи. Динамічний сайт – це сайт основою якого є програмне ядро. Він складається із динамічних сторінок які змінюються. Програмне забезпечення дозволяє швидко та зручно (на відміну від статичного сайту) маніпулювати вмістом сайту.

Цей вид сайту також має певні позитивні риси:

- можливість самостійно корегувати сайт та вносити зміни, без допомоги фахівців;
- наявність великої кількості функціональних можливостей;
- простота в плані верстки сайту та його адміністрування;
- швидке відображення нововведених даних.

Але є й ряд недоліків:

- труднощі при перенесенні на інший хостинг;
- велике навантаження на сервер;
- більші в порівнянні зі статичним сайтом фінансові втрати.

Один із основних принципів роботи мережі Інтернет – це клієнт-серверна архітектура. Взагалі будь-який додаток, або веб-сайт в інтернеті працює на сервері, а користувачі являються клієнтами.

«Клієнт — сервер (англ. client–server) (рис.1.1) — обчислювальна або мережева архітектура, в якій завдання або мережеве навантаження розподілені між постачальниками послуг, які називаються серверами, та замовниками послуг, які називають клієнтами. Фактично клієнт та сервер – це програмне забезпечення. Зазвичай ці програми розташовані на різних обчислювальних машинах і взаємодіють між собою через обчислювальну мережу за допомогою мережевих протоколів, але можуть бути розташовані також і на одній машині. [3]»

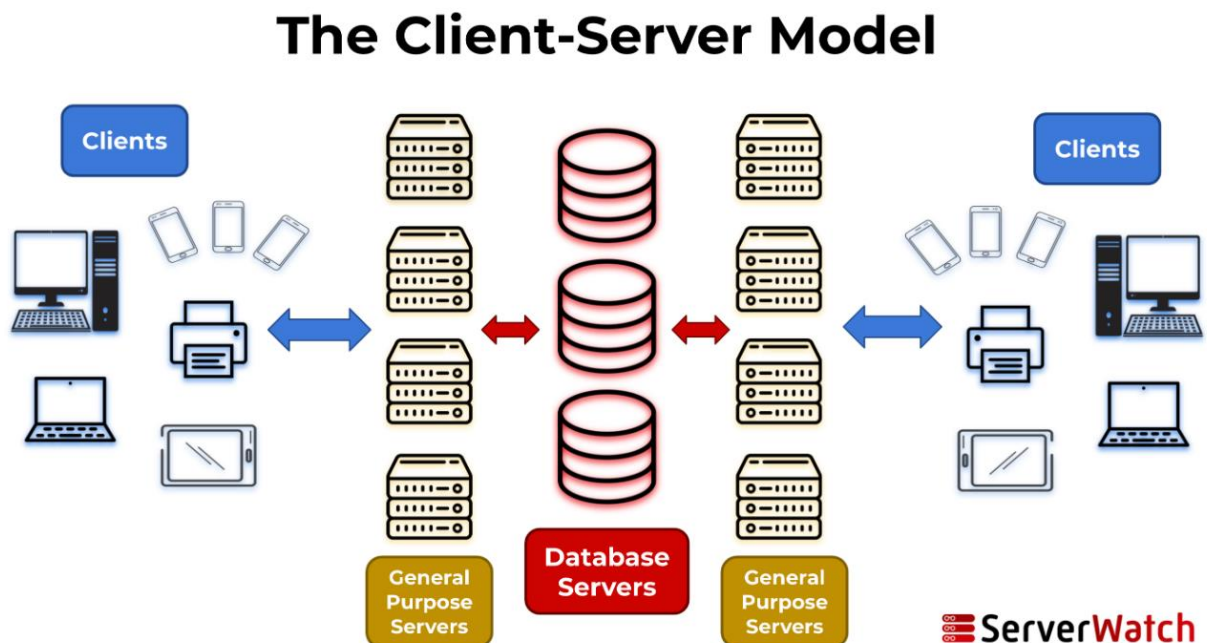


Рис. 1.1. Приклад клієнт-серверної архітектури

В цій архітектурі якась кількість комп'ютерів-клієнтів посилають запити, а потім отримують послуги від сервера.

Клієнтська частина системи займається тим, що надає користувачу зручний та зрозумілий інтерфейс для того, щоб полегшити взаємодію користувача з сервером.

Клієнт-серверну архітектуру взагалі прийнято умовно розділяти на три окремі класи: однорівневу, дворівневу та трирівневу.

Однорівнева клієнт-серверна архітектура (1-Tier) (рис.1.2) – в такій системі усі прикладні програми знаходяться на самих робочих станціях, які в свою чергу використовують спільний файловий сервер або спільний сервер баз даних. Взагалі ніяких прикладних програм сервер не має у своєму функціоналі, він лише надає дані.

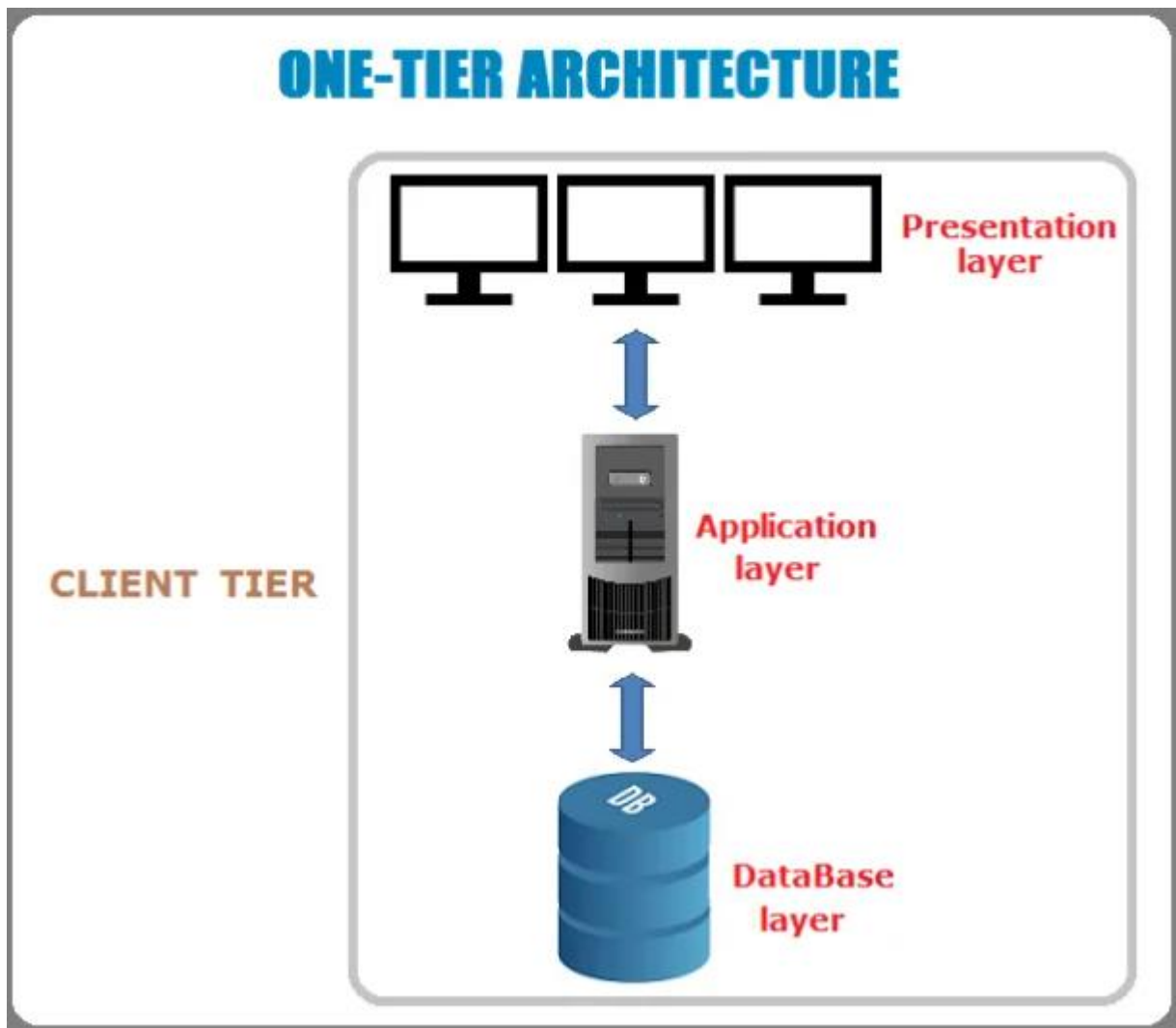


Рис. 1.2. Приклад однорівневої клієнт-серверної архітектури

Дворівнева клієнт-серверна архітектура (2-Tier) (рис.1.3) – в такій прикладні програми розташовані на сервері додатків, як приклад можна привести сервер CRM або сервер 1С, а в самих робочих станціях розташована клієнтська частина, яка в свою чергу надає користувачам потрібний інтерфейс для роботи з тими програмами, що знаходяться на загальному сервері.

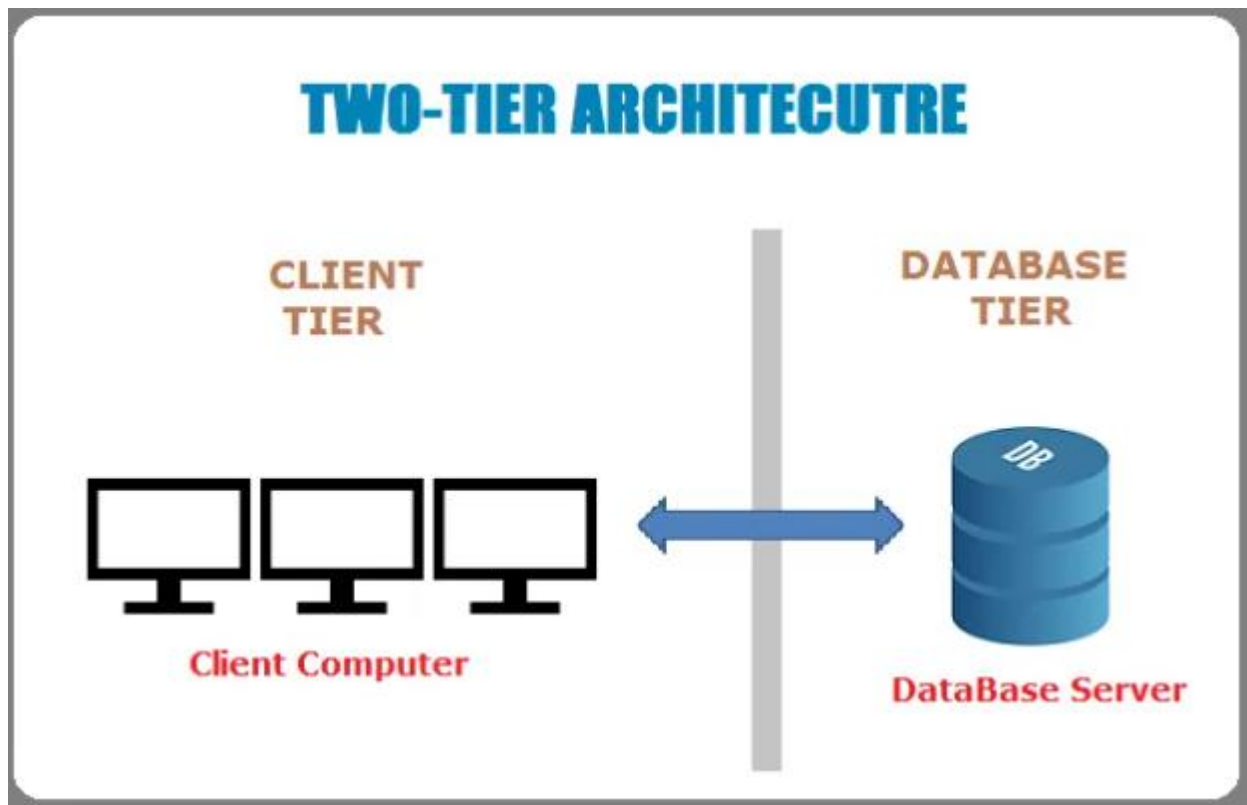


Рис. 1.3. Приклад дворівневої клієнт-серверної архітектури

Така клієнт-серверна архітектура є найбільш логічною, але в ній можна виділити два різні варіанти. Перший варіант – загальні дані розташовуються на сервері, а бізнес-дані та логіка обробки даних зберігаються на клієнтській частині, у такому випадку архітектура має назву “fat client thin server” (товстий клієнт, тонкий сервер). В тому випадку якщо не тільки дані, а ще й логіка їх обробки та бізнес-дані знаходяться саме на сервері, така архітектура має назву “thin client fat server” (тонкий клієнт, товстий сервер). Саме така архітектура послужила прообразом хмарних обчислень.

Дворівнева архітектура має такі переваги:

- зазвичай користувачеві легко працювати в такій системі;
- гарна продуктивність;
- непогана масштабованість;
- легкість налаштування додатку;
- легко модифікувати додаток.

Дворівнева архітектура має і певні недоліки:

- всі клієнти залежать від однієї бази даних;
- через те, що всі програми та дані знаходяться на центральному сервері, можуть бути проблеми з безпекою;
- через збільшення кількості користувачів продуктивність може сильно впасти.

Трирівнева клієнт-серверна архітектура (3-Tier) (рис.1.4) – при такій архітектурі файловий сервер, сервер баз даних та інші являються окремим рівнем, роботу якого в кінцевому підсумку використовує сервер додатків. бізнес-логіка та логіка даних розташовані саме на сервері додатків. Абсолютно всі звернення з боку клієнтів до бази даних проходять через проміжне програмне забезпечення. Як наслідок цього, зростає продуктивність та гнучкість роботи.

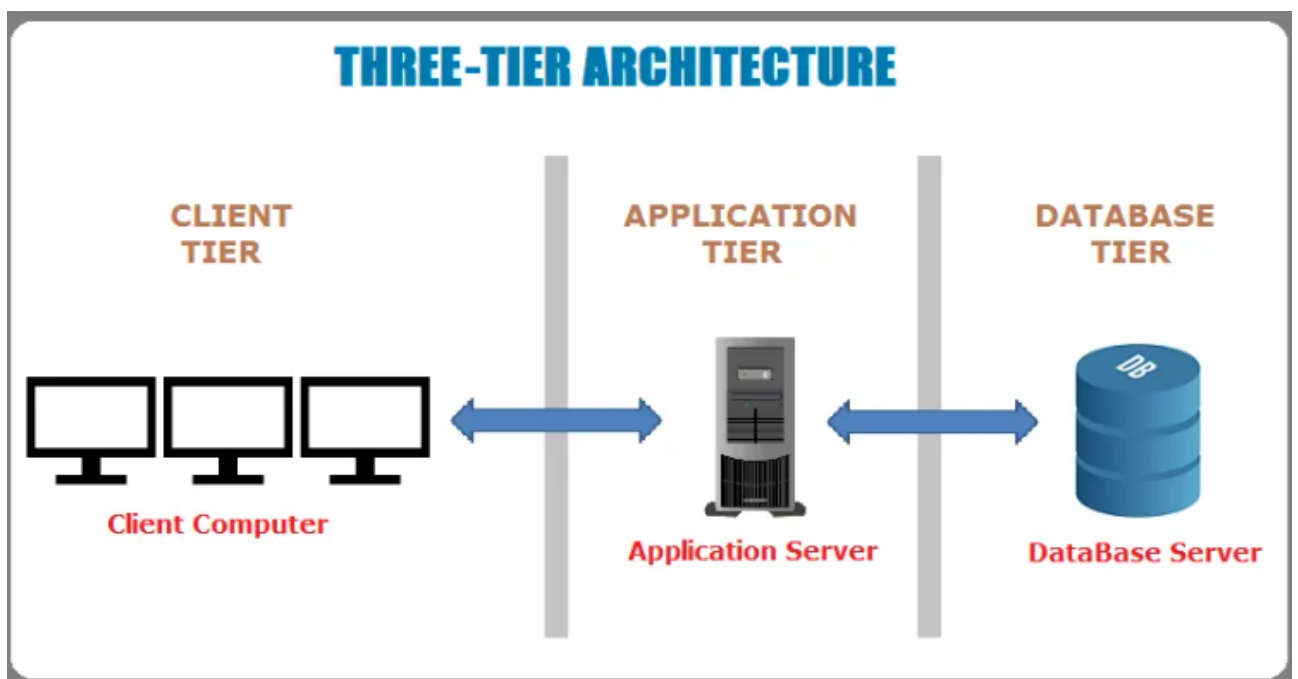


Рис. 1.4. Приклад трирівневої клієнт-серверної архітектури

Трирівнева архітектура має такі переваги:

- в порівнянні з дворівневою архітектурою має кращу безпеку;
- цілісність даних;
- захищеність від несанкціонованого проникнення бази даних.



Як мінус можна виділити більш складну структуру комунікацій між сервером та клієнтами.

Багаторівнева клієнт-серверна архітектура (N-Tier) (рис.1.5) – при такій декілька серверів додатків користуються результатами роботи один одного, а також даними від різних файлових серверів, серверів баз даних та усіх інших видів серверів.

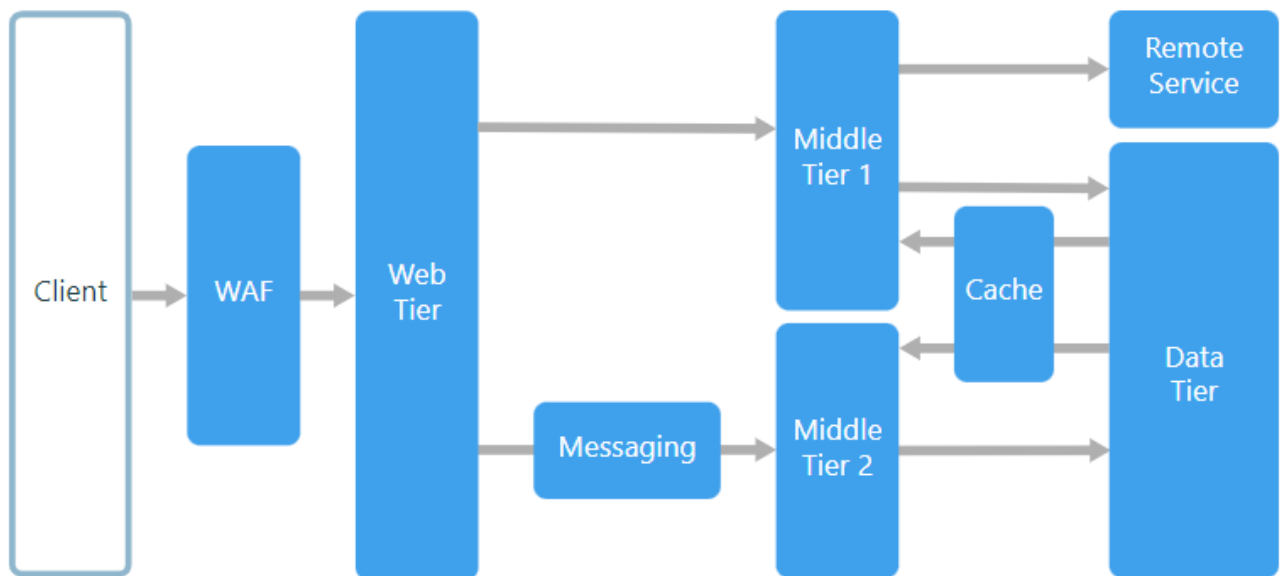


Рис. 1.5. Приклад багаторівневої клієнт-серверної архітектури

Значною перевагою такої архітектури являється гнучкість щодо надання послуг, які в свою чергу можуть бути скомбіновані з результатів роботи різних програм серверів різних рівнів.

Явним недоліком в такому випадку є велика кількість компонентів та складність такої архітектури.

Характеристики клієнт-серверної архітектури:

- Інкапсуляція послуг. Після того як сервер отримав запит від клієнта він вирішує, як саме повинен бути виконаний цей запит. Модифікувати сервер можна без впливу на роботу клієнтської частини, тому що це ніяк не впливає на інтерфейс взаємодії між ними. Якщо перефразувати, то максимум, що при цьому можливо відчує конкретний користувач – це

- незначна затримка відгуку з боку сервера протягом невеликого часу, коли відбувається модифікація.
- Асиметричність протоколів. Між сервером та клієнтами існують стосунки по принципу – один до багатьох. Діалог із сервером зазвичай ініціює клієнт.
  - Місцева прозорість. Сервер – це певний програмний процес, який може виконуватися на тій же самій машині, на котрій виконується і клієнт або на іншій машині, яка підключена до мережі. Програмне забезпечення клієнт-сервер як правило приховує від клієнтів розташування сервера, тим що перенаправляє запит через мережу.
  - Цілісність. Загальні дані та програми для сервера мають централізовану систему управління, яка захищає цілісність даних та зменшує вартість обслуговування. У той же самий час клієнтські дані в свою чергу залишаються незалежними і персоніфікованими.
  - Обмін на основі повідомлень. Сервер та клієнти є не жорстко пов'язаними один з одним процесами, які між собою обмінюються повідомленнями: запитами на послуги та власне відповідями на них.
  - Незалежність від платформи. Досконала клієнт-серверна програма не залежить від операційної системи або платформ обладнання. Сервери та клієнти можуть бути розгорнутими на взагалі різних операційних системах та платформах.
  - Модульний дизайн, який є можливість розширити. Модульний дизайн клієнт-серверної архітектури надає їй певної стійкості щодо відмови, тобто, якщо якийсь модуль відмовив це не буде викликати відмови абсолютно всього додатка. Така система зроблена так, що декілька серверів можуть відмовити без відмови всієї системи взагалі, в цей час послуги сервера, який відмовив, будуть надані з іншого резервного сервера. Ще одна перевага модульної системи в тому, що клієнт-сервер має можливість реагувати в автоматичному режимі на зниження або

підвищення навантаження в системі за допомогою відключення або додавання серверів та послуг.

- Загальне використання ресурсів. Один сервер має можливість водночас надавати послуги великій кількості клієнтів, а також займатися регулюванням їх доступу до тих ресурсів які використовуються спільно.
- Розподіл функціональної частини. Клієнт-серверна архітектура – це співвідношення між процесами, які працюють на декількох різних машинах, або на одній. Клієнт – це споживач послуг, а сервер в свою чергу – це процес надання послуг.
- Масштабованість. Клієнт-серверна архітектура може масштабуватися або за кількістю клієнтів та серверів або за спектром послуг та продуктивністю.

На практиці будь-яка ІТ-система підприємства або корпоративна мережа, зазвичай, побудована на базі клієнт-серверної архітектури. В невеликих мережах один із декількох робочих комп'ютерів може бути використаний в якості сервера. Якщо в організації кількість машин більше ніж 10, то кращим варіантом буде зробити виділений сервер, який в свою чергу буде займатися обслуговуванням клієнтів – телефонів та комп'ютерів співробітників цієї організації.

Клієнт-серверна архітектура також досить часто використовується і в домашніх мережах. Як приклад, у такому випадку до мережі можуть бути підключені комп'ютери членів однієї сім'ї, а функції сервера візьме на себе один з цих комп'ютерів.

Але у клієнт-серверної архітектури є й ряд недоліків, до них слід зарахувати:

- Перевищення меж ресурсів які є у сервера, в такому випадку нові клієнти, що хочуть скористатися послугою, залишаються без обслуговування. В такій ситуація потрібно терміново розширювати серверні ресурси.

- Наявність у маленьких мережах із одним єдиним сервером єдиної точки відмови. У випадку якщо сервер відмовляє, тоді без обслуговування залишаються всі клієнти цієї мережі.
- Перевантаження трафіку в мережі – в мережах клієнт-сервер це головна проблема. Коли водночас забагато клієнтів запитують на сервері одну послугу. Дуже велика кількість запитів може викликати перевантаження мережі.
- Іноді на терміналах користувачів клієнтська програма може відмовити з тої причини, що на сервері необхідні драйвери просто не встановлені.

Клієнт-серверна архітектура зараз – це основа абсолютної більшості корпоративних мереж, вона бере свій початок ще з перших обчислювальних машин. Програмне забезпечення комп'ютерних мереж, велика кількість котрих побудована на клієнт-серверній архітектурі, почало розроблятися приблизно п'ятдесят років тому назад.

«Різні види сайтів у тих чи інших комбінаціях існують в англomовному віртуальному просторі. У ньому поєднуються інформативна, аргументативна, розважальна стратегії на медійних, торгових, рекламних, корпоративних, інституційних сайтах і порталах, які мають утилітарну спрямованість контентів. [4]»

За типом контенту сайти можна умовно категоризувати приблизно так:

- Сервіси потокового відео або телебачення. Розповсюдженість та популярність сайтів такого типу зростає останнім часом. Деякі платформи по типу “Netflix” (рис.1.6) повністю переосмислили концепцію щодо способів подачі фільмів, серіалів та телебачення у абсолютно всьому світі.
- Соціальні медіа. Щодо соціальних мереж, які являються підвидом соціальних медіа, то вони зараз мають неповторні за наповненням та функціоналом веб-сайти, а соціальні медіа на даний момент дозволяють користувачам обмінюватись різними типами контенту серед яких є фото та відео. На даний момент соціальні медіа захоплюють навіть ринок

- навчання та новин, так як такі платформи є більш комфортними та корисними для користувачів. Як приклад можна привести сайт “YouTube” (рис.1.7).
- Сайти новин. Головною метою такого типу сайтів є інформування своїх користувачів про останні події та зміни у світі, або по конкретній темі, яка цікавить користувачів. Як приклад можна привести сайт “CNN” (рис.1.8), який є найпопулярнішим у світі сайтом новин.
  - Освітні ресурси. Такі типи сайтів мають своєю метою допомогу в наданні користувачам тієї інформації, яка їм необхідна. Як приклад можна привести такі сайти як “Стековерфлоу” (рис.1.9) та “Вікіпедія”, які є платформами для надання необхідної інформації тим хто цього потребує.
  - Сайти електронної комерції. Такий тип сайтів існує для того, щоб продавати продукти та послуги всередині мережі Інтернет. В деяких випадках цей тип сайтів може поєднуватись з іншими типами в якихось формах. Як приклад можна привести сайт “ОЛХ” (рис.1.10), що користується великою популярністю в Україні.
  - Корпоративні ресурси. В сучасному світі майже кожна компанія або організація освідомили той факт, що вони повинні мати хоча б простий сайт для того, щоб виглядати в очах потенційних клієнтів або партнерів достатньо професійно та заслуговувати їх довіру. Як приклад можна привести сайт “Карбон КНС” (рис.1.11).
  - Сайти блогів. Такий тип сайтів має оновлюватися на регулярній основі. На практиці такий сайт веде одна людина або певна група людей, яка є не дуже великою. Блог зазвичай пишеться в розмовному або неформальному стилі, він може освітлювати абсолютно будь-яку тематику. Як приклад можна привести кулінарний блог (рис.1.12).
  - Краудфандингові сайти. Краудфандинг в мережі займається тим, що залучає велику кількість окремих людей до фінансування певних підприємств або проектів. В цьому випадку той факт, що суми

пожертвувань малі компенсується їх кількістю. Ті люди які зацікавлені нагодою профінансувати певний проект, мають можливість потім розраховувати на певні бонуси, якщо такий варіант передбачений. Або фінансування може проходити виключно на безкорисній основі. Як приклад можна привести досить популярний “Kickstarter” (рис.1.13).

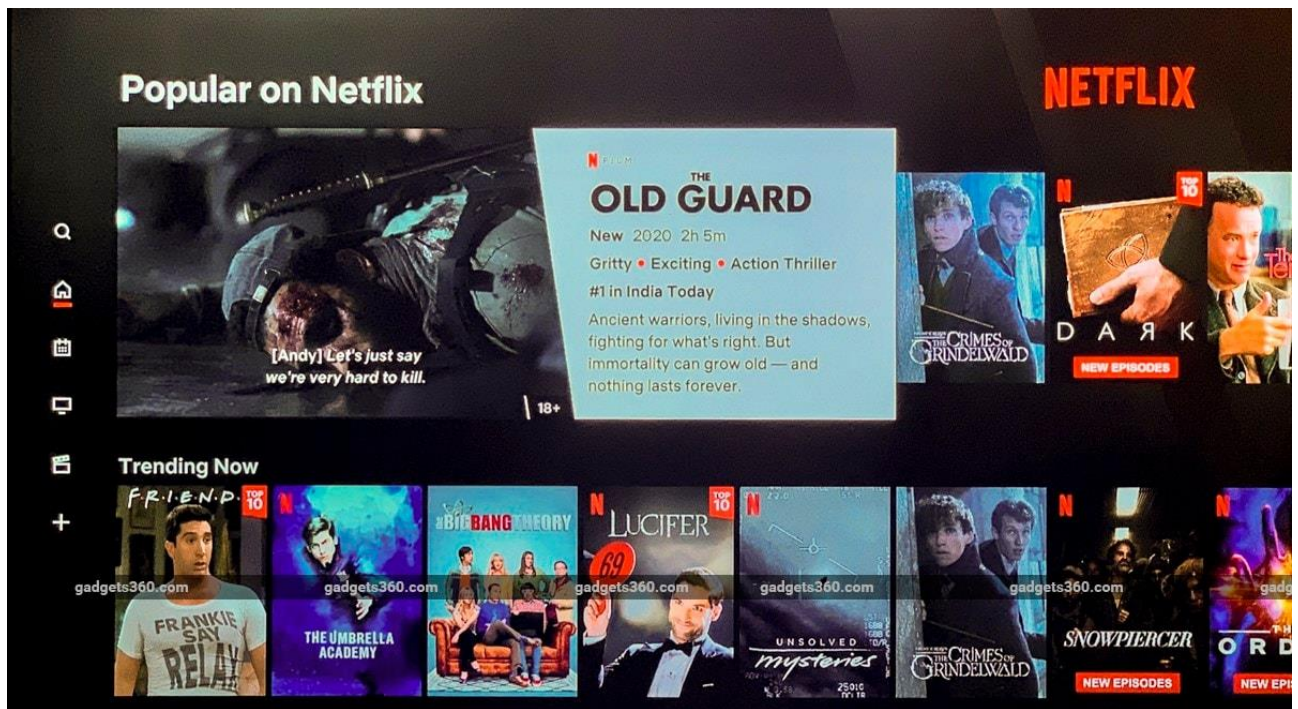


Рис. 1.6. Сервіс потокового відео на прикладі платформи “Netflix”

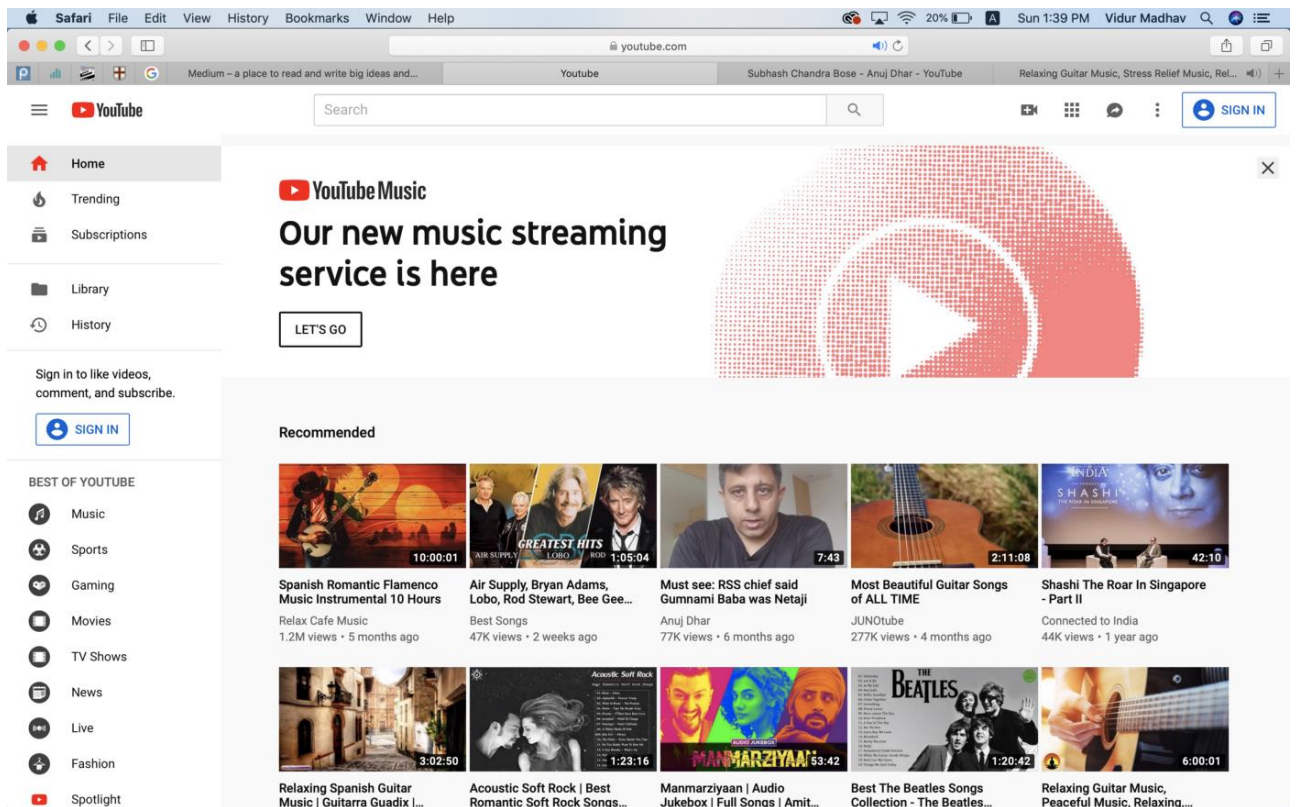


Рис. 1.7. Веб-сайт “YouTube”

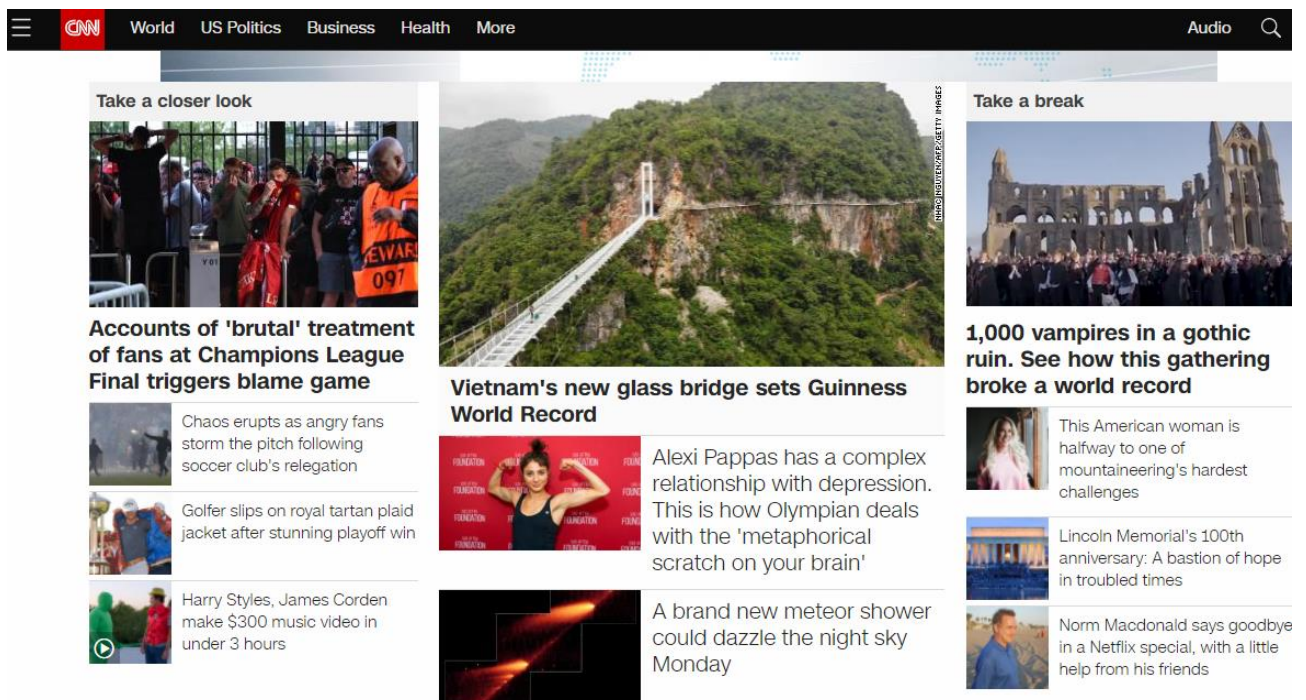


Рис. 1.8. Веб-сайт “CNN”

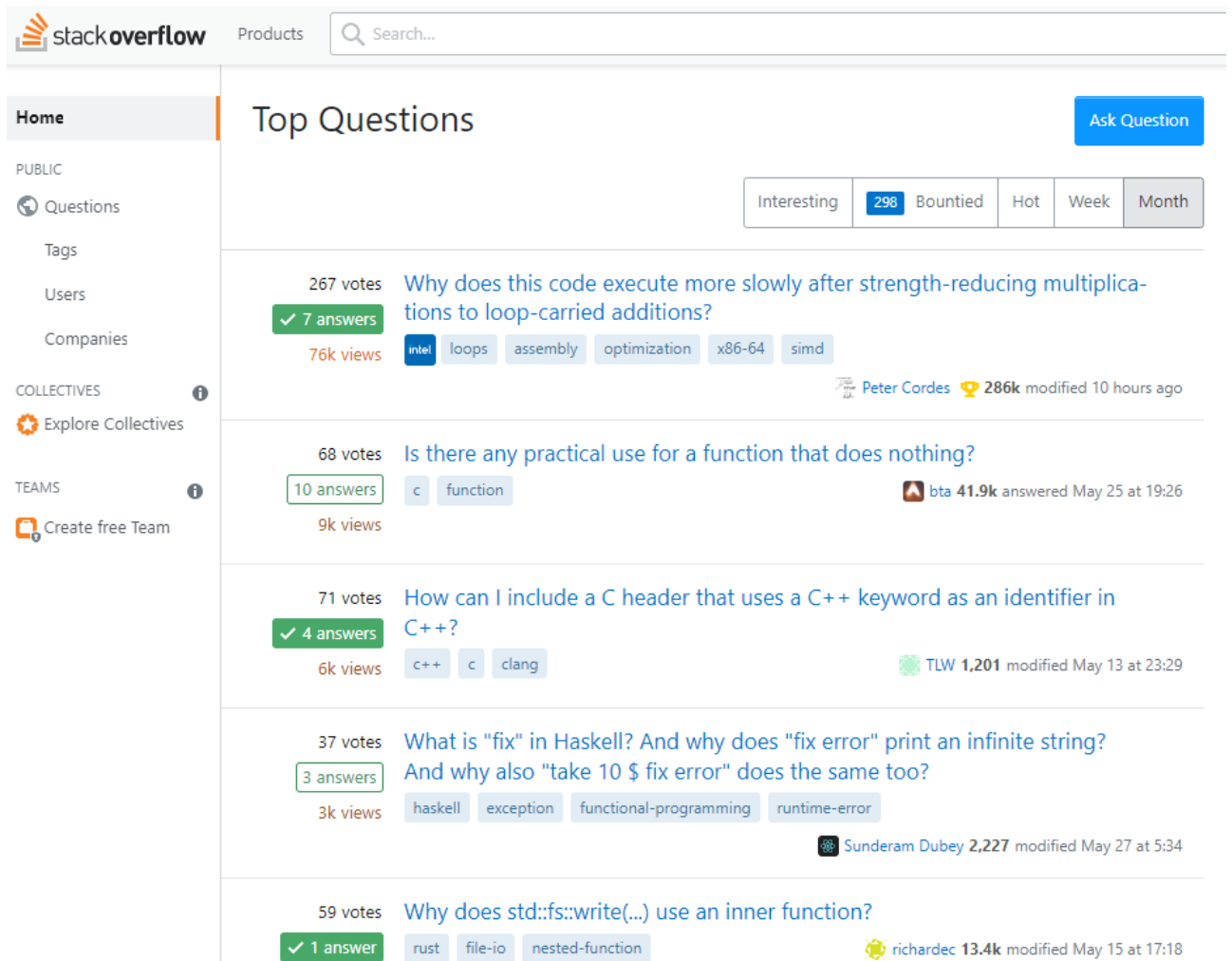


Рис. 1.9. Веб-сайт “Стековерфлоу”

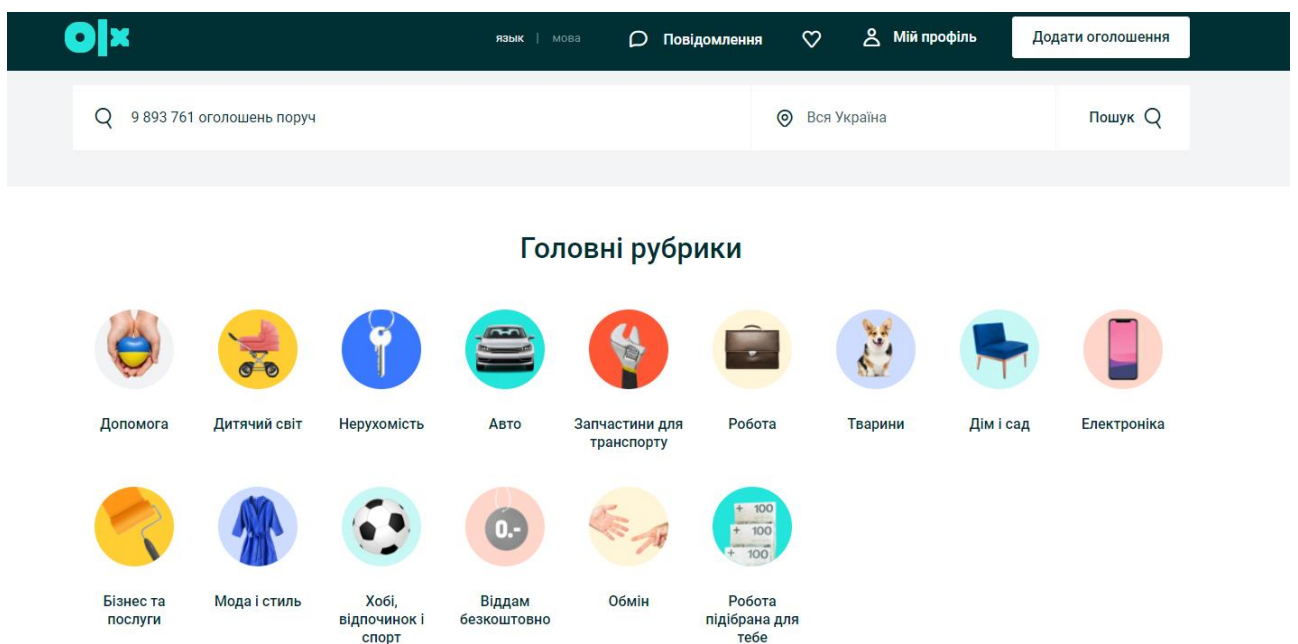


Рис. 1.10. Веб-сайт “ОЛХ”



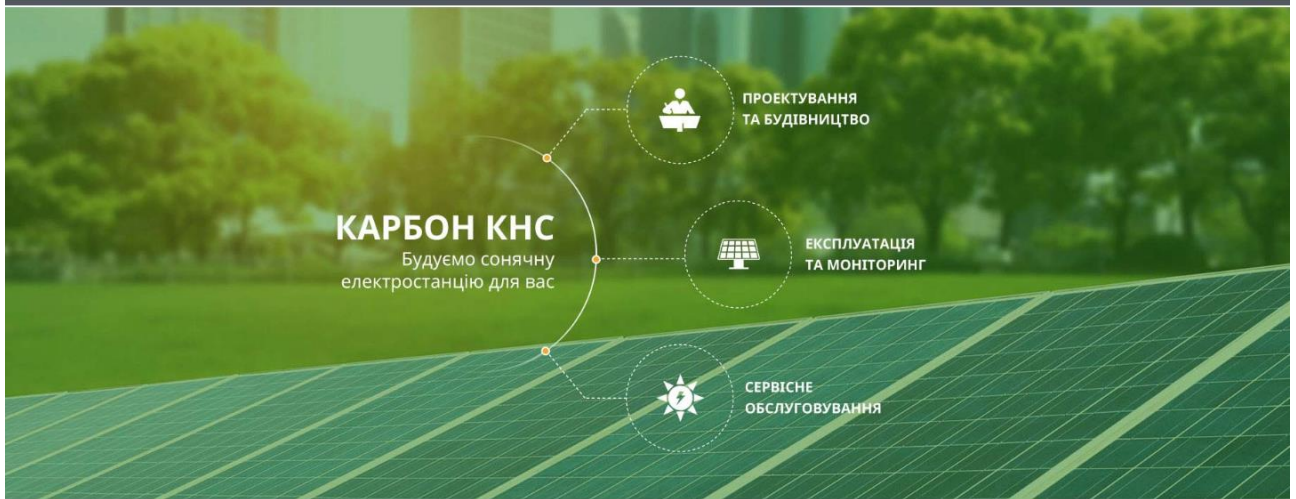


Рис. 1.11. Корпоративний веб-сайт “Карбон КНС”

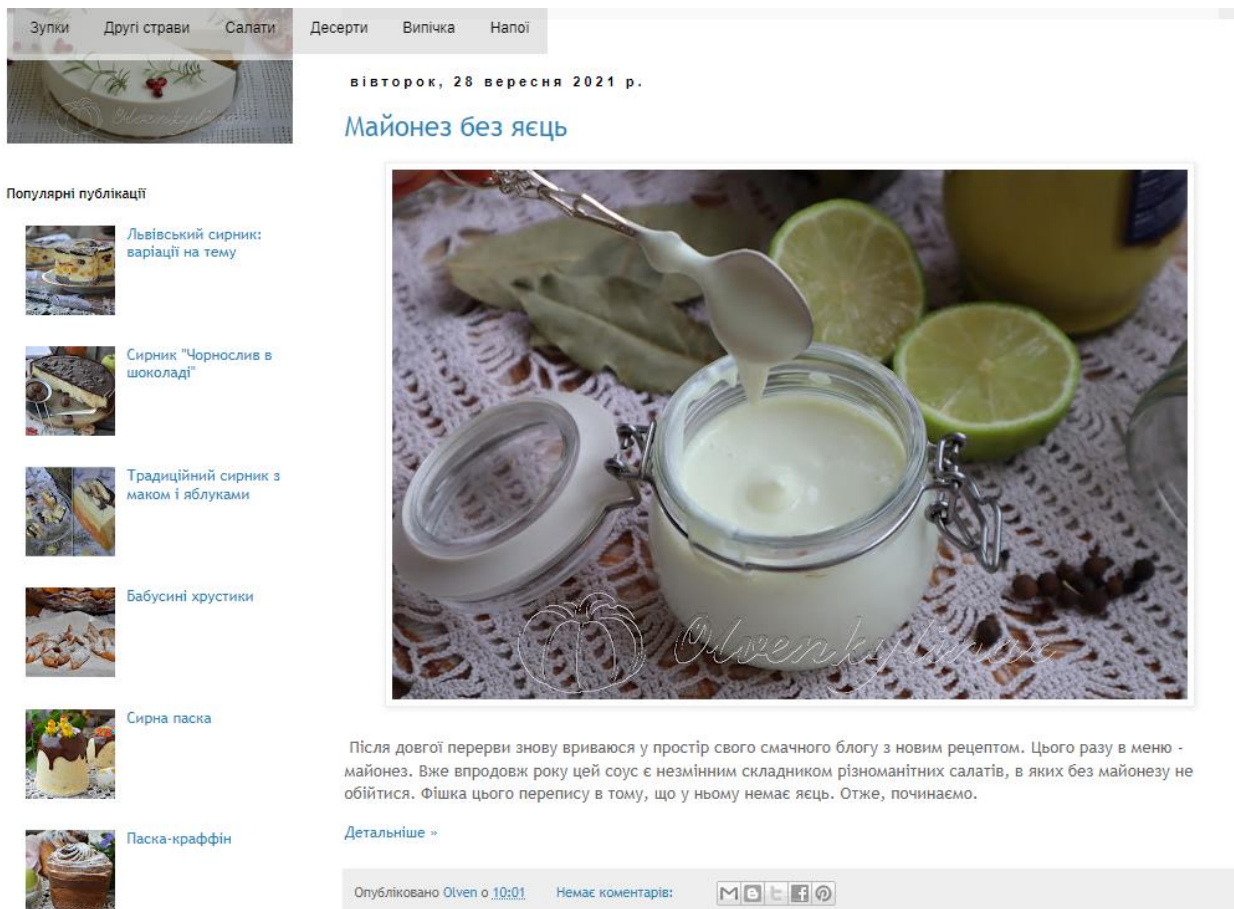


Рис. 1.12. Веб-сайт кулінарного блогу

Arts Comics & Illustration Design & Tech Film Food & Craft Games Music Publishing

Creative work shows us what's possible.  
Help fund it here.

WITHIN THE LAST DAY

38 projects funded	\$1,189,424 towards creative work	10,792 backings
-----------------------	--------------------------------------	--------------------

FEATURED PROJECT

Red Pine: *Dancing With The Dead*

RECOMMENDED FOR YOU

Bark Avenue  
188% funded  
By TerreDice Games

Je t'aime, Julie: A Julie D'Aubigny Story.  
40% funded  
By Fiona Nova

Рис. 1.13. Веб-сайт “Kickstarter”

Якщо категоризувати сайти за рівнем оптимізації для різних платформ та пристроїв, то можна виділити 4 типи шаблонів:

- Адаптивний макет. При такому варіанті дизайну макет дізнається характеристики екрану платформи на якій відображається, а потім використовуючи медіа-запити пристосовується відповідним чином.
- Статичний макет. При такому варіанті не має ніякої оптимізації макету для всіх екранів крім екрану базового ПК. Такий тип сайтів стає менш популярним з кожним роком по причині того, що кількість користувачів інтернету, які користуються мережею зі смартфонів або з планшетів, вже перевищує кількість користувачів зі стандартного ПК.
- Рідкий макет. Такий макет, створений по принципу рідкого дизайну буде відображатися так, щоб зберегти свої пропорції. Абсолютно кожен блок на сайті, незалежно від характеристик екрану, відображається в тих відсоткових пропорціях, які задані.
- Чуйний макет. Значним плюсом такого типу сайту являється те, що при такому підході водночас використовуються і медіа-запити щодо ширини екрана, і відсоткові величини, як у рідкому макеті. В такому

випадку реалізується коректне відображення інформації та плавна зміна. Цей тип макету являється найдосконалішим на даний момент.

Якщо говорити про реалізований в даній кваліфікаційній роботі проект, то він відноситься більше до адаптивного макету сайту електронної комерції, який по суті являється інтернет-магазином.

«Електронна комерція (e-commerce) – вид електронної комерційної діяльності з використанням інформаційних комунікаційних технологій. [5]»

Категоризувати види електронної комерції можна за сторонами які комунікують один з одним:

- С2А – це тип електронної комерції в якому якась адміністрація взаємодіє з клієнтами. Як приклад можна привести електронну охорону здоров'я, або формування та оплату електронних податкових декларацій. Поки що такий тип електронної комерції не користується великим попитом.
- В2А – це тип електронної комерції в якому деяке підприємство продає свої послуги або товари адміністрації. Як приклад можна привести якусь компанію яка продає свої послуги з обслуговування державним сайтам. Також у цю категорію можна включити державні закупівлі в мережі інтернет.
- С2В – це тип електронної комерції в якому деякий клієнт продає свої товари або послуги бізнесу. При такому варіанті клієнти мають можливість диктувати підприємству яке зацікавлене в товарах чи послугах яке надає клієнт. Як приклад можна привести розробників-фрілансерів.
- С2С – це тип електронної комерції в якому один клієнт продає товари або надає послуги іншому клієнту. Як приклад можна привести вищезгадану платформу ОЛХ, на якій користувачі мають можливість виставити на продаж будь-який товар або послугу. Недоліком такої системи є відсутність можливості масштабуватися. Якщо певний

продавець має бажання збільшити свої продажі, йому необхідно переходити до іншого типу електронної комерції.

- B2C – це тип електронної комерції в якому бізнес надає послуги або продає товар клієнту, на практиці це найпоширеніший тип електронної комерції у світі. За допомогою такого типу електронної комерції є можливість масштабувати свій бізнес до вкрай великих величин, деякі компанії такого типу можуть охоплювати мільярди клієнтів.
- B2B – це тип електронної комерції в якому бізнес надає послуги або продає товар іншому бізнесу. Як приклад можна привести торгівлю між собою великих технологічних гігантів, таких як Apple та Sony, коли Sony займалася виробництвом для Apple камер, що встановлювалися на їхню кінцеву продукцію.

«Електронна комерція охоплює відносини управління персоналом, оформлення, виконання та оплати замовлень; співпрацю з постачальниками, фінансовими установами, державними та місцевими органами влади. [6]»

На даний момент розповсюджений доступ до інтернету вирівняв ігрове поле роздрібної торгівлі, дозволяючи певним особам і підприємствам продавати свої товари не беручи до уваги регіональні умови.

«Останніми роками в усьому світі все більшого поширення набувають методи ведення бізнесу в Інтернеті, зокрема. інтернет-торгівля. Застосування Інтернету дозволяє швидко і з незначними витратами вивести і просувати продукцію на національний і міжнародні ринки. [7]»

Відбулася зміна попиту зі сторони ринку, він став більш спрямованим на електронну комерцію. Світова Криза спричинена ковідом дуже сильно прискорила процес поширення електронної комерції у всьому світі (рис.1.14).

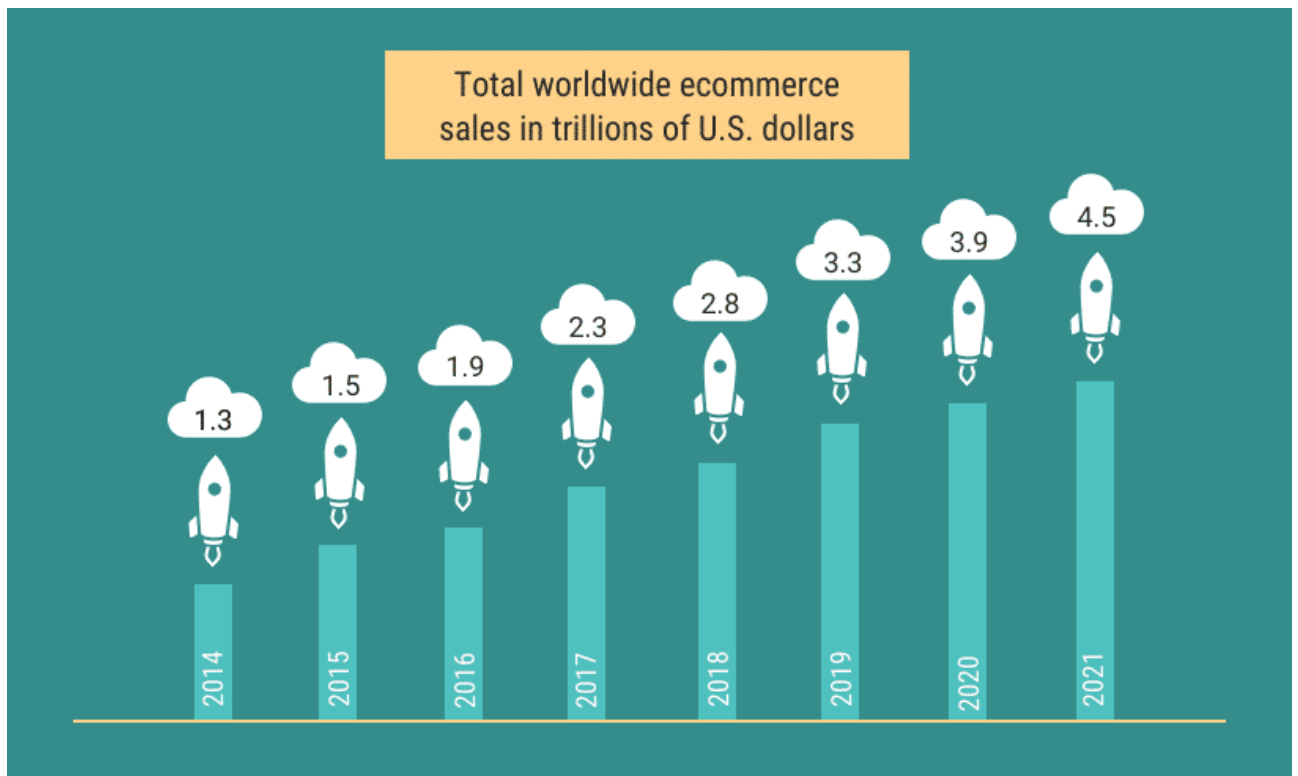


Рис. 1.14. Загальний обсяг світової електронної комерції в трильйонах доларів США з 2014 року по 2021 рік

Сучасне положення ринок розробки інтернет-магазинів рясніє різними CMS та технологічними рішеннями, які дозволяють істотно прискорити та полегшити роботу компаній. Це звільняє підприємців від рутинних завдань і значно покращує швидкість та рівень якості, які безпосередньо стосуються роботи їх співробітників і компанії взагалі.

Саме таким положенням ситуації обумовлена високий попит на розробку сайтів для різноманітних типів інтернет-магазинів. По суті такий вид бізнесу не потребує багато вкладень, достатньо зручний щодо функціоналу, а також може дозволити досягти досить значної переваги не тільки в економії, але й вирішувати різноманітні проблеми зв'язані з самим процесом торгівлі.

«Система управління вмістом (CMS) – це програмне забезпечення або набір відповідних програм, які використовуються для створення та управління цифровим контентом. CMS зазвичай використовуються для управління контентом для підприємств та управління веб-контентом (WCM). [8]»

CMS є можливість умовно поділити на чотири окремі категорії:

- Коробочні. При такому варіанті програмне ядро сайту не доступне для модифікації з боку звичайних людей, цим можуть займатися тільки розробники цієї CMS. Як приклад популярних можна привести Tilda та Wix.
- З відкритим кодом. Завдяки відкритому коду програмне ядро CMS є доступним для модифікації. Такий підхід дозволяє постійно розроблювати нові теми та доповнення. При такій варіанті набагато швидше знаходять уразливості системи. На даний момент популярними CMS з відкритим кодом являються: Drupal, Magento, WordPress, PrestaShop, OpenCart та Joomla.
- Фреймворк – це по суті модифікація функціоналу мови програмування. При такому варіанті розробка сайту буде вимагати значніших вкладень часу та грошей, а ще буде необхідно займатися розробкою адміністративної частини сайту. В такому варіанті замовник може отримати взагалі будь-який функціонал, який захоче. Популярні на даний момент фреймворки: Ruby on Rails, Django, Laravel.
- Самописні. При такому варіанті програмне ядро системи розробляється під особливий, конкретно визначений сайт. Функціонал таких систем не настільки великий, як у інших CMS, але в повній мірі відповідає конкретному проекту, під який розробляється система. Такий спосіб не є дуже вигідним тому, що при якихось проблемах або при наявності бажання модифікувати сайт, буде необхідно звертатися до гарного спеціаліста, котрому буде необхідно розбиратися в коді багато часу.

Якщо брати конкретно OpenCart – це безкоштовна CMS, яку розробляли спеціально для створення різноманітних інтернет-магазинів. Ця CMS з відкритим кодом, являється доступною для модифікування. Підтримкою та доробкою цієї системи займаються багато людей з різних країн світу.

Необхідність розробки веб-сайту впливає з того, що в сучасному світі переважна більшість магазинів будь-якої орієнтації та розміру мають бути

присутніми в Інтернеті, особливо магазини ПК та електронних пристроїв. Інтернет-магазини стали невід'ємною частиною нашого сучасного життя. Важко визначити, в яких сферах це не потрібно, тому що такий спосіб продажу підходить всім.

Інтернет-магазин має багато переваг, вони різні для кожної компанії. До основних переваг сайту треба віднести:

- Абсолютно усі покупки на сайтах є можливість робити не виходячи зі своєї квартири, під час поїздки в автомобілі або на громадському транспорті.
- Наявність зручного підбору і пошуку за параметрами. Така можливість повинна допомогти при пошуку потрібного товару на сайті інтернет-магазину, без великих втрат часу на пошук необхідного. Зручна система підбіру за параметрами може дозволити клієнту вибрати товар без зайвої консультації з менеджером. Це може заощадити багато часу і зробити роботу менеджера більш продуктивною. Без фільтрації клієнти дуже часто стикаються з певними недоліками та труднощами в офлайн торгівлі (рис.1.15).
- Інтернет-магазин пасивно працює над покращенням іміджу фірми або компанії. Якщо у конкурентів немає сайту, тоді зручна сторінка каталогу продуктів дасть магазину певну конкурентну перевагу.
- Відбувається економія часу під час покупки, підбору товару, та багатьох інших дій які супроводжують процес торгівлі. Принаймні деякий час слід дати на дорогу до магазину, ще деякий час на консультацію і певний час на остаточне прийняття рішення. Веб-сайт економить час клієнтів, не доставляє зайвих проблем і є абсолютною заміною походу в магазин і контактуванню з консультантом.
- Компанія покращує якість обслуговування через такі фактори як зручність та економія часу. Завдяки підвищеному рівню сервісу інтернет-магазини користуються великою популярністю зараз, такий спосіб торгівлі зручніше та дешевше в підсумку.

- Складські та торгові приміщення не потрібно орендувати. На початок можна обійтися без менеджера продажів, всю комунікацію з клієнтами можна виконувати самому. Взагалі вдалося буде знайти постачальника, обговорити з ним можливість покупки починаючи з 1 одиниці товару, домовитися з кур'єром який буде працювати з інтернет-магазином, зайнятися аналізом конкурентів, проробити політику щодо цін та почати бізнес. Якщо бізнес буди мати успіх далі є можливість масштабувати результати.
- В інтернет-магазині немає жорсткої прив'язки з приводу регіону. Якщо локація підприємця це - віддалене невелике селище, це зовсім не означає того, що він не може працювати з мегаполісами. А якщо в регіоні є склад і офіс, то і витрати будуть знижені в порівнянні з конкурентами і це зробить необов'язковим зайве розширення бізнесу.
- Аналітика (рис.1.16). Зараз за допомогою найсучасніших аналітичних систем стає доступною можливість проаналізувати те, які джерела реклами приносять магазину або компанії найбільші продажі. Що стосується роздрібною торгівлі, то там нажаль дуже важко відстежити те, звідки конкретно прийшов вірогідний покупець та що саме його цікавить. Цей аналіз майже неможливо провести. Магазин який присутній в інтернеті дає змогу поглянути на свій каталог з боку клієнта і знати, що конкретно він переглядав в каталозі, як саме це відбувалося, до чого він проявив інтерес а до чого ні. Саме це виключає деякі помилки та покращує бізнес. У роздрібній торгівлі про все це можна повністю забути, бо це практично неможливо.



**Фільтри** [Очистити фільтри](#)

**Ціна:**

**Виробник:**


- MiroMark +15
- VMV holding +6
- БМФ** 20
- ВМК (БРВ Україна) +1
- Гербор +3
- Світ Меблів +11

**Фасад:**

- глянцева МДФ 2
- ДСП 3


## Спальні БМФ

Обрано товарів - 20




**Спальня Ліра I БМФ**

21 899 грн



**Спальня Кім НФ Ясен Шимо I БМФ**

14 989 грн






Рис. 1.15. Приклад Підбору та пошуку за параметрами

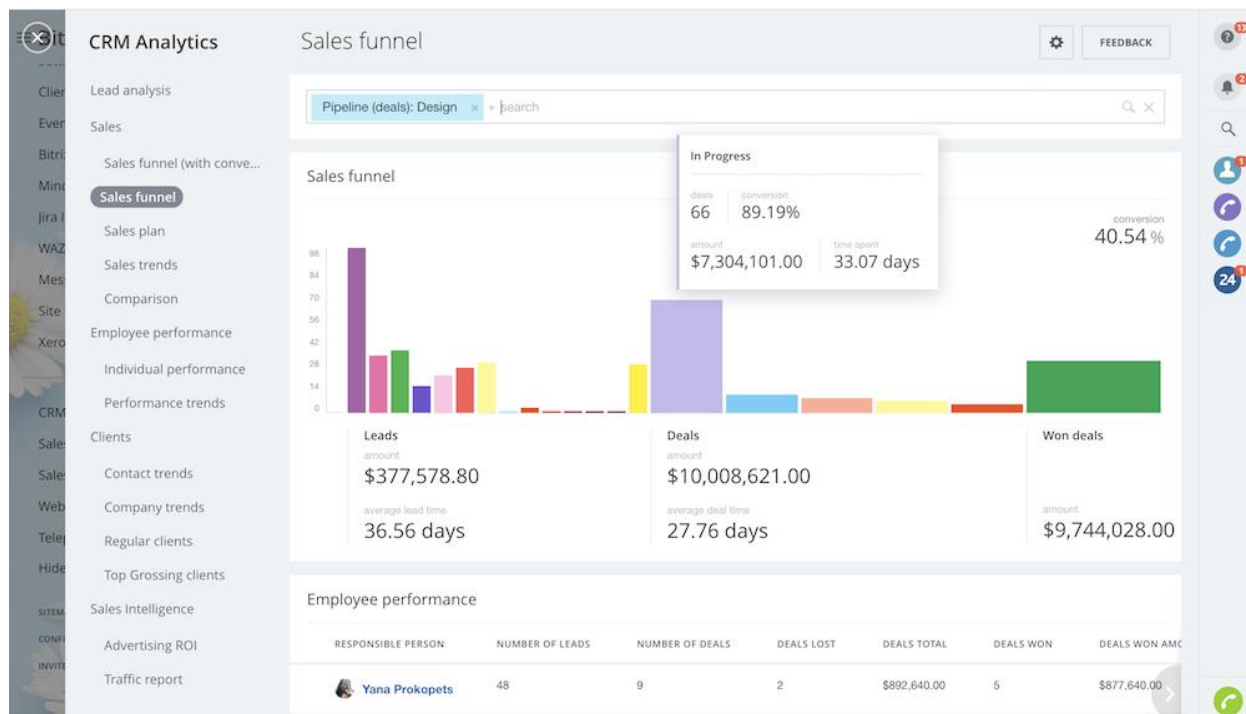


Рис. 1.16. Приклад аналітики CRM

Враховуючи все вищезгадане і сучасні тенденції очевидний той факт, що для магазину персональних комп'ютерів життєво необхідно бути в інтернеті та

займати нішу в галузі інтернет-торгівлі. Власне для цього і розробляється інтернет магазин.

### 1.1.2 Розробка інтернет-магазину

Сьогодні Інтернет стає середовищем яке сильніше розвинене для взаємодії зі споживачами. При цьому показовим є той факт, що Інтернет перетворюється на зручну і відносно недорогу торгову платформу. Все більше компаній намагаються представити свою продукцію в інтернеті. Однак ця ідея не обмежується створенням рекламних сторінок та розміщенням банерів реклами.

З подальшим розвитком інтернету розвивається і сам ресурс пропозиції, на даний момент людина може не тільки лише отримувати інформацію якою цікавиться, а й купувати різні товари та послуги. Зараз в інтернет-магазинах можна придбати різні категорії товарів, як дуже прості, так і достатньо високотехнологічну продукцію. Ця ситуація є пов'язаною з незмінним та постійним зростанням бази потенційних клієнтів в інтернеті.

Зараз велику частку інтернет-аудиторії становлять люди, містом праці яких є офіс, вони проводять багато свого часу за ПК чи смартфоном. Виходячи з цієї ситуації, організація їхнього життя не завжди може дозволити їм витратити надто багато часу на поїздки по офлайн-магазинах, щоб знайти саме те, в чому в них є потреба. Наявність можливості замовити необхідний продукт онлайн для них – це дійсно вирішення проблеми.

Операції, пов'язані з процесом розробки інтернет-магазину, можна, але тільки умовно, розділити на кілька певних етапів:

1. Проектування. Під час цього етапу виконавець робить для основних сторінок сайту прототипи. Прототип – це певна схема. Вказує на те, в якому місці і які блоки будуть розташовані на майбутній сторінці. Прототип потрібен, щоб під час праці ніхто не забув дизайну, а потім в кінці отримати зручний інтерфейс. Зазвичай прототип дизайну затверджується замовником.

2. Підготування тестового середовища. На цьому етапі паралельно з проектною роботою треба починати роботу з підготовки тестового середовища. Основна робота під час розробки відбувається в тестовому середовищі. Процес підготовки по суті включає в себе налаштування самого сервера, створення та налаштування піддомену, встановлення CMS яку потім будуть доробляти, для розробників – налаштування їхніх прав доступу.
3. Дизайн. Після того як прототипи схвалили починається наступний етап – процес дизайну. Під час цієї роботи створюється неповторний зовнішній вигляд кожної зі сторінок сайту, приймаються рішення з приводу кольору, підбір базових медіа та іншого, обробку шрифтів, відступи. Насамперед дизайнер створює дві або більше варіації домашньої сторінки сайту, щоб клієнт мав можливість вибрати ту, яка йому найбільше подобається. Після вибору та змін з боку клієнта, дизайнер починає роботу над всіма іншими сторінками та створює варіації для різних платформ(персональний комп'ютер, ноутбук, планшет, смартфон).
4. Верстка. На цьому етапі, після затвердження дизайну домашньої сторінки, треба починати верстку сайту. Front-End Developer отримує макет дизайну і починає процес верстки сайту. Після завершення вже є можливість побачити, який вигляд матимуть сторінки магазину в різних браузерах. Під час цього етапу відбувається адаптація під різні пристрої та браузери.
5. Програмування. Коли макет готовий, до праці приступає інший програміст (розробник бекенда). Він по суті займається тим що інтегрує верстку з CMS, крім цього він займається розробкою функціоналу сайту. Під час цього етапу розробки розробник працює над логікою сайту. Кажучи про інтернет-магазин, підлягає реалізації весь функціонал яким буде користуватися клієнт, наприклад: відкриття головної сторінки, відкриття категорії, робота фільтрів та сортування, робота пошуку,

- робота кошику, та інше. Відбувається робота над реалізацією функціоналу адміністратора. Після етапу програмування сайт вже майже готовий.
6. Тестування. Після того як програміст виконує завдання, інтернет-магазин переходить безпосередньо до тестувальника, який з метою знайти всі помилки функціональної частини сайту запускає всі базові та неочевидні сценарії роботи сайту. Зазвичай багів багато, і чим більший та складніший проект, тим більше багів знаходить тестувальник. Тестер детально повідомляє про помилки, виявлені на кожному етапі своєї роботи. Програміст ретельно виправляє помилки, перш ніж тестувальник прийме роботу.
  7. Наповнення контентом. Вже після тестування функціональності інтернет-магазину, треба переходити до наступного кроку – наповнення сайту контентом. Існує кілька способів роботи з контентом. Іноді завантаженням займається клієнт, а іноді клієнт наполягає щоб саме розробник наповнив сайт. Якщо говорити про більшу частку інтернет-магазинів, то заповнення відбувається шляхом імпорту прайс-листів. Дуже рідко доводиться додавати товар поштучно.
  8. Реліз. Під час релізу абсолютно вся робота переноситься до основного сайту. Як правило, не часто буває таке, що все розгортається дуже плавно і без нахлестів.

При невеликих розмірах проекту та достатній кваліфікації програміста, всі ці етапи може виконувати одна людина

## **1.2. Призначення розробки та галузь застосування**

Система, що розробляється: Веб-додаток для Інтернет магазину з продажу персональних комп'ютерів на базі CMS OpenCart.

Галузь застосування інтернет магазину: інтернет-торгівля.

Причини з яких виникла необхідність розробки цього програмного забезпечення:

- спрощення оптового продажу;
- прямий та простий вихід на роздрібних клієнтів;
- залучення великої цільової аудиторії;
- значна економія.

В підсумку, уся робота повинна привести до збільшення прибутку від ведення бізнесу. Користуючись потужністю онлайн-ресурсу, є можливість автоматизації великої кількості процесів: спільні покупки, дрібнооптові замовлення, відстеження історії та статусу замовлень. Для дилерів це теж досить вигідно.

Підприємцю для початку цілковито буде підходити досить простий та зручний інтернет-магазин який буде створений на готовому рішенні. Створивши інтернет-магазин, можна миттєво отримати немалу перевагу перед тими, хто скористався нагодою та не зробив цього, адже інтернет-трансформація вже майже захопила усіх гравців цього ринку. Чим раніше станеться впровадження сучасних технологій та адаптація до сучасності, тим краще, в перспективі саме за допомогою цього магазин заробить більше коштів.

### **1.3. Підстава для розробки**

Відповідно до ОПП, згідно з планом навчання та графіків навчального процесу, в кінці навчання студент повинен виконати кваліфікаційну роботу (проект). Тема цієї роботи спочатку узгоджується з керівником проекту, потім з випускаючою кафедрою, та затверджується наказом ректора. В підсумку, підставами для розробки (виконання самої кваліфікаційної роботи) є:

- ОПП за спеціальністю 121 «Інженерія програмного забезпечення»;
- план навчання та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № \_\_\_\_\_ від \_\_\_\_ . \_\_\_\_ .2022 р;

- завдання на дипломний проект на тему «Розробка веб-додатку для Інтернет магазину з продажу персональних комп'ютерів на базі CMS OpenCart».

#### **1.4. Постановка завдання**

При досягненні наступних цілей можна сказати про вигідність для власника магазину розробки сайту. Ці цілі включають:

- зміцнення та поліпшення позицій на ринку послуг та товарів;
- підвищення популярності бренду;
- збільшення прибутку;
- залучення нової аудиторії;
- оптимізація існуючих бізнес-процесів;
- економія часу та грошей.

Тому в сучасному розумінні сайт інтернет-магазину має бути саме тим інструментом, який займається залученням до магазину великої кількості клієнтів, покращенням обслуговування, зниженням витрат на збут і амортизацію, просуванням продукції за рахунок реклами.

Сайт є зручним маркетинговим інструментом для процесу просування товару на ринку, і в кінцевому підсумку призводить до реалізації головного завдання інтернет-магазину — підвищення прибутку який надходить від продажу послуг або продукції та збільшення числа нових покупців та відвідувачів.

Метою кваліфікаційної роботи є розробка сайту для інтернет-магазину з використанням CMS OpenCart для продажу персональних комп'ютерів в мережі інтернет. Розроблений сайт буде дозволяти значно підвищувати виручку та продуктивність існуючого в офлайн магазину, тому що звільнить співробітників компанії від багатьох повсякденних рутинних завдань, залучить нову аудиторію, оптимізує існуючу торгівлю, заощадить гроші та час, значно покращить існуючий рівень послуг магазину.

Треба виконати наступні задачі задля реалізації поставленої в роботі мети:

- розглянути існуючі механізми роботи інтернет-магазинів та способи користування ними;
- провести аналіз сучасного стану галузі з метою виявлення інформації щодо обраної проблематики;
- провести аналіз існуючих інтернет-магазинів з метою виявлення їх властивостей;
- здійснити вибір необхідних інструментів та засобів для розробки інтернет-магазину;
- розробити спеціалізований сайт на базі CMS OpenCart.

### **1.5. Вимоги до програми або програмного виробу**

«Вимоги до програмного забезпечення – це сукупність тверджень щодо атрибутів, властивостей або якостей програмної системи, що підлягає реалізації. Створюються в процесі розробки вимог до програмного забезпечення та у результаті аналізу вимог. Вимоги можуть виражатися у вигляді текстових тверджень і графічних моделей.

За рівнями вимоги можна розділити на:

- Бізнес-вимоги – визначають призначення програмного забезпечення, описуються в документі про бачення (vision) і границях проекту (scope).
- Вимоги користувача – визначають набір користувальницьких завдань, які повинна вирішувати програма, а також способи (сценарії) їх рішення в системі. Користувальницькі вимоги можуть виражатися у вигляді фраз тверджень, у вигляді сценаріїв використання, користувальницьких історій, сценаріїв взаємодії. [9]»

«У класичному технічному підході сукупність вимог використовується на стадії проектування ПЗ. Вимоги також використовуються в процесі перевірки програмного забезпечення, оскільки тести ґрунтуються на вимогах. [10]»

Одними із ключових ознак ефективності впровадження будь-якого програмного продукту є максимальна результативність, також ніяких багів та помилок під час роботи не повинно виникати, це буде неприпустимо. Задля того щоб досягнути максимальної ефективності під час роботи з сайтом треба сформулювати певний ряд вимог, які потім треба виконувати при розробці інтернет-магазину.

Виходячи із бачення кінцевого продукту, можна виділити ряд вимог:

- Сайт повинен реалізувати свою головну функцію – забезпечувати торгові операції бізнесу в інтернеті. Також він повинен мати робочий та якісний функціонал що буде відповідати аналогам, дотримуватися вимог що стосується швидкості, надійності та продуктивності роботи.
- Сайт повинен орієнтуватися на клієнтів та менеджера (адміністратора) інтернет-магазину, і в повній мірі забезпечувати їхні потреби щодо продажу товарів та адміністрування сайту (зрозумілість та інтуїтивність інтерфейсів, легкість в плані використання).
- Сайт має бути вигідним з економічної точки зору.

### **1.5.1. Вимоги до функціональних характеристик**

Задля реалізації мети проекту на web-сайті, який підлягає розробці ,треба реалізувати:

- Зручна навігація по необхідному функціоналу.
- Фільтри товарів за характеристиками. Треба мати можливість сортування продукту за різними характеристиками.
- Методи зворотного зв'язку. Гарним варіантом завоювання довіри з боку клієнтів є зворотній зв'язок з підприємством. Багато людей бажають обговорювати покупки з конкретною людиною і задавати їй різноманітні питання щодо товару. Треба мати можливість зателефонувати.



- Зручна картка товару. клієнту треба знати, що саме за товар він купує, перед оформленням покупки. Наочне представлення товару клієнту це одна з важніших частини торгівлі.
- Мобільна версія магазину. Користувачі повинні отримувати доступ до магазину не лише через комп'ютер, а й через планшет або смартфон. Мобільна версія повинна бути не дуже громіздкою та містити лише необхідний функціонал.
- Зручна, неперевантажена реєстрація. Реєстрація не повинна мати нічого зайвого.
- Зручний модуль оформлення замовлення.

### **1.5.2. Вимоги до інформаційної безпеки**

«Сьогодні більшість людей залежать від веб-сайтів. Вони надають власні імена, дані банківських карток, дати народження, а іноді й паспорти чи ідентифікаційні коди – все це є привабливою метою для кіберзлочинців. Крім того, злам веб-сайту дозволить перенаправити трафік користувачів з метою зараження шкідливим програмним забезпеченням. Від захищеності веб ресурсу також залежить безпека фізичного обладнання, де той розташовано. За умови його некоректного налаштування злочинці можуть здійснити зараження комп'ютера, що може призвести до великих збитків, оскільки знадобиться як мінімум оплатити послуги з видалення шкідливого програмного забезпечення, а як максимум – повністю замінити обладнання.

Втім, віруси – не єдина причина, за якої безпека веб-сайту має значення. Будь-який злам негативно вплине на репутацію як бізнесу в цілому, так й його власника зокрема. До того ж це грозить великою кількістю простоїв та втратою продуктивності, а звідси – до втраченого прибутку (навіть не враховуючи сценарій, за яким зламаний веб-сайт видаляється з усіх пошукових систем). [11]»

Серед основних типів вразливостей веб-сайтів є можливість виокремити наступні:

- Відсутність сесійного таймауту. Завдяки такій ситуації хакер має можливість використати старий ідентифікатор сеансу зв'язку.
- Відсутність обмежень щодо кількості спроб входу до акаунту. В такому випадку хакер має можливість не витрачаючи багато часу перебрати паролі до акаунтів.
- Недостатня автентифікація. В такому випадку хакер має можливість отримати доступ до важливих даних або серверного функціоналу, не маючи необхідних прав доступу.
- Відсутність захисту від зміни HTML-коду сайту.
- SQL-ін'єкція. В такому випадку хакер має можливість внести певні зміни до бази даних або скопіювати її. Як правило це робиться через форми введення.
- Слабкі паролі до адміністративної панелі, а ще типові імена облікових записів (root, admin). В такому випадку хакер може скористатися звичайним брутфорсом (перебором паролів).
- Типове знаходження службових файлів та каталогів.

Щодо переліку вразливостей сайтів, можна привести ще багато прикладів. Від усього на світі захиститися дуже важко, але по можливості необхідно максимально убезпечити сайт.

Для уникнення некоректної роботи web-сайту необхідно реалізувати:

- контроль даних;
- конфіденційність клієнтської інформації;
- цілісність інформації на сайті магазину;
- крос-платформну роботу.

Базовий сайт на CMS OpenCart вже має непоганий рівень захисту, але все одно потребує доробки.

Для виконання вищезгаданих пунктів які необхідно реалізувати, треба зробити наступні кроки:

- видалити стандартний обліковий запис admin, заздалегідь створивши новий;
- замінити адресу адміністративної частини сайту на складну;
- відключити відображення php помилок.

### **1.5.3. Вимоги до складу та параметрів технічних засобів**

Для того щоб сайт на сервері працював треба мати встановлений інтерпретатор мови PHP та всі необхідні утиліти, а для керування базою даних - MySQL. Сайт по суті може працювати на будь-якій сучасній операційній системі. Наприклад: Linux, Windows, Android.

Якщо говорити про мінімальні системні вимоги, то вони для різних браузерів та операційних платформ відрізняються.

Для браузера Google Chrome:

- оперативна пам'ять не менше 512 Mb;
- процесор швидкістю не менше 500 МГц;
- жорсткий диск не менше 350 Mb;
- будь-яка Аудіокарта;
- широкополосний доступ до інтернету;
- екран – SVGA 800x600.

Для браузера Opera:

- оперативна пам'ять не менше 512 Mb;
- процесор швидкістю не менше 500 МГц;
- жорсткий диск не менше 400 Mb;
- будь-яка Аудіокарта;
- інтернет – стабільний;
- екран – SVGA 800x600.

Щодо рекомендованих мінімальних системних вимог сервера:

- оперативна пам'ять не менше 2 Гб;
- жорсткий диск не менше 4 Гб;
- процесор швидкістю не менше 2 ГГц.

#### **1.5.4. Вимоги до інформаційної та програмної сумісності**

Для нормального функціонування web-сайту необхідно, щоб він працював у будь-якому сучасному браузері та на будь-якій сучасній системі. Що стосується Windows, то її версія повинна бути 7 та вище.

## РОЗДІЛ 2

### ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

#### 2.1. Функціональне призначення програми

Під час виконання кваліфікаційної роботи був розроблений веб-додаток, який виконує функції інтернет магазину.

Функціональне призначення розробленого веб-додатку:

- надання власнику сайту зручного інструментарію для модерування інтернет-магазину;
- надання клієнтам сайту необхідної інформації, яка їм необхідна;
- оптимізація бізнес-процесів, а як наслідок зростання ефективності цього бізнесу;
- надання підприємству та клієнтам можливості здійснювати торгові операції в інтернет-просторі;
- надання підприємству всієї необхідної інформації щодо клієнтів та бізнес процесів;
- надання клієнтам сайту комфортного та зручного інтерфейсу;
- спрощення менеджменту інтернет-магазину.

#### 2.2. Опис застосованих математичних методів

Під час виконання даного проекту ніякі математичні методи не використовувались.

#### 2.3. Опис використаної архітектури та шаблонів проектування

OpenCart побудований за принципом MVC. Ця архітектура на даний момент досить широко використовується в веб-програмуванні. Архітектурний шаблон MVC (рис.2.1) можна охарактеризувати за допомогою трьох рівнів які

пов'язані між собою, та складають певну систему, яка допомагає вирішити проблему зі систематизацією коду.

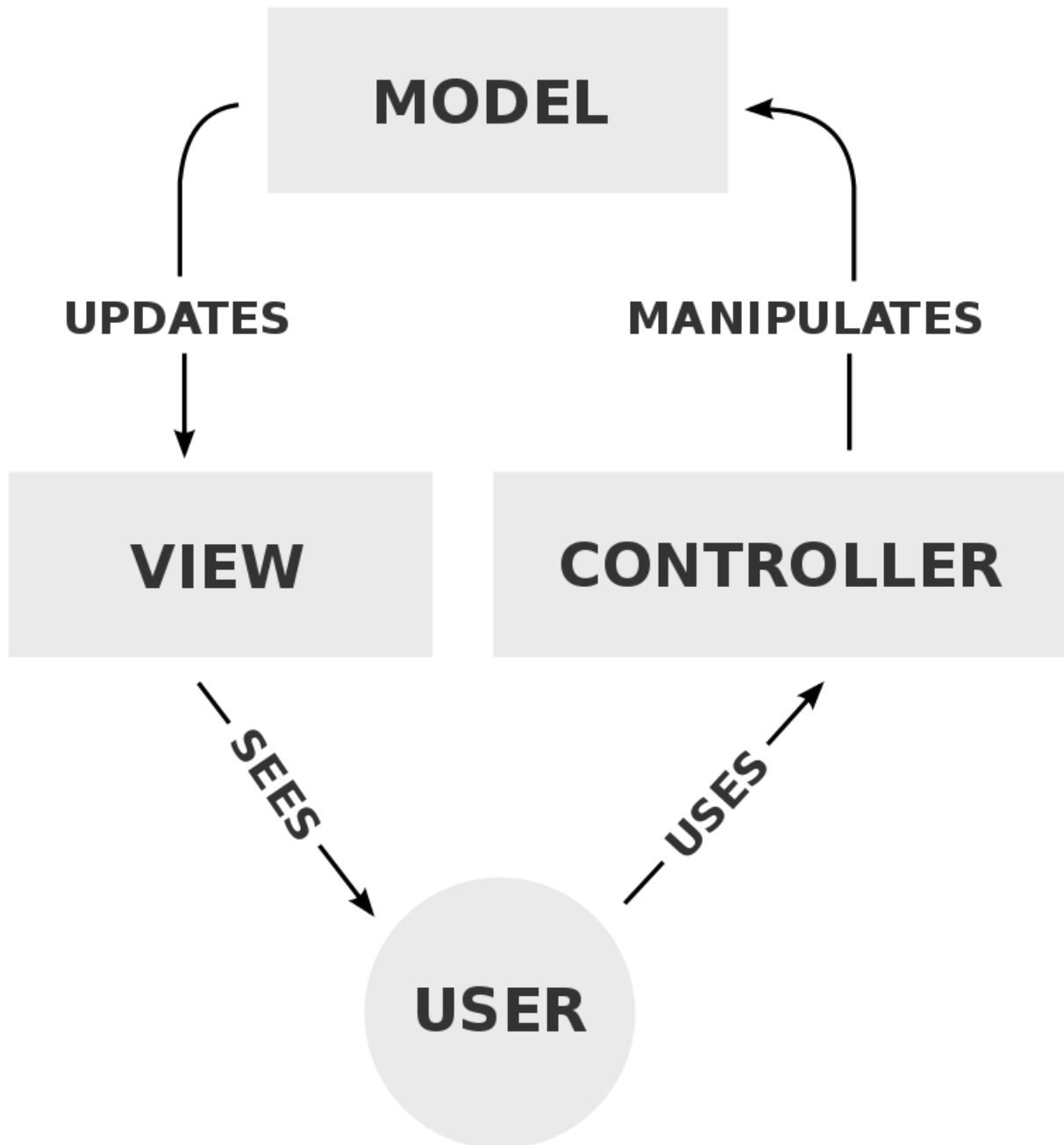


Рис. 2.1. Приклад архітектури MVC

«MVC - це скорочення від Model, View та Controller. MVC - це популярний спосіб організації коду.

Основна ідея MVC полягає в тому, що кожен розділ вашого коду має своє призначення, і ці цілі різні. Частина коду містить дані програми, частина коду робить додаток приємним, а частина коду контролює, як функціонує програма.

MVC - це спосіб впорядкувати основні функції коду у власні, акуратно організовані поля. Завдяки цьому думати про свою програму, переглядати її та ділитися нею з іншими набагато легше.

Model: код моделі зазвичай відображає реальні речі. Цей код може містити необроблені дані, або він визначатиме основні компоненти вашого додатка.

View: код представлень складається з усіх функцій, які безпосередньо взаємодіють з користувачем. Це код, завдяки якому додаток виглядає приємно, а в іншому випадку визначає, як користувач бачить і взаємодіє з ним.

Controller: код контролера діє як зв'язок між моделлю та представленням, отримуючи вхід користувача та вирішуючи, що з ним робити. Це головний мозок програми та пов'язує модель і представлення.

MVC - це основа для роздумів про програмування та організацію файлів програми. Щоб підкреслити ідею, що код повинен бути організований, розробники створюють папки для кожної частини MVC.

MVC дає початкове місце для конвертації ідей у код, а також полегшує повернення до старого коду, оскільки можна буде визначити, який код що робить. Крім того, організаційний стандарт, який просуває MVC, полегшує іншим розробникам розуміння коду. [12]»

MVC не має суворої реалізації, з цієї причини цю архітектуру можна застосовувати по-різному. Немає одного єдиного визначення того, де саме має бути бізнес-логіка. Вона може бути застосована як в контролері, так і в моделі. В останньому випадку модель повинна містити у собі всі бізнес-об'єкти з усіма функціями та даними.

Деякі з фреймворків вказують на те, де саме повинна бути розташована бізнес-логіка, тоді як інші виключають це.

Також невизначено те, де саме повинна бути перевірка даних які ввів користувач. Проста валідація може існувати навіть у вигляді, але все ж таки частіше вона зустрічається в контролері чи моделі.

## **2.4. Опис використаних технологій та мов програмування**

Використання всіх технологій, фреймворків та мови програмування було продиктовано тим фактом, що їх використовує для своєї роботи CMS OpenCart.

Під час виконання даної кваліфікаційної роботи були використані наступні мови програмування, технології та бібліотеки мов програмування:

- «HTML – це стандартизована мова розмітки документів в інтернеті. В першу чергу, необхідно зрозуміти, що HTML - основа кожної веб-сторінки, незалежно від складності сайту або кількості задіяних технологій. Це важливий навик для будь-якого веб-професіонала і відправна точка для всіх, хто має відношення до створення або редагування контенту в Інтернеті. Всупереч поширеній помилці, HTML - це не мова програмування. Вона використовується для того, щоб всі елементи на сторінці (тексти, малюнки, таблиці ...) були розташовані правильно; за його читання відповідають спеціальні програми, які всім знайомі – браузері.
- Bootstrap є найпопулярнішим фреймворком і має саме велике співтовариство, тому розробка за допомогою цього фреймворка буде завжди актуальною. Bootstrap дозволяє верстати сайти в кілька разів швидше, ніж це можна виконати на чистому CSS і JavaScript. Крім цього, його популярність ще обумовлена доступністю. Вона полягає в тому, що на ньому навіть розробник початківець може верстати досить якісні макети, які важко було б виконати без глибоких знань веб-технологій і достатньої практики. Фреймворк Bootstrap це набір CSS стилів і JavaScript файлів. Для того щоб його використовувати ці файли необхідно підключити до сторінки. Після цього стануть доступні



- інструменти даного фреймворка: колоночна система (сітка Bootstrap), класи, скидання стандартних браузерних стилів, плагіни та компоненти. Але Bootstrap – це не просто набір готових інструментів, а добре спроектований фронтенд фреймворк, який досить просто можна налаштувати під себе за допомогою редагування Sass змінних і використання міксинів. [13]»
- «Каскадні таблиці стилів (CSS) - це мова таблиці стилів, що використовується для опису презентації документа, написаного мовою розмітки, такою як HTML. CSS - це наріжна технологія Всесвітньої павутини, поряд із HTML та JavaScript. CSS призначений для розділення презентації та вмісту, включаючи макет, кольори та шрифти. Це розділення може покращити доступність вмісту, забезпечити більшу гнучкість та контроль у специфікації характеристик презентації, дозволити спільному форматуванню декілька веб-сторінок, вказавши відповідний CSS в окремому файлі .css, що зменшує складність та повторюваність структурного вмісту, а також дозволяє файл .css, який буде кешовано, щоб покращити швидкість завантаження сторінки між сторінками, що ділять файл, та його форматування. [14]»
  - «PHP – це мова програмування, мова сценаріїв загального призначення з відкритим вихідним кодом, спеціально розроблена для написання web-додатків, що виконуються на Web-сервері. [15]»
  - «JavaScript – це мова програмування загального призначення. Він не надає низькорівневих засобів роботи з пам'яттю, процесором, так як спочатку був орієнтований на браузери, в яких це не потрібно. Що ж стосується інших можливостей – вони залежать від оточення, в якому запущений JavaScript. У браузері JavaScript вміє робити все, що відноситься до маніпуляції зі сторінкою, взаємодії з відвідувачем і, в якійсь мірі, з сервером. JavaScript спочатку створювався для того, щоб зробити web сторінки живими. Програми на цій мові називаються

скриптами. У браузері вони підключаються безпосередньо до HTML і, як тільки завантажується сторінка – тут же виконуються. [16]»

- «MySQL – це безкоштовна система управління реляційними базами даних. Сайт MySQL розробляється та підтримується корпорацією Oracle, яка спільно придбала права на торгову марку із Sun Microsystems, яка раніше придбала шведську компанію MySQL AB. Продукт поширюється як за загальною публічною ліцензією GNU, так і за власною комерційною ліцензією. Крім того, розробники створюють функції для замовлення ліцензованих користувачів. Саме завдяки цьому наказу механізм реплікації з'явився майже в самих ранніх версіях. Гнучкість бази даних MySQL підтримується великою кількістю типів таблиць: користувачі можуть вибрати як таблиці MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, які підтримують транзакції на рівні окремих записів даних. Крім того, база даних MySQL містить спеціальний тип таблиці ПРИКЛАД, який демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі та ліцензуванню СУБД MySQL постійно з'являються нові типи таблиць. [17]»

- «jQuery - це швидка та стисла бібліотека JavaScript, створена Джоном Резігом у 2006 році з приємним девізом: Пишіть менше, робіть більше. jQuery спрощує обробку документів HTML, обробку подій, анімацію та взаємодію Ajax для швидкої веб-розробки. jQuery - це інструментарій JavaScript призначений для спрощення різних завдань шляхом написання менше коду. [18]»

- CMS OpenCart версії 2.3.0.2.

Хочу зазначити, що стандартні версії Bootstrap та jQuery в базовому варіанті CMS були замінені на новіші під час виконання даної кваліфікаційної роботи.

## 2.5. Опис структури програми та алгоритмів її функціонування

Розроблений веб-додаток побудований на базі CMS OpenCart, яка в свою чергу була розроблена базуючись на основі архітектури MVC. Веб-додаток використовує стандартні для цієї CMS алгоритми.

Структурні складові веб-додатку:

- Controller займається обробкою всіх даних. У цій складовій міститься максимум всієї необхідної логіки сайту. Необхідний для обробки дій користувача.
- Model – взаємодіє з базою даних, ця складова архітектури отримує необхідну інформацію, а потім передає її в тому виді, який необхідний. Код моделі складається в основному із запитів до бази даних та перетворення даних до іншого формату. У цих файлах зберігаються класи та методи, які взаємодіють з БД.
- View – відображає всю інформацію. В ньому міститься мінімум необхідної логіки. Складається з шаблонів з розширенням tpl.

Також можна умовно виділити ще одну складову – Language. Ця складова зручно розділяє всю мовну інформацію на сайту.

Крім цього слід зазначити що система працює за методологією Об'єктно-орієнтованого програмування.

«Об'єктно-орієнтоване програмування – це метод програмування, заснований на поданні програми у вигляді сукупності взаємодіючих об'єктів, кожен з яких є екземпляром певного класу, а класи є членами певної ієрархії наслідування. [19]»

Директорія сайту складається з двох структурних частин – інтерфейсів панелі управління та вітрини, які розташовані в окремих папках.

Усі дані щодо панелі управління знаходяться в папці під назвою admin (для більшої безпеки сайту назву було змінено на 95DxajEC43), клієнтської частини – catalog. Кожна з цих папок містить структурні складові MVC: папку controller,

папку view, папку model та папку, що відповідає за мовну частину сайту – language.

Папка system розміщує у собі класи та функції, що в свою чергу використовуються як в адміністративній частині сайту, так і в каталозі.

В цій папці розміщуються директорії основних визначень контролерів, бібліотеки класів, допомоги, бази даних, моделей та всіх інших складових цієї системи.

Директорія image розміщує в собі зображення, які були завантажені Менеджером зображень.

Файли які стосуються клієнтської частини представлення, таблиць стилів та зображення, розміщуються в директорії catalog/view/theme/default.

Директорія default містить у собі три наступні папки:

- stylesheet – папка з таблицями стилів;
- template – папка з шаблонами;
- image – папка із зображеннями шаблону.

Загальні файли представлення шаблону:

- header.tpl – шаблон верхньої частини сторінки, що відображається, в цьому файлі відбувається підключення скриптів та стилів;
- home.tpl – шаблон головної сторінки;
- content\_top.tpl – верхня частина області що відображається;
- content\_bottom.tpl – нижня частина області що відображається;
- column\_right.tpl – права колонка сторінки, що відображається;
- column\_left.tpl – ліва колонка сторінки, що відображається;
- cart.tpl – шаблон модуля кошика у хедеру;
- footer.tpl – шаблон нижньої частини сторінки, що відображається.

Для розуміння системи потрібно пояснити роль бібліотек. Системні бібліотеки розташовані за адресом /system/library.

Вони необхідні для того, щоб уникнути дублювання коду в системі. Робота з ними відбувається у класах, які наслідуються від класів Model та Controller.

Наприклад `system/library/cart/customer.php` розміщує в собі методи реєстрації та авторизації користувачів системи, отримання інформації яка стосується поточного користувача, перевірки на вхід.

Що стосується Контролерів, то вони завантажуються на базі того значення, яке передає параметр `route`. Контролери клієнтської частини сайту розташовані за адресом `/catalog/controller/`.

Якщо `route` сторінки має значення `product/product`, тоді цю сторінку опрацював `/catalog/controller/product/product.php`. Ця система є дуже зрозумілою та прозорою.

У контролері за замовчуванням спрацьовує метод `index()`.

Усі необхідні змінні, що знаходяться в контролерах, які передаються до `view`, поміщаються до асоціативного масиву `$data`. Змінні з масиву будуть доступні у файлі `view` за значенням ключа. В такій ситуації увесь функціонал змінних зберігається.

Для підключення `view` за допомогою контролера використовуються бібліотеки `response` та `load`.

Метод `view` який розташований в бібліотеці `load` займається підключенням файлу представлення, до цього файлу передається масив даних `$data`.

Метод `setOutput` який розташований в бібліотеці `response` виводить вміст.

Щодо файлів моделей то вони згруповані на основі функцій, які вони виконують. Ці файли розташовані за адресом `admin/model` або `catalog/model`.

Підключити певну модель є можливим за допомогою коду `$this->load->model('folder/file');`

Після підключення до контролера, після цього стануть доступні методи моделі.

Виходячи з того що модель безпосередньо відповідає за отримання даних, SQL запити прописуються в ній. Вони виконуються бібліотекою `db` та її методом `query`. SQL запит має константу `DB_PREFIX`, в якій зберігається префікс бази даних, що задається при інсталюванні.

Що стосується алгоритму роботи програми, то він починається з Файлу `index.php`. Він являється точкою входу цієї системи, та знаходиться в головній директорії. Через цей файл проходять всі запити до сервера

Алгоритм праці цього файлу:

1. Спочатку оголошується константа, яка містить версію системи.
2. В тому випадку якщо сайт вже встановлений підключається `config.php`, в ньому оголошуються константи які містять в собі шляхи до БД, а також шляхи до робочих та службових директорій.
3. Після цього підключається системний файл `startup.php`, який розташований в системній директорії.
4. Останній рядок файлу `index.php` виконується тільки після підключення `startup.php`. До функції надається аргумент, який містить напрям до робочого каталогу.
5. Алгоритм праці `system/startup.php`:
6. Спочатку встановлюється протоколювання помилок.
7. Версія `php` проходить перевірку.
8. Часова зона встановлюється за замовчуванням.
9. Відбувається встановлення певних значень суперглобального масиву `$_SERVER`. Наприклад ці значення містять доменне ім'я сайту.
10. Функції `modification()` оголошується, вона необхідна для того, щоб отримувати шлях до модифікованих файлів системи.
11. Підключення автозавантажувача.
12. Після цього йде код автозавантаження системних класів. Функція `spl_autoload_register()`, кої приходить запит якогось невідомого класу, звертається до функції `library()`, яка в свою чергу згаходить, а потім завантажує клас який запитували.
13. Підключається певний ряд важливих для системи класів з директорії `system\engine`. Екземпляри класів створюються потім в файлі `framework.php`, а після цього розташовуються в реєстрі, завдяки чому

доступ до них є через завантажувач loader.php. В такій ситуації класи можна використовувати в будь-якому місці цієї системи.

14.3 директорії system/helper завантажується декілька файлів, які містять в собі функції допомоги. Наприклад utf8.php – встановлює кодування.

15. В самому кінці коду файлу оголошується функція start(), вона підключає файл фреймворку.

16. Після всього цього скрипт повертається до останнього рядка файлу index.php, що запускає підключення framework.php.

Що стосується алгоритму виведення сторінки, то після підключення бібліотек, які необхідні, у файлі framework.php створюється об'єкт класу Front.

Після цього у цьому ж файлі з файлу конфігурації, за допомогою циклу, до \$pre\_action класу Front заноситься ряд методів службових класів, що виконуються першочергово.

Далі викликається наступний метод – dispatch(), перший аргумент якого – об'єкт Action виконує переданий йому метод класу startup/router. Другим аргументом надається об'єкт Action, що реалізує стандартний метод index контролеру ControllerErrorNotFound, котрий буде використовуватись тоді, коли виникне якась помилка при підключенні методів або використанні контролеру.

Коли доходить до реалізації методу index() контролеру ControllerStartupRouter, з поточного URL у код виходять GET-параметри, це значення параметра route – методу контролера, що необхідно реалізувати.

Потім у класі ControllerStartupRouter реалізується метод trigger(), який викликає реалізацію роботи всіх інших обробників, які є прив'язаними до цієї події.

Після цього вже реалізується та дія, яка міститься в змінній route, потім ще раз реалізуються обробники подій, які реалізують методи, які прив'язані до події.

Щоразу переходячи по сторінках контролер ControllerStartupRouter викликає інший контролер.

Якщо говорити про процес реєстрації, то він має наступний алгоритм:

1. Користувач переходить на сторінку.

2. Запит логіну, контактної інформації та паролю.
3. Перевірка введених даних на відповідність вимогам.
4. Якщо не все коректно, то додається новий обліковий запис.

Якщо говорити про процес оформлення замовлення, то він має наступний алгоритм:

1. Користувач переходить на сторінку товару.
2. Додає товар до кошика.
3. Переходить на сторінку оформлення замовлення.
4. Вибирає варіанти оформлення замовлення.
5. Запит контактної інформації або її підтвердження.
6. Якщо все коректно, то переходить на крок підтвердження замовлення.
7. Підтвердження замовлення, замовлення додається до панелі замовлень в адміністративній частині сайту.

Що стосується бази даних яку використовує веб-додаток, то це стандартна БД CMS OpenCart (рис.2.2).



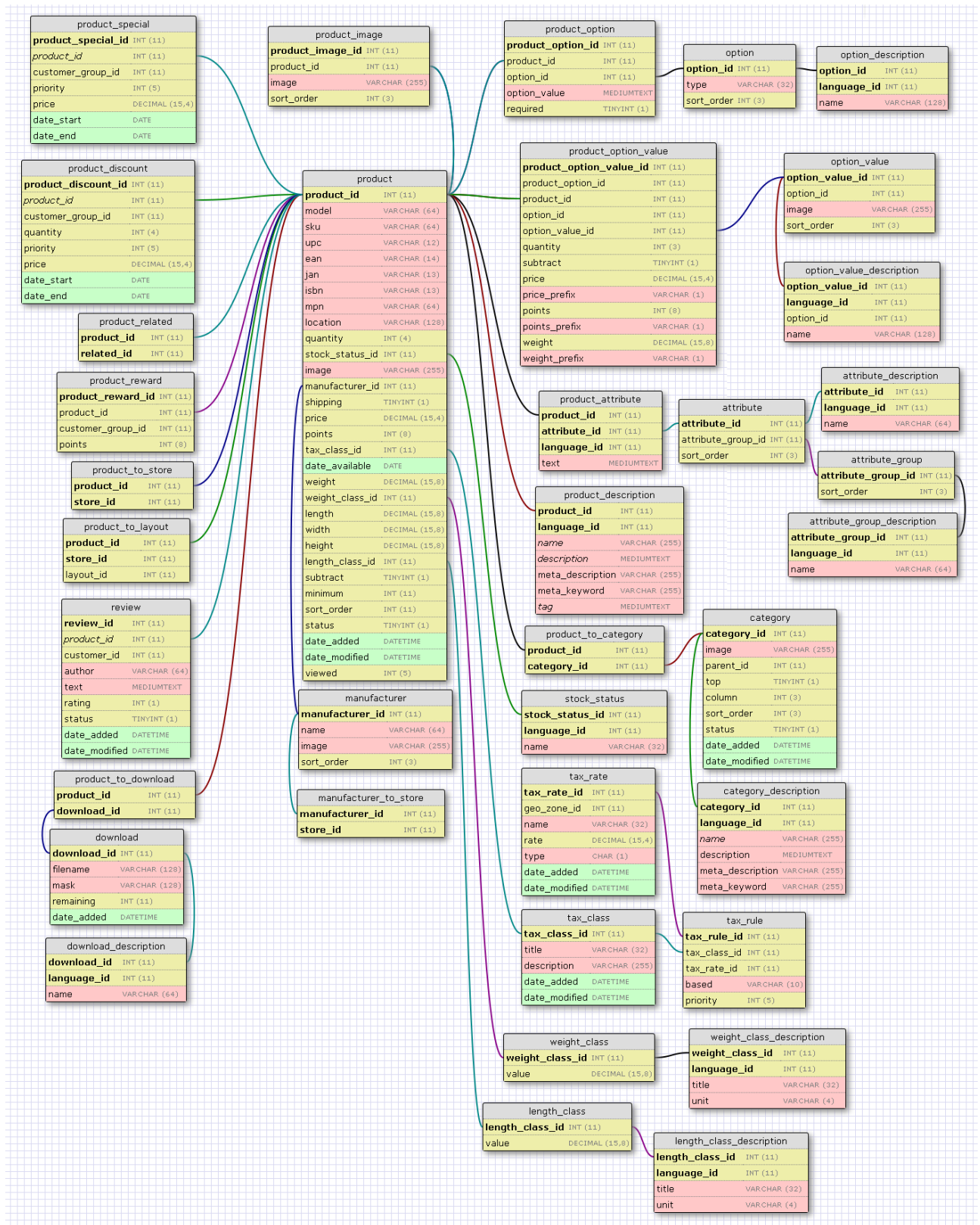


Рис. 2.2. Приклад структури бази даних веб-додатку на базі CMS OpenCart

## 2.6. Обґрунтування та організація вхідних та вихідних даних програми

Що стосується вхідних та вихідних даних інтернет-магазину, то у межах даної інформаційної системи існує багато різних інформаційних потоків які виконують різні функції.

Освітити вхідні та вихідні дані сайту можливо з двох різних сторін, першою являється адміністратор інтернет-магазину, а другою його клієнт.

До вхідних даних клієнтської частини можна віднести:

- дані щодо клієнта, авторизаційні та реєстраційні;
- дані щодо товарів, та їх замовлення;
- дані які стосується зворотного зв'язку.

До вхідних даних адміністративної частини можна віднести:

- дані о клієнтах;
- дані що стосуються товарів;
- дані налаштування сайту.

До вихідних даних клієнтської частини можна віднести:

- дані про інтернет-магазин;
- дані про товари;
- дані про замовлення, реєстрацію, авторизацію;
- контактні дані, та дані щодо зворотного зв'язку.

До вихідних даних адміністративної частини можна віднести:

- дані щодо сайту та його дизайну та структури;
- дані щодо товарів;
- дані щодо клієнтів;
- дані про статистику торгівельної діяльності;
- дані щодо налаштування сайту.

Усі вхідні дані на сайті інтернет-магазину реалізуються за допомогою різних типів форм та кнопок html, де клієнту або адміністратору треба щось писати або вибирати.

Вихідними даними сайту інтернет-магазину є вміст сторінок, як в клієнтській частині сайту, так і в адміністративній.

Усі сторінки на сайті формуються за допомогою шаблонів `tpl`, в яких розміщується як `html` код, так і деяка логіка написана на `php`.

Кодування всіх сторінок сайту – UTF-8.

«Кодування UTF-8 — це різновид кодування Unicode, призначене для Web-дизайну. Кодування Unicode (а виходить, і UTF-8) може закодувати всі символи всіх мов, наявних на Землі. Саме воне в цей час найчастіше застосовується для створення Web-сторінок. [20]»

## **2.7. Опис розробленого програмного продукту**

Розроблений в ході виконання даної кваліфікаційної роботи веб-додаток являється спеціалізованим інтернет-магазином персональних комп'ютерів, що призначений забезпечити стабільну та вигідну комерційну діяльність в цифровому просторі України.

Основним та головним вигодонабувачем, для якого призначається даний веб-додаток, являється власник невеликого бізнесу з продажу персональних комп'ютерів, який потребує розширення свого бізнесу в цифровий простір. Замовник в такому випадку вирішує ряд проблем, які зв'язані з його бізнесом, та становиться більш конкурентним на своєму ринку.

Також вигодонабувачами являються клієнти даного бізнесу, оскільки розроблений веб-додаток вирішує для них ряд певних труднощів та проблем, які зв'язані з процесом покупки товарів.

### **2.7.1. Використані технічні засоби**

Для роботи веб-додатку в інтернеті потрібно, щоб він розташовувався на сервері, на якому встановлене все необхідне програмне забезпечення, та який

відповідає мінімальним системним вимогам з точки зору технічних характеристик. Сервер це по суті звичайний комп'ютер.

Що стосується роботи сайту з боку користувача, то сайт може працювати на різних цифрових обчислювальних пристроях:

- стаціонарний комп'ютер;
- смартфон;
- планшет;
- ноутбук;
- SMART-телевізор.

Головне щоб пристрої мали усе необхідне програмне забезпечення, а також відповідали необхідним технічним характеристикам.

Також є необхідним підключення до інтернету.

### **2.7.2. Використані програмні засоби**

Що стосується програмних засобів, на яких може працювати сайт, то це можуть бути різні операційні системи:

- Linux;
- Microsoft Windows;
- MacOS;
- Android;
- iOS;
- Windows Phone;
- Tizen;
- Web OS.

Це умовний перелік основних операційних систем, які використовують для роботи в інтернеті з різних пристроїв.

Що стосується браузерів на яких може працювати сайт:

- Google Chrome;

- Mozilla Firefox;
- Vivaldi;
- Microsoft Edge;
- Safari;
- Opera.

### **2.7.3. Виклик та завантаження програми**

Процес завантаження сайту з сервера до браузеру:

1. Користувач набирає в адресний рядок, який знаходиться в браузері, адресу сайту.
2. Далі відбувається обробка запитів до DNS-сервера. Іншими словами запит до хосту.
3. Відбувається HTTP-переадресація.
4. Надходить відповідь від сервера, HTTP-сервер передає вміст сторінки браузеру.
5. Браузер обробляє вміст сторінки.
6. Браузер відмальовує вміст сторінки.
7. Повне завантаження головної сторінки сайту.

### **2.7.4. Опис інтерфейсу користувача**

Після запуску сайту користувачеві доступний наступний функціонал навігації верхньої частини сайту (хедеру) (рис.2.3):

- перехід на сторінку зворотного зв'язку;
- перехід на сторінку особистого кабінету;
- перехід на сторінку закладок;
- перехід на сторінку кошика;
- перехід на сторінку оформлення замовлення;
- пошук по сайту;

- перегляд вмісту кошика;
- перехід на сторінку категорії товарів – комп’ютери;
- перехід на сторінку категорії товарів – комплектуючі, або її підкатегорій;
- перехід на сторінку – про нас;
- перехід на сторінку доставки;
- перехід на сторінку гарантії.

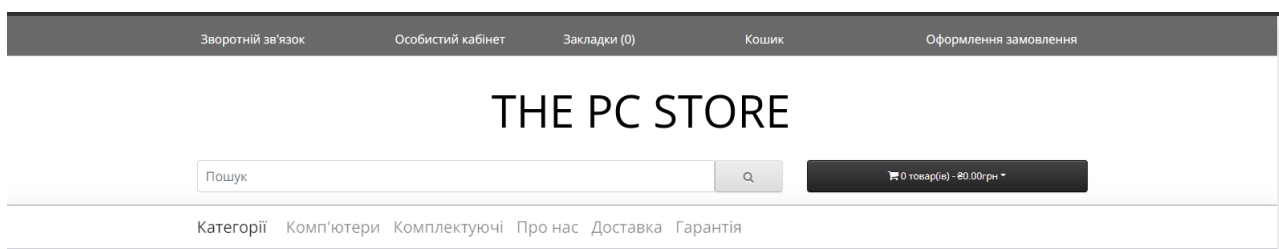


Рис. 2.3. Хедер веб-додатку

При недостатній ширині екрану хедер адаптується та згортається в бургер меню (рис.2.4) (рис.2.5).

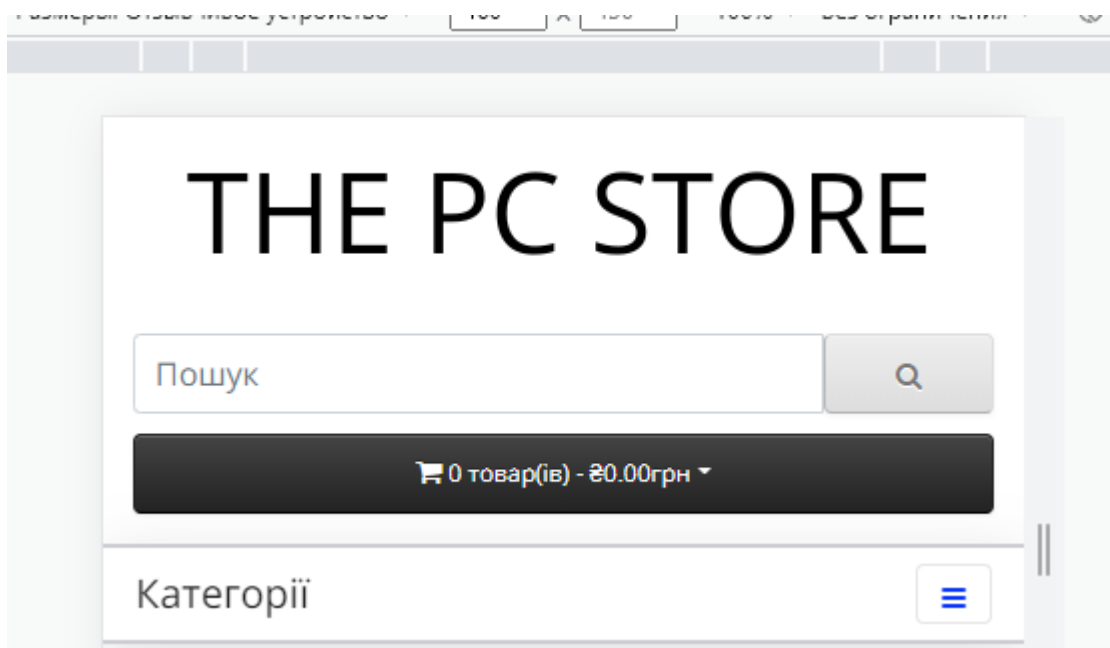


Рис. 2.4. Адаптований під невелику ширину хедер веб-додатку

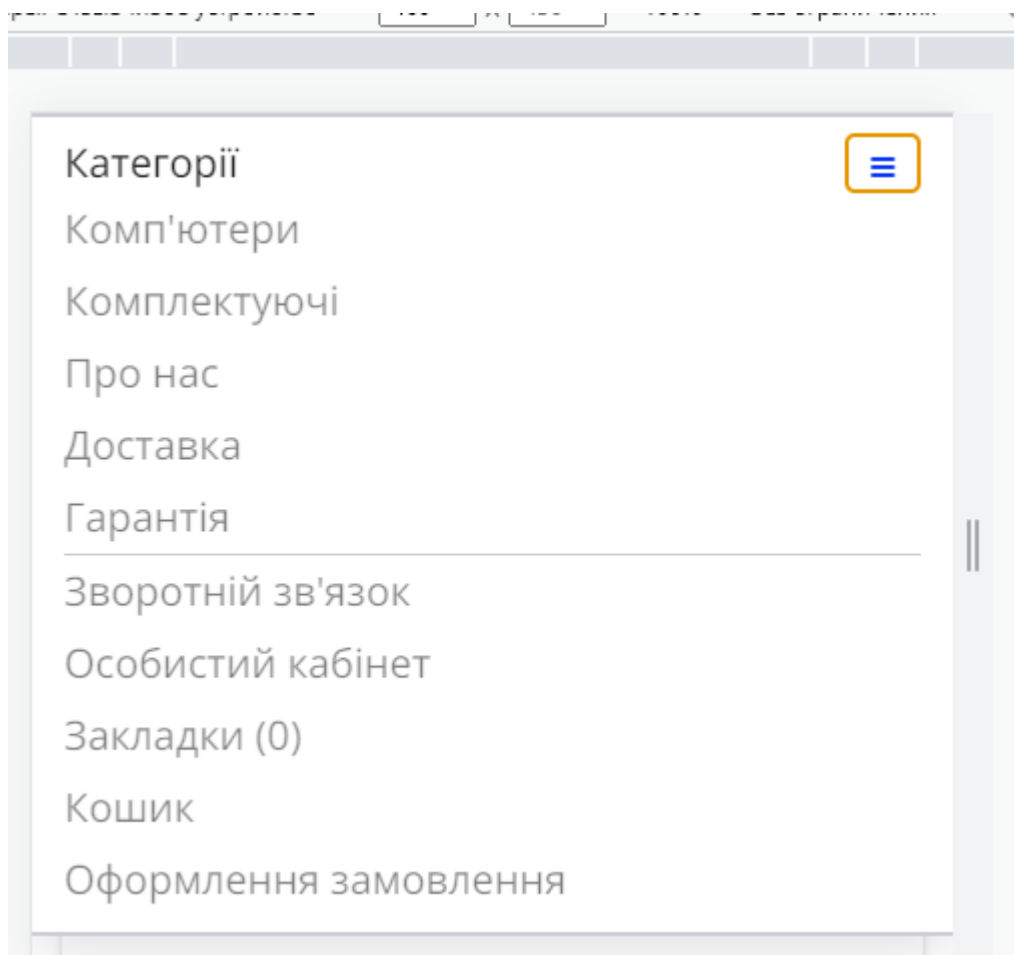


Рис. 2.5. Зміст бургер меню

На сторінці категорії комп'ютерів є фільтрація за параметрами (рис.2.6).

**Фільтр товарів**

**Ціна:**

12888 € 14887

**Застосувати**

**Процесор**

Core™ i5 10400 **1**

Core™ i5 11400 **1**

**Відеокарта**

GeForce GT 1030 **1**

Intel HD Graphics **1**

RX 550 4Gb **1**

**ОЗУ**

16 Гб **2**

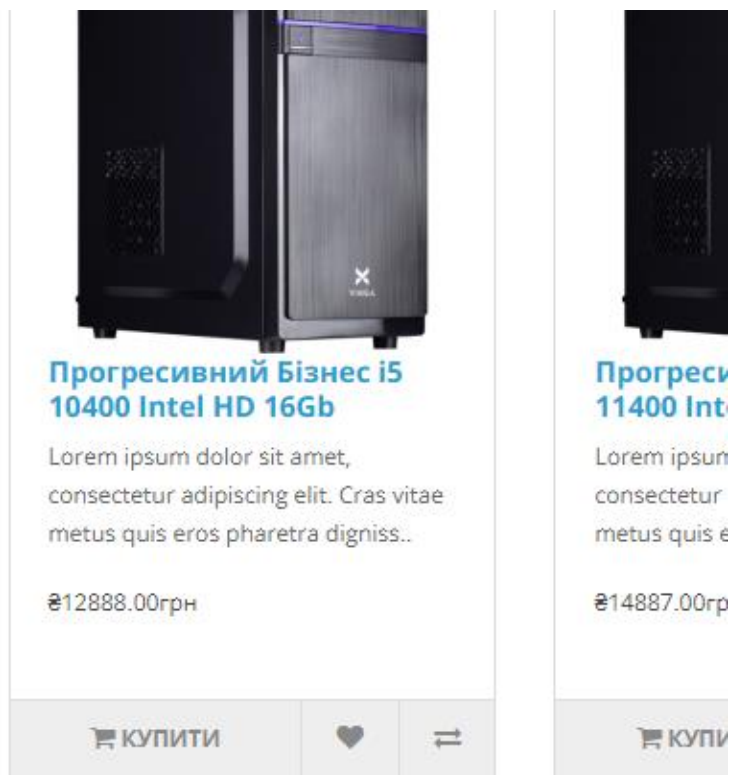


Рис. 2.6. Фільтрація за параметрами

Головна сторінка сайту має наступний вигляд (рис.2.7).



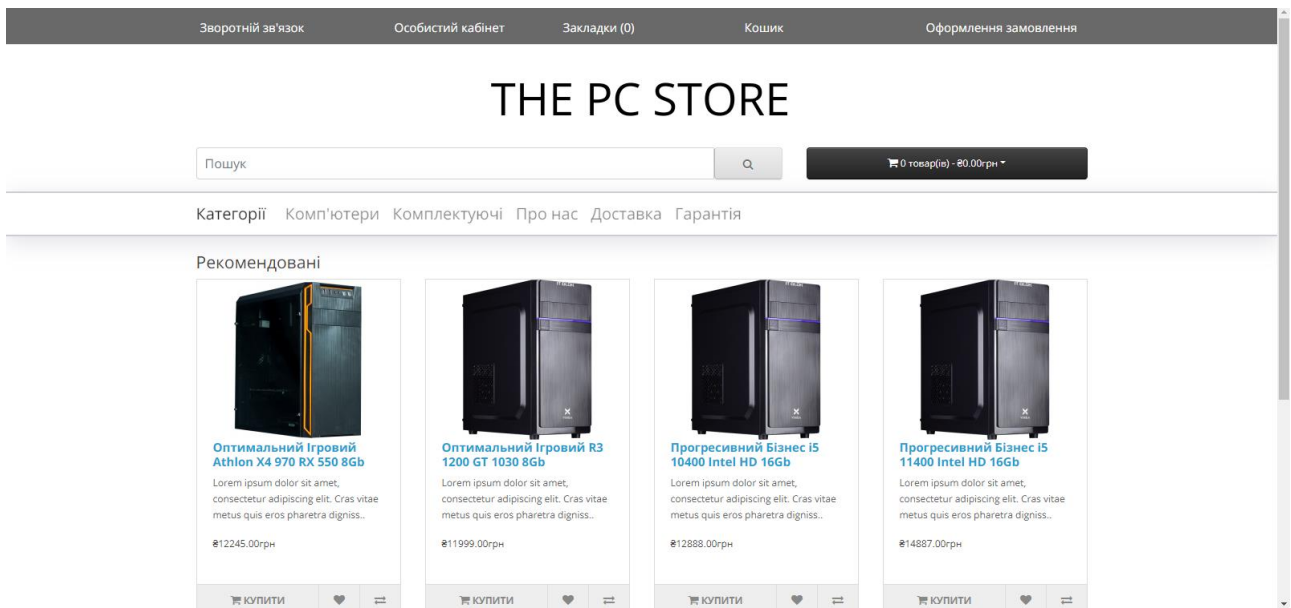


Рис. 2.7. Зовнішній вигляд головної сторінки сайту

Сторінка зворотного зв'язку має наступний вигляд (рис.2.8).

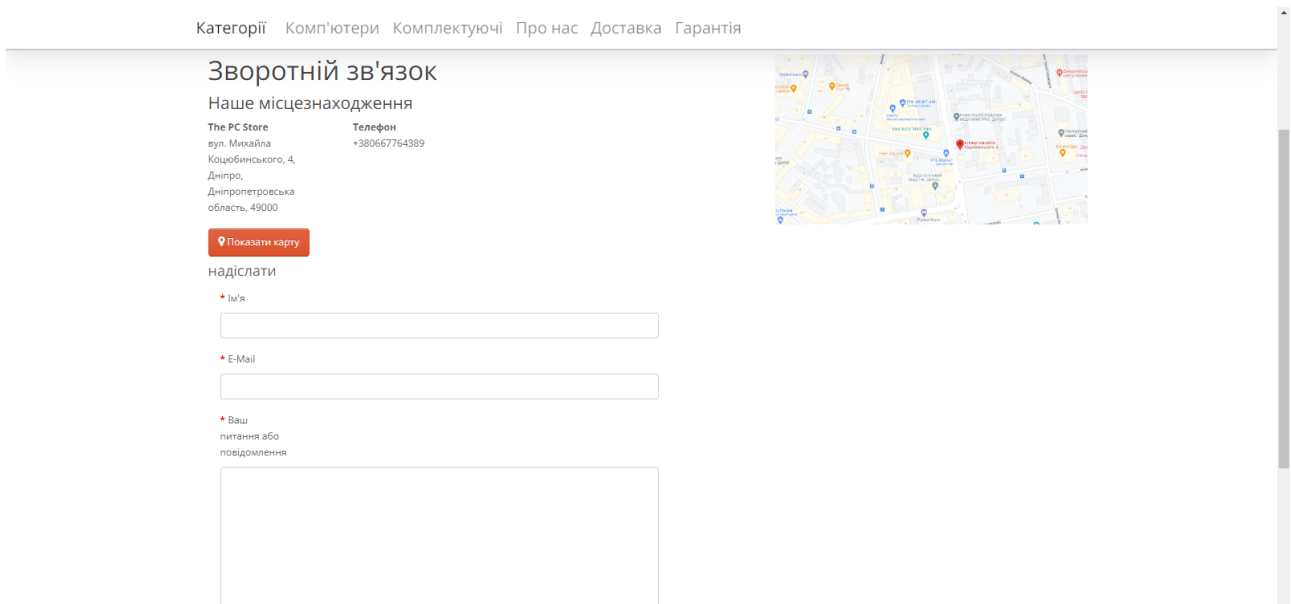


Рис. 2.8. Зовнішній вигляд сторінки зворотного зв'язку

Сторінка категорії має наступний вигляд (рис.2.9).

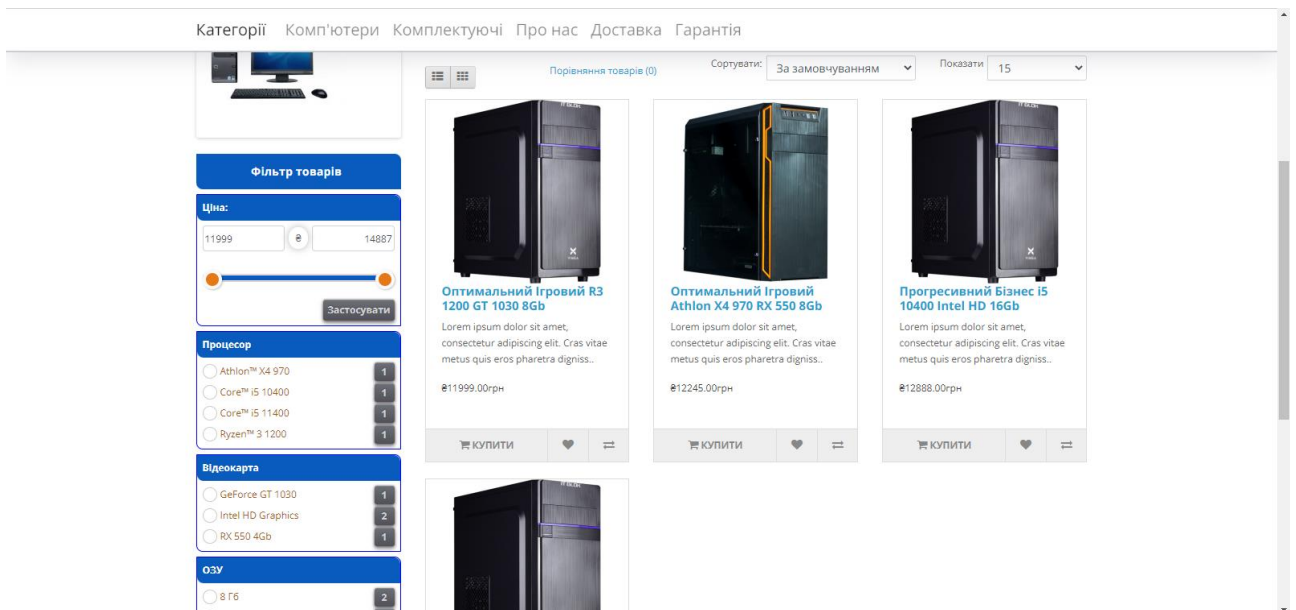


Рис. 2.9. Зовнішній вигляд сторінки кетегорії

При натисканні на особистий кабінет користувач має можливість реєстрації (рис.2.10).

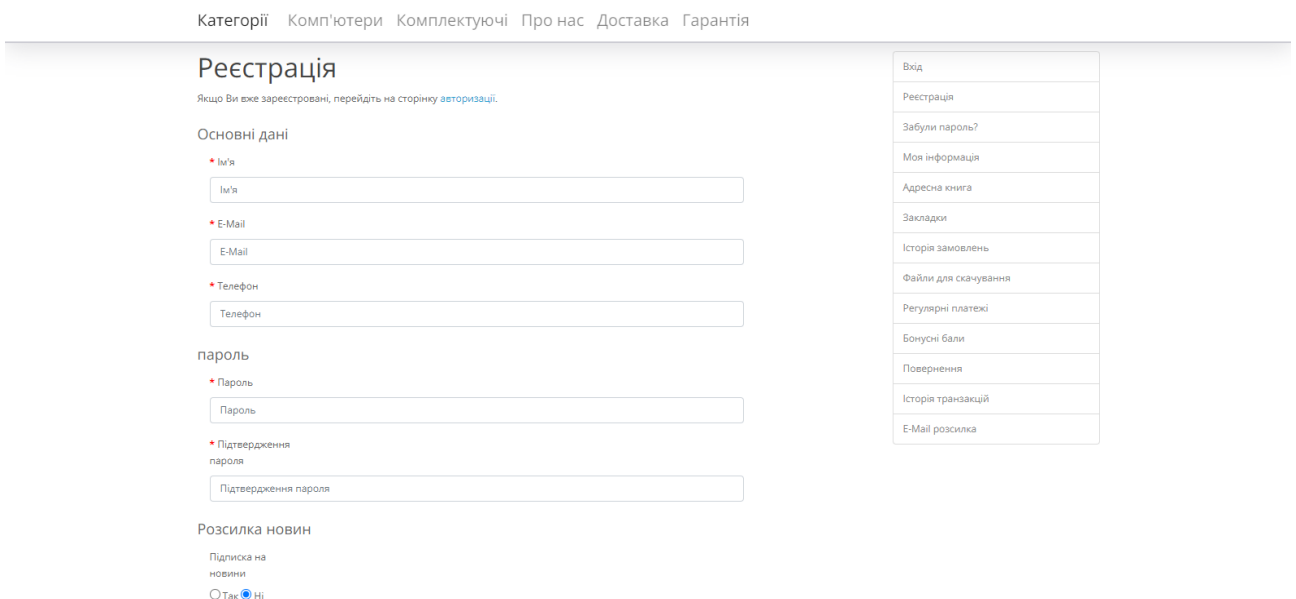


Рис. 2.10. Реєстрація користувача

Якщо реєстрація пройшла успішно, то користувач отримає повідомлення про створення облікового запису (рис.2.11).

# Ваш обліковий запис створено!

Вітаємо! Ваш Особистий Кабінет був успішно створений.

Тепер Ви можете скористатися додатковими можливостями: перегляд історії замовлень, друк рахунку, редагування своєї контактної інформації та адреси доставки та багато іншого.

Якщо у Вас є які-небудь питання, [пишіть нам](#).

[Продовжити](#)

Рис. 2.11. Повідомлення про створення облікового запису

Після додавання товару до кошика користувач має можливість оформити замовлення (рис.2.12).

Категорії [Комп'ютери](#) [Комплектуючі](#) [Про нас](#) [Доставка](#) [Гарантія](#)

## Оформлення замовлення

Крок 1: Спосіб оформлення замовлення ▾

### Новий покупець

Варіанти Оформлення замовлення

Реєстрація

Оформити замовлення без реєстрації

Створення облікового запису допоможе робити покупки швидше і зручніше, а так само отримувати знижки як постійний покупець.

[Продовжити](#)

Крок 2: Платіжна інформація ▾

### Особисті дані

\* Ім'я

\* E-Mail

\* Телефон

Крок 3: Підтвердження замовлення [Продовжити](#)

### Зареєстрований користувач

Я робив тут покупки і раніше реєструвався

E-Mail

Пароль

[Забули пароль?](#)

[Увійти](#)

Адреса

\* Адреса

Рис. 2.12. Сторінка оформлення замовлення

Після вибору способу оформлення замовлення користувач попадає на другий крок (рис.2.13).

Крок 2: Платіжна інформація ▾

Особисті дані

\* Ім'я

Олександр

\* E-Mail

polovynkaa@emeil.com

\* Телефон

65656565

Адреса

\* Адреса

Dnipro

Крок 3: Підтвердження замовлення

Продовжити

Рис. 2.13. Введення особистих даних

Після вводу особистих даних користувач потрапляє на крок підтвердження замовлення (рис.2.14).

Крок 3: Підтвердження замовлення ▾

Продовжити

Назва товару	Модель	Кількість	Ціна	Разом
Оптимальний Ігровий Athlon X4 970 RX 550 8Gb	Athlon X4 970 RX 550 8Gb	1	€12245.00грн	€12245.00грн
			<b>Сума:</b>	€12245.00грн
			<b>Разом:</b>	€12245.00грн

Після підтвердження наш менеджер зв'яжеться з вами для уточнення деталей замовлення

Підтвердження замовлення

Рис. 2.14. Крок підтвердження замовлення

Коли користувач підтверджує замовлення він отримує повідомлення про прийняття замовлення (рис.2.15).

Ваше замовлення прийняте!

Ваше замовлення прийняте!

Якщо у Вас виникли питання, будь ласка зв'яжіться з нами.

Дякуємо за покупки в нашому інтернет-магазині!

Продовжити

Рис. 2.15. Повідомлення про прийняття замовлення

Сторінка товару має наступний вигляд (рис.2.16).

# THE PC STORE



0 товар(ів) - 80.00грн

Категорії    Комп'ютери    Комплектуючі    Про нас    Доставка    Гарантія



Опис    Характеристики    Відгуків (0)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras vitae metus quis eros pharetra dignissim nec ac purus. In quis metus magna. Morbi imperdiet leo est. Proin sit amet dolor nec magna maximus scelerisque. Donec tincidunt ornare odio sed luctus. Sed vel aliquam orci. Etiam sollicitudin pellentesque finibus. In vel sem ultricies massa ultricies ornare in eget felis. Suspendisse accumsan malesuada nisl ac accumsan. Sed at turpis



Оптимальний  
Ігровий R3 1200 GT  
1030 8Gb

Модель: R3 1200 GT 1030 8Gb

Навність: Є в наявності

€11999.00грн

Кількість

Купити

З'яжіться з нами для консультації

☆☆☆☆☆ 0 відгуків / Написати відгук

Рис. 2.16. Сторінка товару

На сторінці товару користувач має можливість дізнатися номер телефону та перейти на сторінку зворотного зв'язку (рис.2.17) (рис.2.18).

З'яжіться з нами для консультації

Рис. 2.17. Кнопка зворотного зв'язку

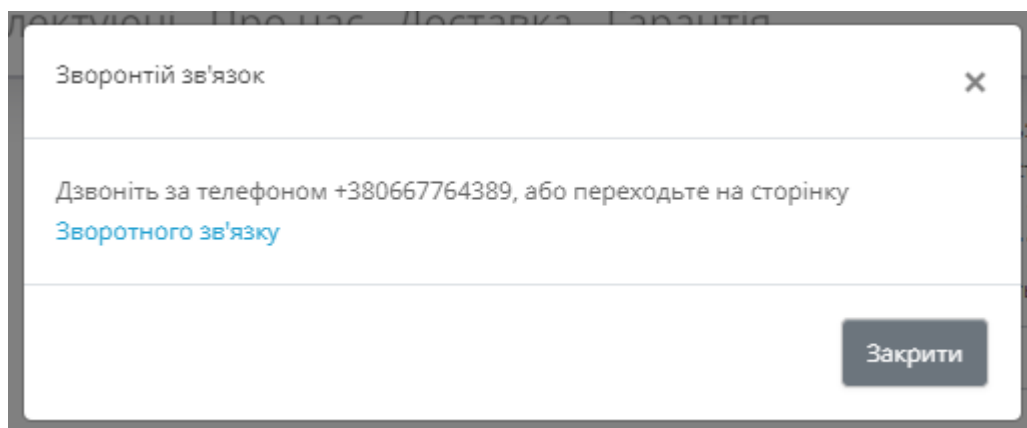


Рис. 2.18. Модальне вікно з номером телефону та посиланням на сторінку зворотного зв'язку

## РОЗДІЛ 3

### ЕКОНОМІЧНИЙ РОЗДІЛ

#### 3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 600;
2. коефіцієнт складності програми – 1,35;
3. коефіцієнт корекції програми в ході її розробки – 0,05;
4. годинна заробітна плата програміста – 132 грн/год;

Середня годинна заробітна плата Junior PHP розробника вираховувалась з даних сайту dou.ua [21]. Середнє значення заробітної плати розробника такого типу з досвідом роботи менш ніж один рік дорівнює 790 доларів США у місяць. За даними НБУ, на початок червня 2022 року один долар США дорівнює 29,53 грн. Виходячи з цих даних – зарплата в гривнях приблизно дорівнює 23300 грн. При стандартному восьмигодинному робочому дні (176 робочих годин в місяць) середня годинна заробітна плата розробника становить 132 грн.

5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,35;
6. коефіцієнт кваліфікації програміста, обумовлений стажем роботи з даної спеціальності – 0,8;
7. кількість виконавців – 1;
8. вартість машино-години ЕОМ – 14 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_d, \text{ людино-годин,} \quad (3.1)$$

де  $t_o$  – витрати праці на підготовку й опис поставленої задачі (приймається 50);

$t_u$  – витрати праці на дослідження алгоритму рішення задачі;

$t_a$  – витрати праці на розробку блок-схеми алгоритму;

$t_n$  – витрати праці на програмування по готовій блок-схемі;

$t_{омл}$  – витрати праці на налагодження програми на ЕОМ;

$t_o$  – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p), \quad (3.2)$$

де  $q$  – передбачуване число операторів;

$C$  – коефіцієнт складності програми;

$p$  – коефіцієнт корекції програми в ході її розробки.

Звідси умовне число операторів в програмі:

$$Q = 600 \cdot 1,35 \cdot (1 + 0,05) = 850,5;$$

Витрати праці на вивчення опису задачі  $t_u$  визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \dots 85) \cdot K}, \text{ людино-годин} \quad (3.3)$$

де  $B$  – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

$K$  – коефіцієнт кваліфікації програміста, обумовлений стажем роботи з даної спеціальності;

$$t_u = \frac{850.5 \cdot 1,35}{76 \cdot 0,8} = 18,8 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20 \dots 25) \cdot K}; \quad (3.4)$$

$$t_a = \frac{850.5}{25 \cdot 0,8} = 42,5 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \dots 25) \cdot K}; \quad (3.5)$$

$$t_n = \frac{850.5}{21 \cdot 0,8} = 47,9 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

–за умови автономного налагодження одного завдання:

$$t_{отл} = \frac{Q}{(4 \dots 5) \cdot K}; \quad (3.6)$$

$$t_{отл} = \frac{850.5}{5 \cdot 0,8} = 212,6 \text{ людино-годин}$$

–за умови комплексного налагодження завдання:

$$t_{отл}^K = 1,5 \cdot t_{отл}; \quad (3.7)$$

$$t_{отл}^K = 1,5 \cdot 212,6 = 318,9 \text{ людино-годин}$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial p} + t_{\partial o}; \quad (3.8)$$



де  $t_{\partial p}$  – трудомісткість підготовки матеріалів і рукопису;

$$t_{\partial p} = \frac{q}{(15 \dots 20) \cdot K}; \quad (3.9)$$

$$t_{\partial p} = \frac{850,5}{18 \cdot 0,8} = 59 \text{ людино-годин,}$$

де  $t_{\partial o}$  – трудомісткість редагування, печатки й оформлення документації;

$$t_{\partial o} = 0,75 \cdot t_{\partial p}; \quad (3.10)$$

$$t_{\partial o} = 0,75 \cdot 59 = 44,25, \text{ людино-годин.}$$

$$t_{\partial} = 59 + 44,25 = 103,25, \text{ людино-годин.}$$

Отримаємо трудомісткість розробки програмного забезпечення:

$$t = 50 + 18,8 + 42,5 + 47,9 + 212,6 + 103,25 = 475,05 \text{ людино-годин.}$$

Результатом розрахунків є те, що для розробки цього програмного забезпечення треба 475,05 людино-годин.

### 3.2. Рахунок витрат на створення програми

Витрати на створення ПЗ Кпо включають витрати на заробітну плату виконавця програми  $Z_{ЗП}$  і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{по} = Z_{ЗП} + Z_{МВ}, \text{ грн,} \quad (3.11)$$

$Z_{ЗП}$  – заробітна плата виконавців, яка визначається за формулою:

$$Z_{ЗП} = t \cdot C_{ПР}, \text{ грн,} \quad (3.12)$$

де  $t$  – загальна трудомісткість, людино-годин;

$C_{ПР}$  – середня годинна заробітна плата програміста, грн/година.

$$Z_{ЗП} = 475,05 \cdot 132 = 62706,6 \text{ грн.}$$

Вартість машинного часу  $Z_{МВ}$ , необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{МВ} = t_{омл} \cdot C_{М}, \text{ грн,} \quad (3.13)$$

де  $t_{омл}$  – трудомісткість налагодження програми на ЕОМ, год;

$C_{МЧ}$  – вартість машино-години ЕОМ, грн/год.

$$Z_{МВ} = 212,6 \cdot 14 = 2976,4 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{ПО} = 62706,6 + 2976,4 = 65683 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс.} \quad (3.14)$$

де  $B_k$  – число виконавців;

$F_p$  – місячний фонд робочого часу (при 40 годинному робочому тижні  $F_p=176$  годин).

Витрати на створення програмного продукту:

$$T = \frac{475,05}{1 \cdot 176} = 2,69 \text{ міс.}$$

**Висновок:** За допомогою даних розрахунків була отримана кінцева вартість розробленого веб-додатку – 65683 грн. Також був розрахований ймовірний термін розробки – 2,69 міс. Отриманий під час розрахунків часовий термін включає в себе час необхідний для розробки алгоритму розв’язання поставленої задачі, розробки бекенд та фронтенд частини веб додатку, а також час необхідний для створення документації. На процес розробки даного веб-додатку витрачено 475,05 людино-годин.

## ВИСНОВКИ

Метою даної кваліфікаційної роботи бакалавра являється розробка веб-додатку для Інтернет магазину з продажу персональних комп'ютерів на базі CMS OpenCart.

Веб-додаток являється веб-сайтом інтернет-магазину, який в підсумку призначений для:

- оптимізації бізнес-процесів, які в свою чергу призводять до зростання ефективності цього бізнесу;
- надання клієнтам сайту комфортного та зручного інтерфейсу;
- надання зручного інструментарію для модерування інтернет-магазину;
- надання можливості проводити торгові операції в інтернет-просторі.

Для реалізації необхідного функціоналу був створений адаптивний та зручний дизайн сайту; розроблена зручна реєстрація, оформлення замовлення та навігація; був встановлений та дороблений модуль фільтрації та сортування за параметрами.

Останнім часом актуальність розробки подібних веб-додатків для діяльності в сфері електронної комерції не підлягає ніяким сумнівам по тій причині, що в цій галузі існує немалий попит.

Створений веб-додаток призначений для використання в бізнесі, який займається торгівлею персональними комп'ютерами.

Розроблений веб-додаток повинен надати таким підприємствам можливість спрощення та оптимізації їх торгівельної діяльності, а також підвищити результативність ведення бізнесу. В такому випадку компанія стане більш конкурентною на ринку.

Даний проект був створений за допомогою мови програмування PHP. СУБД системи являється MySQL. В якості технологій для верстання сторінок використовувались HTML, CSS та JS з фреймворком Bootstrap та бібліотекою jQuery.

Під час процесу розробки сайту на локальному персональному комп'ютері використовувався локальний сервер Openserver. Процес розробки проходив в IDE PhpStorm та в текстовому редакторі Sublime Text.

В Економічному розділі була визначена трудомісткість розробки веб-додатку (475,05 людино-годин), а також були підраховані витрати на створення веб-додатку (65683 грн.) і період розробки (2,69 місяці).

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Need Analysis: Portable Web Server Development Kits for Teaching and Learning / Nor Masharah Husain, Nadia Akma Ahmad Zaki, Rasyidi Johan, Noor Anida Zaria Mohd Noor // International Journal of Multimedia & Its Applications. 2021. Vol. 13, No. 3. – 1 p.
2. Вовк М. Особливості розробки Web-сайтів [Текст] / М. Вовк // Збірник наукових праць викладачів, аспірантів, магістрантів і студентів фізико-математичного факультету. – Полтава: АСМІ, 2009. – 173 с.
3. Технологія «Клієнт – сервер». [Електронний ресурс] / Клієнт — сервер — Вікіпедія — Режим доступу: [https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82\\_%E2%80%94%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_%E2%80%94%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) – 05.05.2018 .
4. Гаврилюк О. Комунікативні стратегії і тактики взаємодії продавця з покупцем на спортивних сайтах / Гаврилюк О., Нікіфорова Є. //Актуальні питання гуманітарних наук: міжвузівський збірник наукових праць молодих вчених Дрогобицького державного педагогічного університету імені Івана Франка / [редактори упорядники М. Пантук, А. Душний, І. Зимомря]. – Дрогобич: Видавничий дім «Гельветика», 2020. – Вип. 34. Том 1. – 280 с.
5. Тардаскіна Т.М. Електронна комерція : навч. посіб. / Тардаскіна Т.М., Стрельчук Є.М., Терешко Ю.В. – Одеса : ОНАЗ ім. О.С. Попова, 2011. – 24 с. – ISBN 978-966-7598-66-2.
6. Євтушенко Д.Д. Електронний бізнес, електронна комерція, інтернет-торгівля: сутність та взаємозв'язок понять / Д.Д. Євтушенко // Бизнес Информ. – 2014. – № 8. – С. 187.
7. Ілляшенко, С.М. Перспективи та основні проблеми розвитку інтернет-торгівлі в Україні [Текст] / С.М. Ілляшенко, Т.Є. Іванова // Механізм регулювання економіки. - 2014. - № 3. - С. 72.

8. Бондаренко К. Системи управління вмістом / К. Бондаренко ; наук. керівник Микола Кравець // Інформаційні технології в агробізнесі та аграрній освіті : тези доп. VIII Всеукр. наук.-практ. конф. ( Дніпро, 22-24 квіт. 2020 р.) / ННІЕ ДДАЕУ. – Дніпро : ДДАЕУ, 2020. – С. 42-43. – Присвяч. 100-річчю ДДАЕУ. – Режим доступу : <http://dspace.dsau.dp.ua/jspui/handle/123456789/4383>

9. Методичні вказівки для виконання курсової роботи з дисципліни "Архітектура та проектування програмного забезпечення. Ч. 2" [Електронний ресурс] : для студентів спец. 121 "Інженерія програмного забезпечення" / уклад.: Д. Л. Орловський, А. М. Копп, О. Ю. Чередніченко ; Нац. техн. ун-т "Харків. політехн. ін-т". – Електрон. текст. дані. – Харків, 2020. – 11 с. – URI: <http://repository.kpi.kharkov.ua/handle/KhPI-Press/48508>.

10. Вимоги до програмного забезпечення [Електронний ресурс] / Вимоги до програмного забезпечення — Вікіпедія — Режим доступу: <https://w.wiki/5Emw> .– 05.05.2018 .

11. Акшенцев М. О. Метод двофакторної автентифікації на CMS WordPress : пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 125 - Кібербезпека / М. О. Акшенцев ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2021. – 8 с.

12. Боляк М. В. Розробка програмного забезпечення для розвитку інституту сталого розвитку при ЧНУ імені Петра Могили : автореф. кваліфік. роботи на здобуття освітнього ступеня «бакалавр» : спец. 122 «Комп'ютерні науки та інформаційні технології» / М. В. Боляк , ЧНУ ім. Петра Могили. – Миколаїв, 2021. – 9-10 с.

13. Железняк, С.О. Інтернет-магазин дизайнерських меблів [Текст]: робота на здобуття кваліфікаційного ступеня бакалавра; спец.: 122 - комп'ютерні науки (інформатика) / С.О. Железняк; наук. кер. О.Б. Проценко. – Суми: СумДУ, 2021. – 16-18 с.

14. Щекотов Р. В. Створення автоматизованої системи управління персоналом : автореф. кваліфік. роботи на здобуття освітнього ступеня

«бакалавр» : спец. 122 «Комп'ютерні науки та інформаційні технології» / Р. В. Щекотов, ЧНУ ім. Петра Могили. – Миколаїв, 2021. – 5-6 с.

15. Вострікова М. В. Інформаційно-пошукова система електронної реєстрації приватних дошкільних дитячих закладів : автореф. кваліфік. роботи на здобуття освітнього ступеня «бакалавр» : напрям підготовки 6.050101 «Комп'ютерні науки» / М. В. Вострікова, ЧНУ ім. Петра Могили. – Миколаїв, 2019. – 8 с.

16. Білецький В. Р. Інформаційна система для модуля програми Жана Моне : автореф. кваліфік. роботи на здобуття освітнього ступеня «бакалавр» : напрям підготовки 6.050101 «Комп'ютерні науки» / В. Р. Білецький, ЧНУ ім. Петра Могили. – Миколаїв, 2019. – 3-4 с.

17. Мальченко П. М., Дворецький М. Л., Іванов Г. О. Застосування інформаційних технологій в освіті створенні зручного застосунку для проведення контролю успішності учнів для шкіл, та вирішення деяких проблем з уже існуючими системами для університетів // Перспективна техніка і технології – 2021 : матеріали XVII міжнар. наук.-практ. конф. молодих учених, аспірантів і студентів, 21-23 вересня 2021 р., м. Миколаїв. Т. 1 / Міністерство освіти і науки України ; Миколаївський національний аграрний університет. Миколаїв : МНАУ, 2021. С. 9.

18. Скуба, О.В. Реалізація лендінг пейдж з продажу інформаційних продуктів за допомогою фрейворків JQuery [Текст]: робота на здобуття кваліфікаційного ступеня бакалавра; спец.: 122 - комп'ютерні науки (інформатика) / О.В. Скуба; наук. кер. С.П. Шаповалов. – Суми: СумДУ, 2021. – 12-20 с.

19. Основи об'єктно-орієнтованого програмування на мові JAVA : метод. вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування» для підготовки бакалаврів у галузі знань «Інформаційні технології» / М. Л. Дворецький, С. Ю. Боровльова, Ю. О. Нездолій, С. В. Дворецька. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2019. – 5 с. – (Методична серія ; вип. 268).



20. Шидула, О. М. Вступ до сучасного Web-дизайну: HTML5+CSS3 : навчальний посібник / О. М. Шидула – Київ : ПЦДО, 2019. – 16 с.

21. Середня місячна заробітна плата Junior PHP Software Engineer [Електронний ресурс] / Середня місячна заробітна плата Junior PHP Software Engineer — jobs.dou.ua – Режим доступу: <https://jobs.dou.ua/salaries/?period=2021-12&position=Junior%20SE&technology=PHP&experience=0> .– 05.05.2018.

## КОД ПРОГРАМИ

## Лістинг коду файлу header.tpl

```

<!DOCTYPE html>
<html dir="<?php echo $direction; ?>" lang="<?php echo $lang; ?>">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>
    <?php echo $title; ?>
  </title>
  <base href="<?php echo $base; ?>" />
  <?php if ($description) { ?>
    <meta name="description" content="<?php echo $description; ?>" />
  <?php } ?>
  <?php if ($keywords) { ?>
    <meta name="keywords" content="<?php echo $keywords; ?>" />
  <?php } ?>

  <script src="catalog/view/theme/default/js/jquery-3.6.0.min.js"
type="text/javascript"></script>
  <link href="catalog/view/theme/default/stylesheet/bootstrap.min.css" rel="stylesheet"
media="screen" />
  <script src="catalog/view/theme/default/js/popper.js" type="text/javascript"></script>
  <script src="catalog/view/theme/default/js/bootstrap.min.js" type="text/javascript"></script>
  <script src="catalog/view/theme/default/js/main.js" type="text/javascript"></script>
  <link href="catalog/view/javascript/font-awesome/css/font-awesome.min.css"
rel="stylesheet" type="text/css" />
  <link href="//fonts.googleapis.com/css?family=Open+Sans:400,400i,300,700"
rel="stylesheet" type="text/css" />
  <link href="catalog/view/theme/default/stylesheet/stylesheet.css" rel="stylesheet">

  <?php foreach ($styles as $style) { ?>
    <link href="<?php echo $style['href']; ?>" type="text/css" rel="<?php echo $style['rel']; ?>"
media="<?php echo $style['media']; ?>" />
  <?php } ?>
  <script src="catalog/view/javascript/common.js" type="text/javascript"></script>
  <?php foreach ($links as $link) { ?>
    <link href="<?php echo $link['href']; ?>" rel="<?php echo $link['rel']; ?>" />
  <?php } ?>
  <?php foreach ($scripts as $script) { ?>
    <script src="<?php echo $script; ?>" type="text/javascript"></script>
  <?php } ?>
  <?php foreach ($analytics as $analytic) { ?>
    <?php echo $analytic; ?>
  <?php } ?>
</head>

```

```

<body class="<?php echo $class; ?>">
  <nav id="top" class="gotohumbout">
    <div class="container">
      <div id="top-links">
        <ul class="nav row align-items-center">

          <li class="col ">
            <a href="<?php echo $contact; ?>">Зворотній зв'язок</a>
          </li>
          <li class="dropdown col"><a class="nav-link" href="<?php echo $account; ?>" title="<?php
echo $text_account; ?>" class="dropdown-toggle" data-toggle="dropdown"><span class="hidden-
xs hidden-sm hidden-md"><?php echo $text_account; ?></span> <span class="caret"></span></a>
            <ul class="dropdown-menu ">

              <?php if ($logged) { ?>
                <li>
                  <a class="nav-link fix-link-dropdown-see-all" href="<?php echo $account; ?>">
                    <?php echo $text_account; ?>
                  </a>
                </li>
                <li>
                  <a class="nav-link fix-link-dropdown-see-all" href="<?php echo $order; ?>">
                    <?php echo $text_order; ?>
                  </a>
                </li>
                <li>
                  <a class="nav-link fix-link-dropdown-see-all" href="<?php echo $transaction; ?>">
                    <?php echo $text_transaction; ?>
                  </a>
                </li>
                <li>
                  <a class="nav-link fix-link-dropdown-see-all" href="<?php echo $download; ?>">
                    <?php echo $text_download; ?>
                  </a>
                </li>
                <li>
                  <a class="nav-link fix-link-dropdown-see-all" href="<?php echo $logout; ?>">
                    <?php echo $text_logout; ?>
                  </a>
                </li>
              <?php }
              else { ?>
                <li>
                  <a class="nav-link fix-link-dropdown-see-all" href="<?php echo $register; ?>">
                    <?php echo $text_register; ?>
                  </a>
                </li>
                <li>
                  <a class="nav-link fix-link-dropdown-see-all" href="<?php echo $login; ?>">
                    <?php echo $text_login; ?>
                  </a>
                </li>
              }
            }
          }
        </ul>
      </div>
    </div>
  </nav>

```

```

        <?php } ?>

    </ul>
</li>
<li class="col ">
    <a href="<?php echo $wishlist; ?>" id="wishlist-total" title="<?php echo $text_wishlist;
?>">
        <span class="hidden-xs hidden-sm hidden-md">
            <?php echo $text_wishlist; ?>
        </span>
    </a>
</li>
<li class="col ">
    <a href="<?php echo $shopping_cart; ?>" title="<?php echo $text_shopping_cart; ?>">
        <span class="hidden-xs hidden-sm hidden-md">
            <?php echo $text_shopping_cart; ?>
        </span>
    </a>
</li>
<li class="col ">
    <a href="<?php echo $checkout; ?>" title="<?php echo $text_checkout; ?>">
        <span class="hidden-xs hidden-sm hidden-md">
            <?php echo $text_checkout; ?>
        </span>
    </a>
</li>

</ul>
</div>
</div>

</nav>

<header>
<div class="container">
    <div class="row">
        <div class="col-sm-12 d-flex justify-content-center">
            <div id="logo">
                <?php if ($logo) { ?>
                    <a href="<?php echo $home; ?>">" alt="<?php echo $name; ?>" class="img-responsive" /></a>
                <?php } else { ?>
                    <h1><a href="<?php echo $home; ?>"><?php echo $name; ?></a></h1>
                <?php } ?>
            </div>
        </div>
        <div class="col-md-8 d-flex justify-content-center">
            <?php echo $search; ?>
        </div>
        <div class="col-md-4 d-flex justify-content-center">
            <?php echo $cart; ?>
        </div>
    </div>
</div>

```

```

</div>
</div>
</header>

<?php if ($categories) { ?>
<div class="navbarstyle">

<! -- Навігаційна панель -->
<nav class="navbar navbar-expand-md navbar-light">
<div class="container">
<a class="navbar-brand" href="#">
<?php echo $text_category; ?>
</a>

<! -- Кнопка згорнути -->
<button class="navbar-toggler toggler-example" type="button" data-toggle="collapse" data-
target="#navbarContent" aria-controls="navbarContent" aria-expanded="false" aria-label="Toggle
navigation">
<span class="dark-blue-text">
<i class="fa fa-bars"></i>
</span>
</button>

<! -- КОНТЕНТ ЩО ЗГОРТАЄТЬСЯ-->
<div class="collapse navbar-collapse" id="navbarContent">
<ul class="nav navbar-nav mr-auto">
<?php foreach ($categories as $category) { ?>
<?php if ($category['children']) { ?>
<li class="dropdown nav-item">
<a class="nav-link" href="<?php echo $category['href']; ?>" class="dropdown-toggle
nav-link" data-toggle="dropdown">
<?php echo $category['name']; ?>
</a>
<div class="dropdown-menu dropdown-menu-customize">

<div class="dropdown-inner">
<?php foreach (array_chunk($category['children'], ceil(count($category['children']) /
$category['column'])) as $children) { ?>
<ul class="list-unstyled">
<?php foreach ($children as $child) { ?>
<li class="nav-item">
<a class="nav-link" href="<?php echo $child['href']; ?>">
<?php echo $child['name']; ?>
</a>
</li>
<?php } ?>
</ul>
<?php } ?>
</div>

<a href="<?php echo $category['href']; ?>" class="nav-link fix-link-dropdown-see-
all">

```

```

        <?php echo $text_all; ?>
        <?php echo $category['name']; ?>
    </a>
</div>
</li>
<?php } else { ?>
    <li class="nav-item">
        <a class="nav-link" href="<?php echo $category['href']; ?>">
            <?php echo $category['name']; ?>
        </a>
    </li>
<?php } ?>
<?php } ?>
    <?php foreach ($informations as $information) { ?>
        <li class="nav-item">
            <a href="<?php echo $information['href']; ?>" class="nav-link">
                <?php echo $information['title']; ?>
            </a>
        </li>
    <?php } ?>

    <div class="nav-inner-menu">
        <li class="gotohumbinside nav-item"><a class="nav-link" href="<?php echo
$contact; ?>">Зворотній зв'язок</a></li>
        <li class="dropdown gotohumbinside nav-item"><a class="nav-link"
href="<?php echo $account; ?>" title="<?php echo $text_account; ?>" class="dropdown-toggle"
data-toggle="dropdown"><span class="hidden-xs hidden-sm hidden-md"><?php echo
$text_account; ?></span> <span class="caret"></span></a>
        <ul class="dropdown-menu ">
            <?php if ($logged) { ?>

                <li>
                    <a class="nav-link" href="<?php echo $account; ?>">
                        <?php echo $text_account; ?>
                    </a>
                </li>

                <li>
                    <a class="nav-link" href="<?php echo $order; ?>">
                        <?php echo $text_order; ?>
                    </a>
                </li>

                <li>
                    <a class="nav-link" href="<?php echo $transaction; ?>">
                        <?php echo $text_transaction; ?>
                    </a>
                </li>

                <li>
                    <a class="nav-link" href="<?php echo $download; ?>">
                        <?php echo $text_download; ?>
                    </a>
                </li>
            <?php } ?>
        </ul>
    </div>

```

```

        </a>
    </li>

    <li>
        <a class="nav-link" href="<?php echo $logout; ?>">
            <?php echo $text_logout; ?>
        </a>
    </li>

    <?php } else { ?>

    <li>
        <a class="nav-link fix-link-dropdown-see-all" href="<?php echo $register;
?>">
            <?php echo $text_register; ?>
        </a>
    </li>

    <li>
        <a class="nav-link fix-link-dropdown-see-all" href="<?php echo $login;
?>">
            <?php echo $text_login; ?>
        </a>
    </li>

    <?php } ?>
</ul>

</li>
<li class="gotohumbinside nav-item">
    <a class="nav-link" href="<?php echo $wishlist; ?>" id="wishlist-total"
title="<?php echo $text_wishlist; ?>">
        <span class="hidden-xs hidden-sm hidden-md"><?php echo $text_wishlist;
?></span>
    </a>
</li>
<li class="gotohumbinside nav-item">
    <a class="nav-link" href="<?php echo $shopping_cart; ?>" title="<?php echo
$text_shopping_cart; ?>">
        <span class="hidden-xs hidden-sm hidden-md"><?php echo
$text_shopping_cart; ?></span>
    </a>
</li>
<li class="gotohumbinside nav-item">
    <a class="nav-link" href="<?php echo $checkout; ?>" title="<?php echo
$text_checkout; ?>">
        <span class="hidden-xs hidden-sm hidden-md"><?php echo $text_checkout;
?></span>
    </a>
</li>
</div>

```

```

        </ul>
    </div>
</div>
</nav>
</div>
<?php } ?>

```

### Лістинг коду файлу header.php

```

<?php
class ControllerCommonHeader extends Controller {
    public function index() {
        // Analytics
        $this->load->model('extension/extension');

        $data['analytics'] = array();

        $analytics = $this->model_extension_extension->getExtensions('analytics');

        foreach ($analytics as $analytic) {
            if ($this->config->get($analytic['code'] . '_status')) {
                $data['analytics'][] = $this->load->controller('extension/analytics/' .
                $analytic['code'], $this->config->get($analytic['code'] . '_status'));
            }
        }

        if ($this->request->server['HTTPS']) {
            $server = $this->config->get('config_ssl');
        } else {
            $server = $this->config->get('config_url');
        }

        if (is_file(DIR_IMAGE . $this->config->get('config_icon'))) {
            $this->document->addLink($server . 'image/' . $this->config->
            >get('config_icon'), 'icon');
        }

        $data['title'] = $this->document->getTitle();

        $data['base'] = $server;
        $data['description'] = $this->document->getDescription();
        $data['keywords'] = $this->document->getKeywords();
        $data['links'] = $this->document->getLinks();
        $data['styles'] = $this->document->getStyles();
        $data['scripts'] = $this->document->getScripts();
        $data['lang'] = $this->language->get('code');
        $data['direction'] = $this->language->get('direction');
        $data['name'] = $this->config->get('config_name');

        if (is_file(DIR_IMAGE . $this->config->get('config_logo'))) {
            $data['logo'] = $server . 'image/' . $this->config->get('config_logo');
        } else {
            $data['logo'] = "";
        }
    }
}

```



```

$this->load->language('common/header');

$data['text_home'] = $this->language->get('text_home');

// Wishlist
if ($this->customer->isLoggedIn()) {
    $this->load->model('account/wishlist');

    $data['text_wishlist'] = sprintf($this->language->get('text_wishlist'), $this-
>model_account_wishlist->getTotalWishlist());
} else {
    $data['text_wishlist'] = sprintf($this->language->get('text_wishlist'),
(isset($this->session->data['wishlist']) ? count($this->session->data['wishlist']) : 0));
}

$data['text_shopping_cart'] = $this->language->get('text_shopping_cart');
$data['text_logged'] = sprintf($this->language->get('text_logged'), $this->url-
>link('account/account', "", true), $this->customer->getFirstName(), $this->url-
>link('account/logout', "", true));

$data['text_account'] = $this->language->get('text_account');
$data['text_register'] = $this->language->get('text_register');
$data['text_login'] = $this->language->get('text_login');
$data['text_order'] = $this->language->get('text_order');
$data['text_transaction'] = $this->language->get('text_transaction');
$data['text_download'] = $this->language->get('text_download');
$data['text_logout'] = $this->language->get('text_logout');
$data['text_checkout'] = $this->language->get('text_checkout');
$data['text_category'] = $this->language->get('text_category');
$data['text_all'] = $this->language->get('text_all');

$data['home'] = $this->url->link('common/home');
$data['wishlist'] = $this->url->link('account/wishlist', "", true);
$data['logged'] = $this->customer->isLoggedIn();
$data['account'] = $this->url->link('account/account', "", true);
$data['register'] = $this->url->link('account/register', "", true);
$data['login'] = $this->url->link('account/login', "", true);
$data['order'] = $this->url->link('account/order', "", true);
$data['transaction'] = $this->url->link('account/transaction', "", true);
$data['download'] = $this->url->link('account/download', "", true);
$data['logout'] = $this->url->link('account/logout', "", true);
$data['shopping_cart'] = $this->url->link('checkout/cart');
$data['checkout'] = $this->url->link('checkout/checkout', "", true);
$data['contact'] = $this->url->link('information/contact');
$data['telephone'] = $this->config->get('config_telephone');

$this->load->model('catalog/information');

$data['informations'] = array();

foreach ($this->model_catalog_information->getInformations() as $result) {

```

```

    if ($result['bottom']) {
        $data['informations'][] = array(
            'title' => $result['title'],
            'href' => $this->url->link('information/information', 'information_id=' .
$result['information_id'])
        );
    }
}

// Menu
$this->load->model('catalog/category');

$this->load->model('catalog/product');

$data['categories'] = array();

$categories = $this->model_catalog_category->getCategories(0);

foreach ($categories as $category) {
    if ($category['top']) {
        // Level 2
        $children_data = array();

        $children = $this->model_catalog_category-
>getCategories($category['category_id']);

        foreach ($children as $child) {
            $filter_data = array(
                'filter_category_id' => $child['category_id'],
                'filter_sub_category' => true
            );

            $children_data[] = array(
                'name' => $child['name'] . ($this->config-
>get('config_product_count') ? ' (' . $this->model_catalog_product->getTotalProducts($filter_data) .
)': ''),
                'href' => $this->url->link('product/category', 'path=' .
$category['category_id'] . '_' . $child['category_id'])
            );
        }

        // Level 1
        $data['categories'][] = array(
            'name' => $category['name'],
            'children' => $children_data,
            'column' => $category['column'] ? $category['column'] : 1,
            'href' => $this->url->link('product/category', 'path=' .
$category['category_id'])
        );
    }
}

```

```

$data['language'] = $this->load->controller('common/language');
$data['currency'] = $this->load->controller('common/currency');
$data['search'] = $this->load->controller('common/search');
$data['cart'] = $this->load->controller('common/cart');

// For page specific css
if (isset($this->request->get['route'])) {
    if (isset($this->request->get['product_id'])) {
        $class = '-' . $this->request->get['product_id'];
    } elseif (isset($this->request->get['path'])) {
        $class = '-' . $this->request->get['path'];
    } elseif (isset($this->request->get['manufacturer_id'])) {
        $class = '-' . $this->request->get['manufacturer_id'];
    } elseif (isset($this->request->get['information_id'])) {
        $class = '-' . $this->request->get['information_id'];
    } else {
        $class = "";
    }

    $data['class'] = str_replace('/', '-', $this->request->get['route']) . $class;
} else {
    $data['class'] = 'common-home';
}

return $this->load->view('common/header', $data);
}
}

```



## ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файлу	Опис
Пояснювальні документи	
Диплом_Половинка.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Диплом_Половинка.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Web-application.rar	Архів. Містить директорію веб-додатку з усіма файлами
Презентація	
Презентація_Половинка.pptx	Презентація кваліфікаційної роботи