

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія з продажу комп'ютерної техніки»

Виконав: студент 2-го курсу, групи 2КН-22м
спеціальності 122 «Комп'ютерні науки»

(шифр назва напрямку підготовки, спеціальності)

Волков А.О.
(прізвище та ініціали)

Керівник: к.т.н., доц. каф. КН

Озеранський В.С.
(прізвище та ініціали)

«07» 12 2023 р.

Опонент: к.т.н., доц. каф. САІТ

Варчук І.В.
(прізвище та ініціали)

«07» 12 2023 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

« 08 12 » 2023 р.

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 – Інформаційні технології
Спеціальність – 122 – Комп'ютерні науки
Освітньо-професійна програма – Системи штучного інтелекту

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
д.т.н., проф. Яровий А.А.



(підпис)

“ 19 ” 08 2023 року

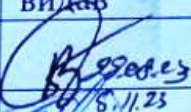
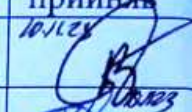

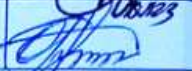
ЗАВДАННЯ
на магістрекську кваліфікаційну роботу
студенту
Волкову Андрію Олександровичу

- 1 Тема роботи: «Інформаційна технологія з продажу комп'ютерної техніки».
Керівник роботи: Озеранський Володимир Сергійович, к.т.н., доц. каф. КН
затверджені наказом вищого навчального закладу «18» 09 2023 року № 247
- 2 Строк подання студентом роботи 13.11.2023
- 3 Вихідні дані до роботи: кількість товарів - не менше 100шт; кількість користувачів для одночасного опрацювання - не менше 100чол; Мова програмування – об'єктно-орієнтована з можливістю маніпулювання даними і управління базами даних, сервер яких підтримує HTTP протокол.
- 4 Зміст пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз сучасного рівня розвитку технологій з продажу комп'ютерної техніки, розробка інформаційної технології з продажу комп'ютерної техніки, програмна реалізація інформаційної технології з продажу комп'ютерної техніки, економічна частина, висновки, перелік використаних джерел, додатки
- 5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): схема алгоритму функціонування інформаційної технології з продажу комп'ютерної техніки, головні вінка розробленого програмного забезпечення.

6 Консультанти розділів проекту (роботи)

Консультанти розділів роботи в таблиці 1.

Таблиця 1 - Консультанти роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Озеранський В. С., к.т.н., доц. каф. КН	 08.11.23	 10.11.23
4	Ратушняк О. Г., к.т.н, доц. каф. ЕПВМ		

7 Дата видачі завдання 29.08.2023 р.

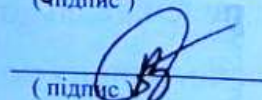
КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз сучасного рівня розвитку технологій з продажу комп'ютерної техніки	01.09.23 - 05.09.23	
2	Розробка інформаційної технології з продажу комп'ютерної техніки	06.09.23 - 16.09.23	
3	Програмна реалізація інформаційної технології з продажу комп'ютерної техніки	17.09.23 - 07.10.23	
4	тестування та аналіз результатів роботи розробленої програми	08.10.23 - 23.10.23	
5	Розробка інструкції користувача	24.10.23 - 04.11.23	
6	Оформлення матеріалів до захисту МКР	02.11.23 - 10.11.23	

Студент


(підпис)

Керівник роботи


(підпис)

Волков А.О.

(прізвище та ініціали)

Озеранський В

(прізвище та ініціали)

АНОТАЦІЯ

УДК 621.374.415

Волков А.О. Інформаційна технологія з продажу комп'ютерної техніки. Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма - Системи штучного інтелекту. Вінниця: ВНТУ, 2023. 112 с.

На укр. мові. Бібліогр.: 20 назв; рис.: 20; табл. 14.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології з продажу комп'ютерної техніки.

В магістерській кваліфікаційній роботі проаналізовано етапи процесу розробки сучасні інформаційні технології з продажу комп'ютерної техніки, проаналізовано існуючі інструменти веб-розробки, визначено актуальні підходи до веб-розробки та обгрунтовано доцільність їх використання під час розробки інформаційної технології з продажу комп'ютерної техніки. Наведенно переваги і недоліки існуючих програмних продуктів з продажу комп'ютерної техніки. Досліджено способи вирішення даної задачі.

Програмна реалізація інформаційної технології з продажу комп'ютерної техніки здійснена з використанням мов програмування JavaScript та PHP. Для розробки графічного інтерфейсу користувача були використані мови розмітки HTML5 та CSS3, середовище розробки Visual Studio Code, а також графічний редактор Adobe Photoshop 2020.

В розділі економічної частини проведено оцінювання комерційного потенціалу розробки інформаційної технології з продажу комп'ютерної техніки, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, які склали 770903,59 грн, розраховано період окупності – 2,4 року.

Ключові слова: інформаційна технологія, комп'ютер, інтернет-магазин, електронна комерція.

ABSTRACT

Volkov A.O. Information technology for the sale of computer equipment. Master's thesis on specialty 122 - computer science, educational program - Artificial intelligence systems. Vinnytsia: VNTU, 2023. 112 p.

In Ukrainian speech Bibliography: 20 titles; Fig.: 20; table 14.

This master's thesis is devoted to the development of information technology for the sale of computer equipment.

The master's thesis analyzed the stages of the development process of modern information technology for the sale of computer equipment, analyzed existing web development tools, determined current approaches to web development, and substantiated the feasibility of their use during the development of information technology for the sale of computer equipment. The advantages and disadvantages of existing software products for the sale of computer equipment are given. Ways to solve this problem have been studied.

Software implementation of information technology for the sale of computer equipment was carried out using JavaScript and PHP programming languages. The HTML5 and CSS3 markup languages, the Visual Studio Code development environment, and the Adobe Photoshop 2020 graphic editor were used to develop the graphical user interface.

In the section of the economic part, an assessment of the commercial potential of the development of information technology for the sale of computer equipment was carried out, the costs of carrying out scientific work and the implementation of the results were predicted, which amounted to UAH 770,903.59, and the payback period was calculated - 2.4 years.

Keywords: information technology, computer, online store, electronic commerce.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ СУЧАСНОГО РІВНЯ РОЗВИТКУ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ.....	7
1.1 Аналіз сучасних технологій інтернет-комерції	7
1.2 Аналіз існуючих програмних засобів з продажу комп'ютерної техніки	8
1.3 Постановка задачі.....	15
1.4 Висновок до розділу 1.....	15
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ	16
2.1 Аналіз інформаційного забезпечення сучасних веб-ресурсів	16
2.2 Розробка алгоритму роботи інформаційної технології з продажу комп'ютерної техніки	16
2.3 Розробка структури інформаційної технології з продажу комп'ютерної техніки	19
2.4 Розробка інтерфейсу та дизайну веб-ресурсу з продажу комп'ютерної техніки	22
2.5 Висновок до розділу 2.....	30
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ.....	31
3.1 Обґрунтування вибору мови програмування.....	31
3.2 Обґрунтування вибору середовища розробки	33
3.3 Робота системи керування базами даних для веб-ресурсу з продажу комп'ютерної техніки	36
3.4 Програмна реалізація інтерфейсу інформаційної технології з продажу комп'ютерної техніки	39
3.5 Тестування програмного забезпечення з продажу комп'ютерної техніки та аналіз результатів його роботи	51
3.6 Висновок до розділу 3.....	58

4 ЕКОНОМІЧНА ЧАСТИНА.....	59
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки	59
4.2 Визначення рівня конкурентоспроможності розробки	63
4.3 Розрахунок витрат на проведення науково-дослідної роботи.....	66
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором	76
4.5 Висновки до розділу 4	80
ВИСНОВКИ.....	81
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	83
ДОДАТКИ.....	85
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	86
Додаток Б (обов'язковий) Лістинг програми	87
Додаток В (обов'язковий) Ілюстративна частина.....	103
Додаток Г (довідниковий) Інструкція користувача	111

ВСТУП

Актуальність теми дослідження. У наш час поширення та доступність мережі Інтернет росте величезними темпами. Завдяки цьому онлайн-шопінг стає дедалі більше популярним. Це вид електронної комерції, де клієнт може придбати будь-який товар через мережу Інтернет. Інакше кажучи, це процес покупки або продажі товарів за допомогою комп'ютеризованих ділових операцій в електронному вигляді. Тому актуальною є розробка веб-системи як торговельного майданчика для продажу комп'ютерної техніки [1].

Інтернет – це чудова можливість для компаній, щоб вийти на світовий ринок, розширити свої канали збуту, постачальники і покупці об'єднуються в єдину систему. За останні роки темпи росту електронної комерції зросли в кілька разів. Торгові площі, ринки, в даний момент замінюються Інтернет-простором.

Основна мета будь-якого інтернет-магазину – залучення більшого числа потенційних покупців і отримання більшої кількості продажів із найменшими витратами бюджету на рекламу.

Насправді найбільш важливим завданням є не збільшення відвідуваності як такої, а покращення показника конверсії відвідувачів у покупців. І одним із перевірених і надійних способів підняти продажі є покращення юзабіліті вашого інтернет-магазину. Під терміном «юзабіліті» розуміють оцінку зручності та простоти використання певного інтерфейсу з боку користувача. Таким чином, якщо у вашому інтернет-магазині він складний або розуміння взаємодії з ним неочевидне для користувача, потенційні клієнти будуть залишати його, не здійснюючи покупки, а ви, у свою чергу, недоотримувати прибуток. Так можна прийти до простого, але правильного висновку: Краща якість юзабіліті = Більше продажів.

Звичайно, є певні природні обмеження, і для більшості інтернет-магазинів у СНД показник конверсії становить від 1 до 5%. Але завжди потрібно прагнути до кращого, і навіть збільшення її рівня на 0,5-1% для солідного продавця може обернутись істотним зростанням прибутку.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка спеціалізованих засобів штучного інтелекту на основі інтелектуального аналізу даних та машинного навчання» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження магістерської кваліфікаційної роботи є підвищення якості функціонування програмного забезпечення з продажу комп'ютерної техніки.

Для досягнення поставленої мети потрібно виконати наступні завдання:

1. Здійснити аналіз сучасних технологій з онлайн продажу комп'ютерної техніки;
2. Здійснити аналіз існуючих програмних рішень з онлайн продажу комп'ютерної техніки;
3. Розробити структуру інформаційної технології з продажу комп'ютерної техніки;
4. Розробити алгоритм функціонування інформаційної технології з продажу комп'ютерної техніки;
5. Виконати програмну реалізацію інформаційної технології з продажу комп'ютерної техніки;
6. Провести тестування програмного забезпечення та проаналізувати отримані результати.
7. Здійснити економічні розрахунки доцільності розробки нової інформаційної технології.

Об'єкт дослідження – процес продажу комп'ютерної техніки в інтернет-магазині.

Предмет дослідження – програмні засоби продажу комп'ютерної техніки в інтернет-магазині.

Методи дослідження. У роботі використано такі методи наукових досліджень: методи математичної статистики для обрахунків результатів досліджень щодо

ефективності роботи програмного продукту; методи об'єктно-орієнтованого програмування для автоматизації розрахунків.

Наукова новизна одержаних результатів полягає в:

– Удосконалено інформаційну технологію з продажу комп'ютерної техніки, що відрізняється від існуючих удосконаленою інформаційною моделлю, що дозволило підвищити якість функціонування програмного забезпечення з продажу комп'ютерної техніки.

Практичне значення одержаних результатів полягає в тому, що:

– розроблено алгоритм функціонування інформаційної технології з продажу комп'ютерної техніки.

– розроблено програмне забезпечення для продажу комп'ютерної техніки.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими підходами аналізу та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок здобувача. Усі результати, що наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, які написано у співавторстві, здобувачу належать: огляд технологій для он-лайн продажу комп'ютерної техніки [1].

Апробація результатів роботи. Результати роботи були апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2023)» (м. Вінниця, Україна, 2023 р.).

Публікації. За результатами магістерської кваліфікаційної роботи опубліковано 1 тезу доповіді на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2023)» [1].

1 АНАЛІЗ СУЧАСНОГО РІВНЯ РОЗВИТКУ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ

1.1 Аналіз сучасних технологій інтернет-комерції

Сучасна комп'ютерна техніка представляє одне з найзначніших досягнень людської думки, вплив якого на розвиток людства важко переоцінити. Область застосування комп'ютерів величезна і безперервно розширюється: управління виробництвом, транспорт і зв'язок, інформаційно-обчислювальна техніка, військова техніка, побутова техніка, навчання, авіація і космос, медицина, наукові дослідження, комунальне і міське господарство, банківський облік, метрологія і т.д.

Сьогодні процеси глобалізації охоплюють практично всі сфери людської діяльності: економіку, культуру, інформаційний простір, технології та менеджмент. Інтернет відіграє в цьому відношенні важливу роль. Наслідками цього процесу стали, наприклад, створення віртуальних компаній, поява нового типу медіа, можливість працювати з дому і, звичайно ж, розвиток електронної комерції [3].

Люди віддають перевагу Інтернет-магазинам через їх велику кількість переваг: постійні знижки, конфіденційність, у традиційних магазинах ми маємо обмежений вибір та невелике різноманіття товарів, клієнти можуть придбати лише те, що є наразі в наявності, тоді як у випадку Інтернет-шопінгу, покупець може робити замовлення з широкого асортименту товарів різних марок, невиходячи з дому. Це набагато зручніше, ніж бігати по роздрібних магазинах, намагаючись знайти потрібні товари. Особливо це стосується великих предметів, таких як побутова та комп'ютерна техніка.

Всі ці фактори відіграють життєво важливу роль у набутті популярності системи Інтернет-магазинів серед клієнтів. З іншого боку, є декілька питань, які потребують вирішення, як-от проблема повернення товару, запізнення доставки, шахрайство та можливі додаткові витрати на доставку.

Цифрова трансформація бізнесу, зокрема комерції і повсякденного життя в цілому, є гарячою темою вже багато років. Поступово в Інтернеті знаходять своє

місце навіть самі повсякденні і матеріальні заняття. Споживачам подобається робити покупки в невимушеній обстановці, з телефоном в руці або перед ноутбуком.

Дослідження показують, що на сегмент роздрібно́ї торгівлі електронікою через Інтернет припадає основна частка ринку і він буде продовжувати домінувати наступні роки. Деякі з основних факторів, які відповідають за ріст даного сегмента – це зростання онлайн-продажів електроніки, такої як смартфони, планшети і ноутбуки через онлайн-магазини. Крім того, зростаючі зусилля онлайн-продавців скоротити різницю між офлайн-цінами і онлайн-цінами на електроніку, сприятимуть подальшому зростанню продаж техніки через Інтернет в найближчі роки [4].

Глобальний ринок роздрібно́ї торгівлі електронікою є висококонкурентним завдяки присутності великої кількості регіональних і міжнародних продавців. Торгівля електронікою онлайн відкриває величезні можливості росту для підприємців по всьому світу.

Інтернет створює нові способи продажів, реклами, зв'язку з клієнтами, постачальниками і іншими користувачами.

Отже, розробка для компанії з продажу комп'ютерної техніки власного Інтернет-магазину виправдана, так як:

- існуючі та потенційні клієнти зможуть швидко отримати докладну інформацію про товари та послуги;
- компанія зможе розширити взаємозв'язок з клієнтами;
- процес роботи з клієнтами можна частково автоматизувати.

Тому створення інформаційної технології з продажу комп'ютерної техніки має першочергове значення для розвитку бізнесу в цій галузі.

1.2 Аналіз існуючих програмних засобів з продажу комп'ютерної техніки

Наразі існують Інтернет-магазини комп'ютерної техніки різних масштабів – гіганти Amazon, Ebay, Walmart, Rozetka або невеликі магазинчики, які ще не встигли стати досить популярними.

Через велику кількість торгових майданчиків з продажу комп'ютерної техніки, створити магазин, який би міг в короткі терміни привернути увагу клієнтів доволі важко. Можна взяти все у свої руки і спробувати для розробки онлайн-конструктор, а можна найняти професіоналів, які зроблять якісний продукт. Так чи інакше, успіх сайту залежить від ряду факторів, тому ми проаналізуємо аналоги і виділимо деталі, які варто буде врахувати при розробці продукту.

Перший аналог – Інтернет-магазин Rozetka (рис. 1.1). Тут продається величезна кількість товарів, тому було важливим зробити таку структуру сайту, яка була б зрозуміла кожному.

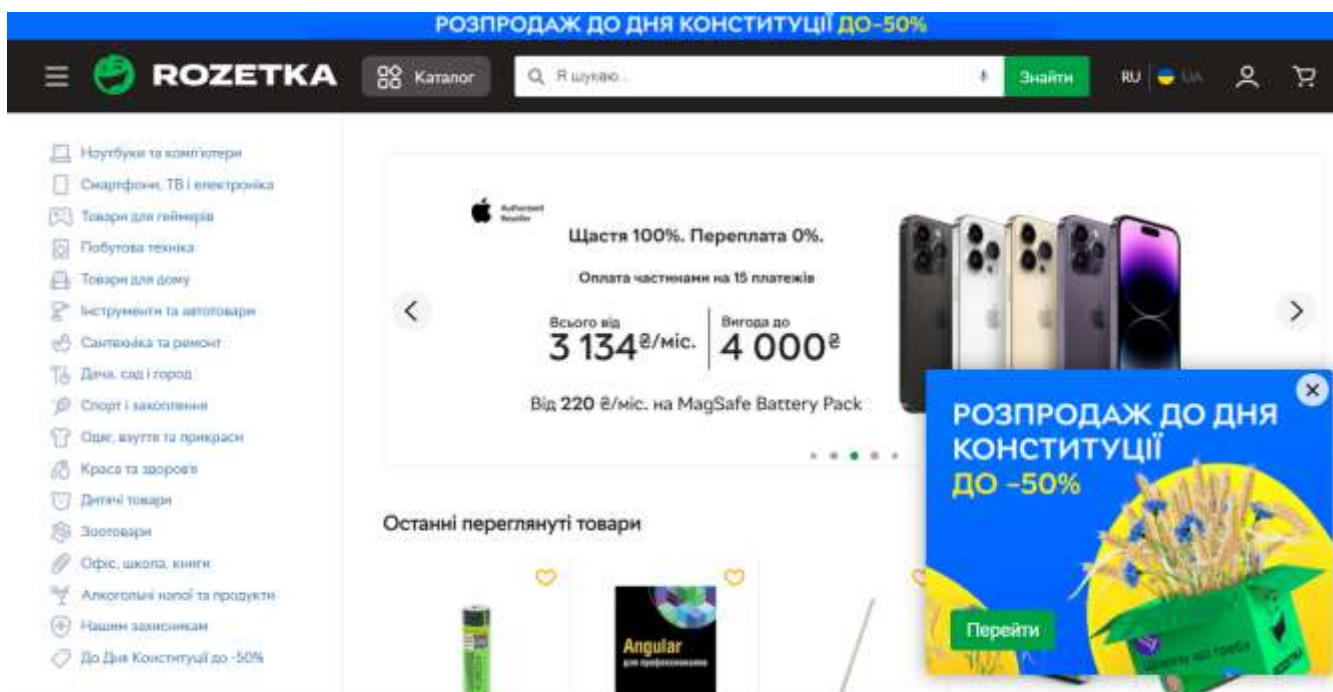


Рисунок 1.1 – Інтернет-магазин «Rozetka»

Також приємно дивує функціонал сайту. При виборі категорії товару, зліва показується фільтр з безліччю параметрами, за допомогою яких можна вибрати потрібний товар. Дизайн доволі простий, що є лише плюсом, немає нічого відволікаючого уваги. Швидкість завантаження бажає бути кращою, можна припустити, що причиною цьому є велика кількість даних [5].

Наступний аналог – Telemart.ua (рис. 1.2). Тут також наявні зручні навігація і функції пошуку по фільтрам. Дизайн сайту виконаний у фірмових кольорах. Завдяки білому фону поєднаному з фотографіями товарів створюється єдина концепція.

Головною фішкою порталу є так званий «Конфігуратор ПК». Даний інструмент дозволяє підібрати повністю сумісні компоненти, а також в реальному часі інформує про помилки та проблеми у збірці. Контент завантажується доволі швидко, всі елементи відображаються коректно [6].

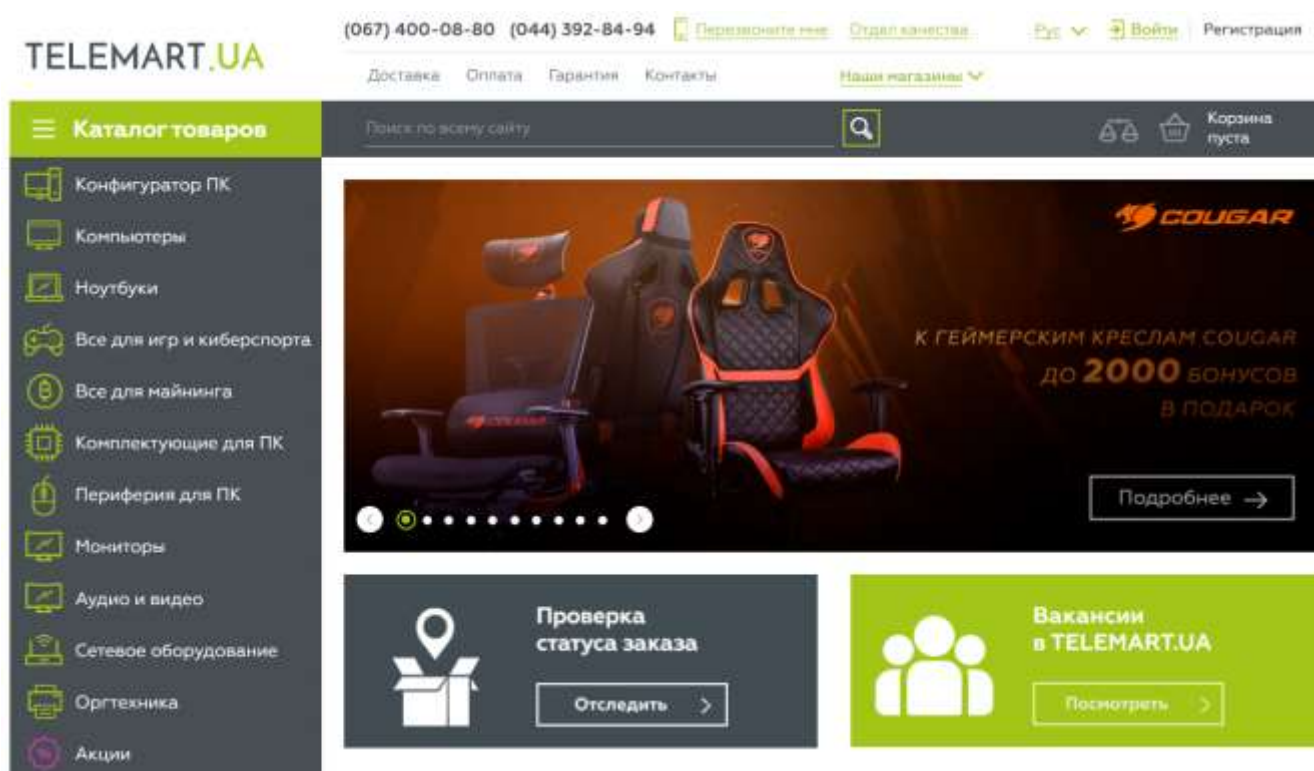


Рисунок 1.2 – Інтернет-магазин «Telemart.ua»

Останній аналог, який буде розглянуто – «ЗОНА51» (рис. 1.3). Проаналізувавши даний ресурс, можна зробити деякі висновки. Не для усіх користувачів буде візуально зручною чорна кольорова гама, через це головні елементи не відразу потрапляють у поле зору, теги «alt» вказані не для усіх важливих картинок. Картка товару є простою та зручною, є все необхідне з точки зору зручності. Товар завжди на виду і можна швидко перейти до інформації про нього, це дуже зручно, особливо, якщо сторінка довга [7].

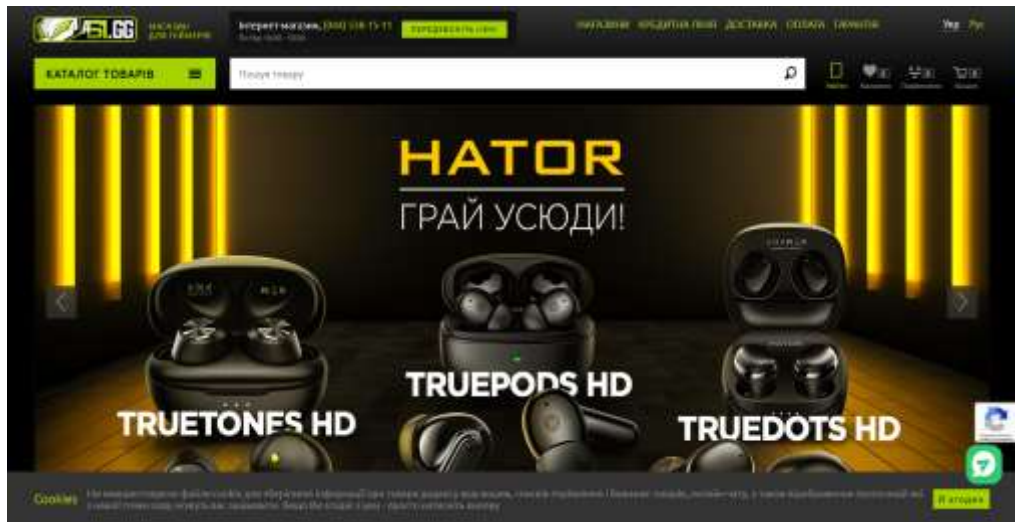


Рисунок 1.3 – Інтернет-магазин «ЗОНА51»

Для кращого результату порівняння було створено табл. 1.1, у якій зазначено деякі критерії оцінки сайтів.

Таблиця 1.1 – Порівняльна характеристика веб-сайтів аналогів

Критерії	Rozetka	Telemart.ua	ЗОНА51
Повільне завантаження сайту	+	-	-
Кольори, шрифти, графіка витримані у єдиному стилі	+	+	+
Кольори сайту “ріжуть” очі	-	-	+
Текст легко читається	+	+	-
До всіх іконок використані спливаючі підказки	+	-	-
Проста навігація	-	+	+
Кросбраузерність	+	+	+
Елементи притиснуті один до одного	-	-	+
Наявність посилань на неіснуючі сторінки	-	+	-

Зробивши аналіз даних, які були подані в таблиці 1.1, можна зробити висновки про аналоги і у майбутньому при розробці власної інформаційної технології врахувати їх переваги і уникнути повторення недоліків.

В епоху активного розвитку інформаційних технологій, створення інтернет-магазину вже не є такою проблемою як це було, наприклад, десять років тому. Але є багато нюансів, які варто передбачити перед тим як запускати власний ресурс. Існує декілька способів розробки, кожен з них має як недоліки так і переваги. Одним із перших кроків якраз являється рішення про те, який з них обрати.

Перший метод полягає в тому, що для створення ресурсу використовується конструктор.

Конструктор веб-сайтів – це програмне забезпечення, за допомогою якого можна створювати і в подальшому публікувати на хостингу ресурс, при цьому не використовуючи ручне кодування. Найпопулярніші конструктори: Wix, Tilda, SiteBuilder, SquareSpace, Weebly [8].

Головна перевага в тому, що цей спосіб ідеально підходить для новачків. Шаблони мають безліч функцій, які упрощують налаштування сайту. Також в основному всі конструктори безкоштовні, але за окремі функції доведеться платити.

Недоліки. Хоча даний спосіб і простий, але він має обмеження. Головне з них – Ви не маєте доступу до коду, що не дає можливості робити більш поглиблені коректування.

Ще один спосіб створення сайту – це використання CMS (система управління контентом).

CMS – це тип програмного забезпечення, яке дозволяє створювати і керувати цифровим контентом. Від конструкторів вони відрізняються своєю універсальністю. WordPress – найпопулярніша CMS, більше половини усіх сайтів в мережі Інтернет базуються на даній системі.

Переваги. Якщо при виборі CMS Ви віддасте перевагу WordPress, то матимете доступ до тисячі шаблонів з темами, зможете встановити плагіни, які суттєво розширяють функціонал сайту [9].

Тим не менш, CMS можуть не задовольнити усі потреби великих компаній, які використовують складні системні розрахунки (банки), для них потрібно вручну писати програмні модулі. Але за наявності достатнього грошового бюджету і

спеціальних навичок, WordPress, все-таки, може вирішити дану проблему, адже це найуніверсальніша CMS.

Останній варіант – створення сайту “з нуля”. Використовуючи поєднання з HTML, CSS і JavaScript, Ви створите динамічну частину сайту, інакше кажучи – front-end. Це те, з чим взаємодіє користувач.

JavaScript – найпопулярніша мова для написання клієнтської частини сайту. Це відносно молодий проект, який постійно оновлюється і вдосконалюється. JavaScript має велике значення у розвитку технологій для створення веб-ресурсів. В останній версії мови реалізовано безліч потужних функцій: регулярні вирази, форматування для числового виводу, інструменти управління, контролю виняткових ситуацій, жорсткий пошук помилок [10].

HTML – мова розмітки, яка є базовою частиною при створенні сайтів. Це так званий “кістяк”. Вона призначена для формування сторінок ресурсу, які в свою чергу призначені для перегляду в Інтернеті за допомогою браузерів.

Основні можливості HTML – це реалізація мережевих веб-сторінок з текстом, таблицями, фотографіями і т.д; отримання інформації з мережі через посилання гіпертексту; створення форм для відсилання різних запитів: пошук інформації, бронювання квитків, замовлення товарів і т.п.; додавання на сторінки відео, аудіо та інші програмні додатки.

CSS (Cascading Style Sheets) є формальною мовою, спрямованою на опис зовнішнього вигляду документів, які створюються за допомогою мов розмітки, таких як HTML. Розробники веб-сторінок використовують CSS для точного визначення елементів дизайну, таких як кольори, шрифти, розташування блоків та інші аспекти представлення вмісту на веб-сторінках.

Основна мета створення CSS полягала в розділенні опису логічної структури веб-сторінки, яке зазвичай виконується за допомогою HTML або інших мов розмітки, від опису зовнішнього вигляду цієї сторінки. Це дозволяє розробникам ефективно управляти стилями і макетом веб-сайтів, не залежно від їхньої структури. CSS працює за принципом "каскаду", що означає, що стилі можуть успадковуватися та

перевизначатися, що забезпечує більш гнучке та систематичне управління виглядом вмісту.

Ця технологія стала ключовою для розробки сучасних веб-сторінок, де важлива не лише інформація, а й естетика та користувацький досвід. Розширення можливостей CSS дозволяє реалізувати різноманітні дизайнерські концепції та забезпечує велику свободу у виразі творчості при створенні веб-інтерфейсів.

Поділ між логічною структурою веб-сторінки, який визначається мовами розмітки, і зовнішнім виглядом, який формується за допомогою CSS, привносить низку важливих переваг у розробці веб-документів. Цей підхід сприяє покращенню доступності і розширює межі гнучкості та контролю над відображенням інформації.

Збільшення доступності стає можливим завдяки можливості створення альтернативних стилів або адаптивних макетів, які можуть краще відповідати потребам користувачів з різних обмежень або пристроїв. Це включає в себе можливість налаштовувати розміри шрифтів, кольорову палітру та інші параметри зовнішнього вигляду для полегшення сприйняття інформації. Гнучкість, надана CSS, дозволяє розробникам ефективно контролювати стиль і вигляд веб-сторінок. Вони можуть швидко внести зміни в оформлення, не впливаючи на саму логічну структуру документа. Це особливо корисно при великих проектах або при необхідності швидкої модифікації дизайну. Зменшення складності і повторюваності в структурному вмісті стає можливим завдяки тому, що стилі визначаються централізовано і можуть бути застосовані до різних елементів без необхідності повторювати однакові правила для кожного елемента окремо. CSS також надає можливість представляти один і той же документ у різних стилях чи форматах виведення. Це важливо для забезпечення оптимального відображення на різних пристроях та в різних умовах. Наприклад, один документ може мати стилі для екранного відображення, друку або читання голосом, забезпечуючи кращу адаптабельність та зручність для різних користувацьких сценаріїв [11].

А за допомогою баз даних MySQL, мов програмування PHP або Python створюється back-end – серверна частина.

Переваги. Даний спосіб ідеальний для створення статичних сайтів. Це дозволить Вам, якщо Ви не плануєте в майбутньому змінювати контент, не витратитися на технічне обслуговування.

Недоліки. Якщо Ви не веб-програміст, дана задача може стати непідйомною. Тоді доведеться найняти веб-розробника, а це, зазвичай, дорога послуга. Також Ви не зможете хоч якось управляти сайтом, доведеться постійно звертатись до програмістів. Даний спосіб не найкращий варіант для динамічних сайтів, де постійно необхідно змінювати наповнення ресурсу.

1.3 Постановка задачі

Для розробки інформаційної технології з продажу комп'ютерної техніки в рамках виконання магістерської кваліфікаційної роботи необхідно виконати наступні задачі:

- здійснити аналіз предметної області продажу комп'ютерної техніки;
- проаналізувати засоби для реалізації інформаційної технології з продажу комп'ютерної техніки;
- розробити структуру та алгоритм функціонування інформаційної технології з продажу комп'ютерної техніки;

Використання сучасних ІТ-технологій дасть можливість істотно покращити якість роботи програмного забезпечення для продажу комп'ютерної техніки.

1.4 Висновок до розділу 1

У даному розділі було розглянуто предметну область та актуальність проблеми створення інформаційної технології з продажу комп'ютерної техніки. Електронна комерція активно розвивається і заповнює усі сфери діяльності. Оф-лайн продажі поступово переходять в Інтернет, а це в свою чергу викликає необхідність створення інформаційної технології з продажу комп'ютерної техніки у вигляді інтернет-магазину.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ

2.1 Аналіз інформаційного забезпечення сучасних веб-ресурсів

Ефективне функціонування сайту визначається не тільки комплексом технічних засобів, логічною структурою, але й в першу чергу раціональною організацією інформаційного забезпечення.

В Інтернеті, як ні в якому іншому засобі мас-медіа, важливо максимально швидко заволодіти увагою користувачів, оскільки є інші сайти подібної тематики або сайти конкурентів. При переході на головну сторінку, клієнту відразу має бути зрозуміло, що являє собою веб-сайт і чим він буде йому цікавий або корисний.

Інформацію доцільно ділити на невеликі порції. Мало хто з користувачів читає з екрану дуже довгі тексти. Дуже важливим є обсяг сторінок.

Також не треба забувати періодично оновлювати контент на сайті. Особливо важливо це для Інтернет-магазинів, які прагнуть сформувати у себе постійну аудиторію. Потрібно зрозуміти, що треба публікувати новини, стару інформацію коригувати і підтримувати актуальною. Чудовим рішенням є створення власного блогу. Тому на етапі розробки потрібно вирішити, хто саме буде відповідати за це на ресурсі і як часто це буде відбуватися. На невеликих ресурсах коригування можна проводити навіть в html-кодi заданих сторінок. Дану процедуру може робити і сам власник.

2.2 Розробка алгоритму роботи інформаційної технології з продажу комп'ютерної техніки

При переході на веб-ресурс, користувача зустрічає «Головна» сторінка, з якої можна перейти до повного каталогу продукції або вибрати конкретну категорію.

Вибравши конкретну категорію, клієнту відобразиться список товарів саме з обраної категорії. Щоб користувач не заплутався, у верхній частині сторінки відображається його поточне місцезнаходження – дерево сторінок.

Переглядаючи список товарів і натиснувши на назву товару, відвідувач перейде на сторінку з повним описом даного продукту. На цій сторінці клієнт може не тільки додати товар в корзину, а й побачити його опис і характеристики.

Додавання товару в корзину є досить простим, достатньо лише натиснути на кнопку «В кошик». Даний елемент інтерфейсу присутній біля кожного товару в загальному списку (при перегляді всього каталогу або якоїсь категорії) або на сторінці з докладною інформацією.

Після натискання кнопки, користувач потрапляє на сторінку самого кошика. На цій сторінці відображається список з обраних для замовлення товарів, який можна корегувати.

Далі клієнт має можливість перейти на сторінку замовлення або продовжити переглядати каталог.

Для оформлення замовлення клієнту потрібно надати необхідну контактну інформацію (ім'я, прізвище, адресу доставки і т.д.). При натисканні на кнопку «Оформити замовлення» система зберігає замовлення в базі даних.

Для реалізації даного функціоналу були розроблені різноманітні модулі:

- модуль бази даних;
- модуль клієнтського кабінета;
- модуль адміністратора;
- модуль авторизації та реєстрації;
- модуль кошика товарів;
- модуль візуалізації контенту;

Основний алгоритм замовлення товару, що об'єднує розроблені програмні модулі та описує їх взаємодію між собою наведений на рисунку 2.1.

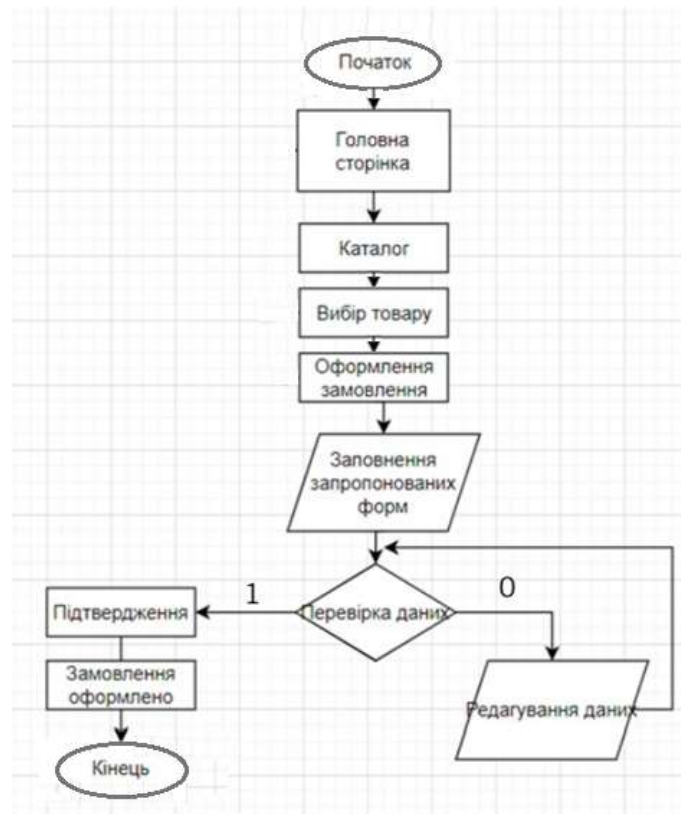


Рисунок 2.1 – Алгоритм функціонування інформаційної технології з продажу комп'ютерної техніки

База даних інформаційної технології з продажу комп'ютерної техніки складається з таблиць «Замовлення», «Клієнт», «Категорія» і «Товар» (рис. 2.2).

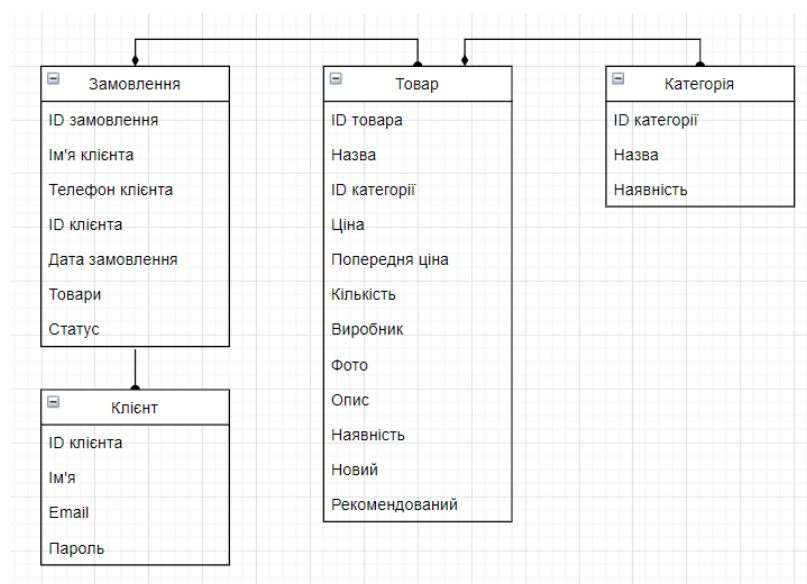


Рисунок 2.2 – Узагальнена схема бази даних інформаційної технології з продажу комп'ютерної техніки

2.3 Розробка структури інформаційної технології з продажу комп'ютерної техніки

Також було проаналізовано специфіку предметної області і спроектовано схему, що ілюструє роботу компонентів системи. Структура інформаційної технології з продажу комп'ютерної техніки представлена на рисунку 2.3.

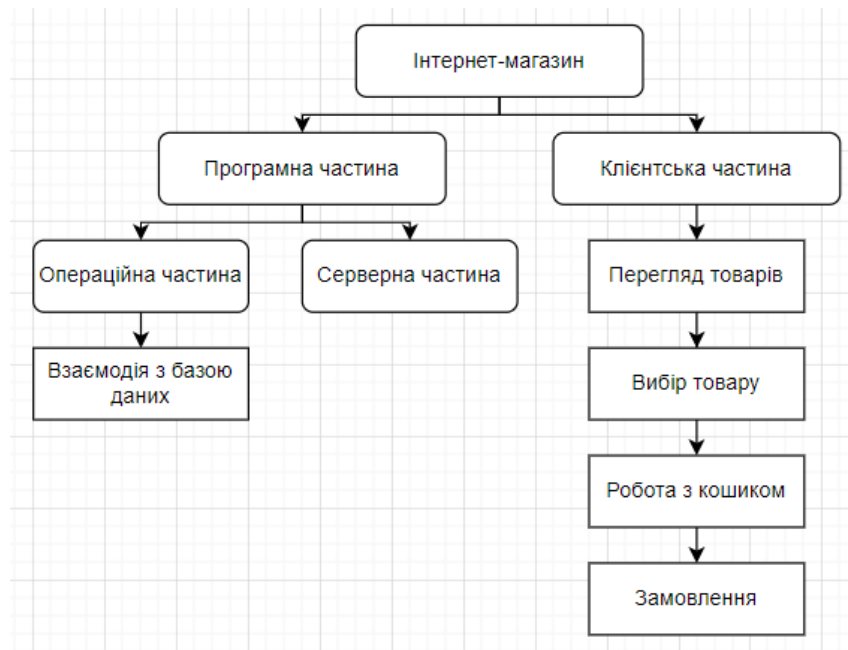


Рисунок 2.3 – Структура інформаційної технології з продажу комп'ютерної техніки

Архітектура інформаційної технології з продажу комп'ютерної техніки у вигляді інтернет-магазину розглядається як важлива взаємодія між двома ключовими компонентами: операційною та серверною частинами. Це взаємодія є критичною для забезпечення ефективного функціонування інтернет-магазину.

Клієнтська частина включає в себе розробку інтерфейсу інтернет-магазину, де ключову роль відіграють скрипти на мовах програмування, таких як JavaScript та PHP. JavaScript використовується для динамічної взаємодії з користувачем на стороні клієнта, наприклад, для валідації даних, асинхронного завантаження контенту або створення інтерактивних елементів. PHP використовується для обробки даних на сервері та генерації динамічного вмісту, такого як сторінки з каталогом товарів або обробка замовлень.

Серверна частина включає в себе розміщення інтернет-магазину на сервері, який може бути наданий провайдером хостингу. Важливо, щоб обрані сервери підтримували технології, які використовуються в розробці інтернет-магазину. Це може включати в себе підтримку мов програмування (наприклад, PHP), баз даних, а також застосування безпекових протоколів для захисту конфіденційності та цілісності даних.

Спільна робота між клієнтською та серверною частинами дозволяє створити повноцінний та ефективний інтернет-магазин. Клієнтська частина відповідає за зручний та привабливий інтерфейс для користувачів, тоді як серверна частина забезпечує надійну і стабільну роботу всіх функціональних складових, включаючи обробку замовлень, управління товаром та взаємодію з базою даних.

Метод оптимізації роботи з контентом в інформаційній технології з продажу комп'ютерів полягає у виконанні такої послідовності дій:

1. Авторизація користувача як адміністратора системи.
2. Доступ до адмінпанелі для оперативного керування контентом сайту.
3. Реалізація Інтернет-магазину клієнт-серверної архітектури веб-ресурсу.
4. Авторизація клієнтів з правом доступу перегляду та вибору товарів і формуванням кошика замовлення.

Сьогодні при розробці веб-ресурсу, багато уваги приділяють такому питанню, як структура. Декілька років тому усі сайти між собою були дуже схожі, але процес розробки веб-ресурсів не стоїть на місці і сучасні правила кардинально змінилися.

Схема структури веб-ресурсу обов'язково повинна розроблятися уже на початкових етапах створення будь-якого проєкту, слід визначитися – яким чином можна структурувати інформацію, з якої складається сайт.

Існує декілька типів структури веб-ресурсу, кожен з яких має свої переваги. Найпоширенішими вважаються:

- Лінійна структура веб-ресурсу. Дана схема є найбільш простою і зрозумілою. У ній кожна сторінка посилається на попередню і на наступну.
- Лінійна з відгалуженнями чим схожа на згадану вище, але має деяку відмінність. Вона передбачає використання декількох проєктів на одному сайті.

Пріоритетність передається від головної сторінки до останньої (останніх може бути декілька штук).

- Блокова структура. При такому підході всі складові сторінки рівні між собою. При цьому доцільно застосовувати таку схему в тих випадках, коли кожна сторінка деяким чином залежна від іншої. Ця структура найкраще підходить для малих сайтів [12].

Для веб-ресурсу інтернет-сайту магазину було вирішено використовувати блокову структуру (рис. 2.4).

На сторінці «Головна» будуть розміщені банер, який вітає клієнта і пропонує відразу перейти до каталогу, анонси новин, товари з категорії «Топ продаж» та останні опубліковані у блозі статті. Також буде наявна секція зі спеціальними пропозиціями. Фактично – це ті ж новини, але з можливістю замовлення відповідного товару через перехід на його сторінку в каталозі.

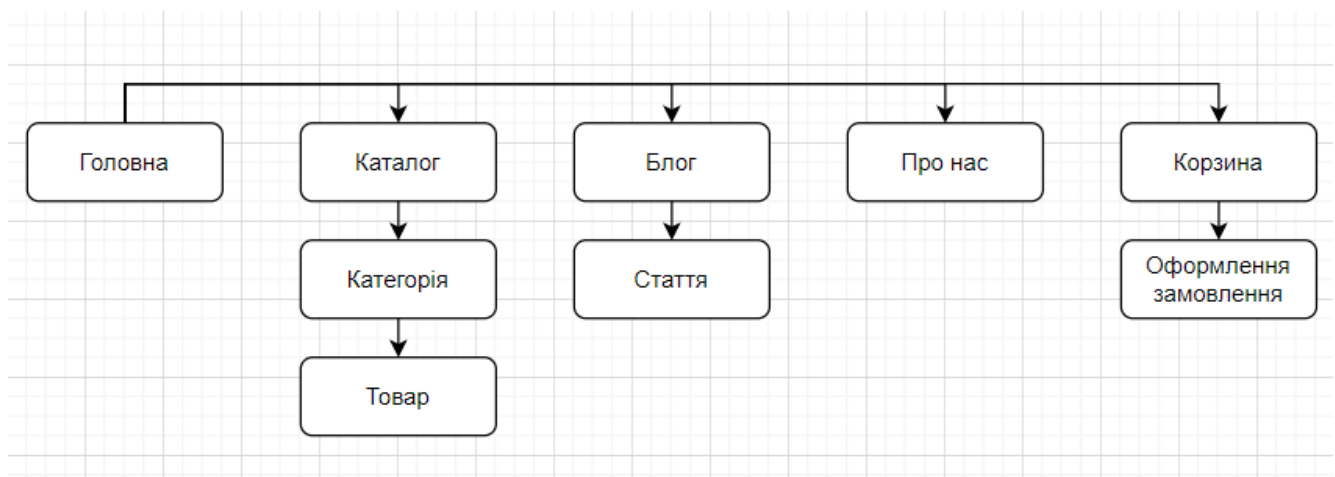


Рисунок 2.4 – Структура інтерфейсу інформаційної технології з продажу комп'ютерної техніки

Така інформація буде корисною для відвідувачів, якщо їх зустрічає щось нове, то це причина щоб повернутись сюди ще пізніше. До того ж така мобільність – показник динамічного розвитку компанії і сайту.

На сторінці «Каталог», де буде розміщена детальна інформація про всі товари, які пропонує магазин. Тут можна отримати розширену інформацію по конкретній позиції, таку як опис товару, його зовнішній вигляд (фото), його характеристики, ціна

та інше. З каталогу можна вибрати товар, помістивши його в кошик. Графіка буде максимально оптимізована для якнайшвидшого завантаження веб-сторінок сайту.

Сторінка «Блог» буде складатися з невеликих інформативних статей про комп'ютерну техніку, ознайомившись з якими, користувач може дізнатися багато чого нового для себе.

На сторінці «Про нас» надається опис компанії, переваги, контактна інформація магазину: телефон, графік роботи, карта, де показується розташування магазину.

В панелі навігації буде використовуватись іконка віртуального кошика, куди відвідувач може відкласти товар під час перегляду каталогу. При натисканні на дану іконку буде відображатися стан кошика з можливістю видалення з нього товарів. З цього ж блоку можна перейти на сторінку «Оформлення замовлення».

2.4 Розробка інтерфейсу та дизайну веб-ресурсу з продажу комп'ютерної техніки

При створенні будь-якого веб-ресурсу, обов'язково потрібно приділяти увагу розробці інтерфейсу.

Інтерфейс – це комплекс засобів, інструментів, призначених для взаємодії двох систем один з одним. В якості таких систем може виступати що завгодно, включаючи людей і штучний інтелект. Слово «інтерфейс» запозичене з англійської мови: interface означає «місце зіткнення».

В ролі таких інструментів взаємозв'язку можуть бути:

- текстові поля;
- кнопки;
- випадаючі списки;
- впливаючі вікна, підказки;
- перемикачі;
- елементи меню;
- і багато іншого.

Інтерфейс задає певні параметри, характеристики і функції взаємодії користувача з об'єктами інтерфейсу.

Проектування інтерфейсу потрібно не лише для візуального оформлення сайту. Це також можливість, щоб покращити користувацький досвід на веб-сайті. Інтерфейс потрібен для виконання наступних задач:

- Залучення відвідувачів. При переході на сайт, користувач відразу оцінює його зовнішній вигляд і зручність. Якщо він швидко знаходить те, що йому потрібно, він залишається. В іншому випадку він переходить до сайтів конкурентів.
- Швидке перетворення звичайного відвідувача у клієнта. Якщо сайт викликає довіру і позитивні емоції, то підвищується кількість продажів.
- Адаптація під усі пристрої. Деякі Інтернет-магазини не замислюються над тим, як їхнє веб-представництво буде виглядати на мобільних пристроях. Хоча зараз більшість покупок відбувається через смартфони.
- Підвищення довіри. Власний логотип, фірмові кольори та інші графічні елементи роблять ресурс більш впізнаваним, що підвищує довіру у користувачів [13].

Для реалізації інтерфейсу веб-ресурсу було обрано такі елементи, функція яких полягає у діалозі відвідувача із ресурсом – панель навігації, кнопки, гіперпосилання, шапка сайту, розмітка сторінок, банери, слайдери.

Розробка структури та дизайну інтерфейсу проводилась таким чином, щоб спонукати користувач до виконання певної дії: дзвінок, обрання товару, оформлення замовлення, перехід до блогу і т.д.

Взагалі інтерфейс складається з декількох сторінок, але всі дії починаються саме з головної сторінки. Вона складається із панелі навігації, банера, пропозицій популярних товарів, відображення статей з блогу, блоку email-розсилки, футера (рис. 2.5). При цьому хедер (логотип, навігація, кошик) та футер (email-розсилка, контактні дані та навігація) присутні на усіх сторінках сайту.



Рисунок 2.5 – Інтерфейс головної сторінки веб-ресурсу з продажу комп'ютерної техніки

Сьогодні у декого складається така думка, що дизайн веб-ресурсу це лише його зовнішній вигляд, його анімація та кольори. Але насправді це не так. Веб-дизайн обов'язково повинен бути «розумним» і адаптивним для користувача, це те з чим він стикається у першу чергу.

Від обсягу і сегментації інформації на сайті залежить вибір засобів навігації. Однією з найголовніших вимог до навігації є те, що вона має бути максимально зрозуміла для кожного, незважаючи на рівень обізнаності. Система навігації повинна виконувати функцію швидкого та інтуїтивного доступу до будь-якої сторінки веб-сайту. Якщо користувач без зайвих труднощів знайшов те, що шукав, то висока ймовірність, що він повернеться на сайт ще раз [14].

Панель навігації – список основних розділів сайту, розташовується на всіх сторінках сайту. У нашому прикладі посилання на сторінки всередині панелі навігації мають деревоподібну структуру (рис. 2.6).

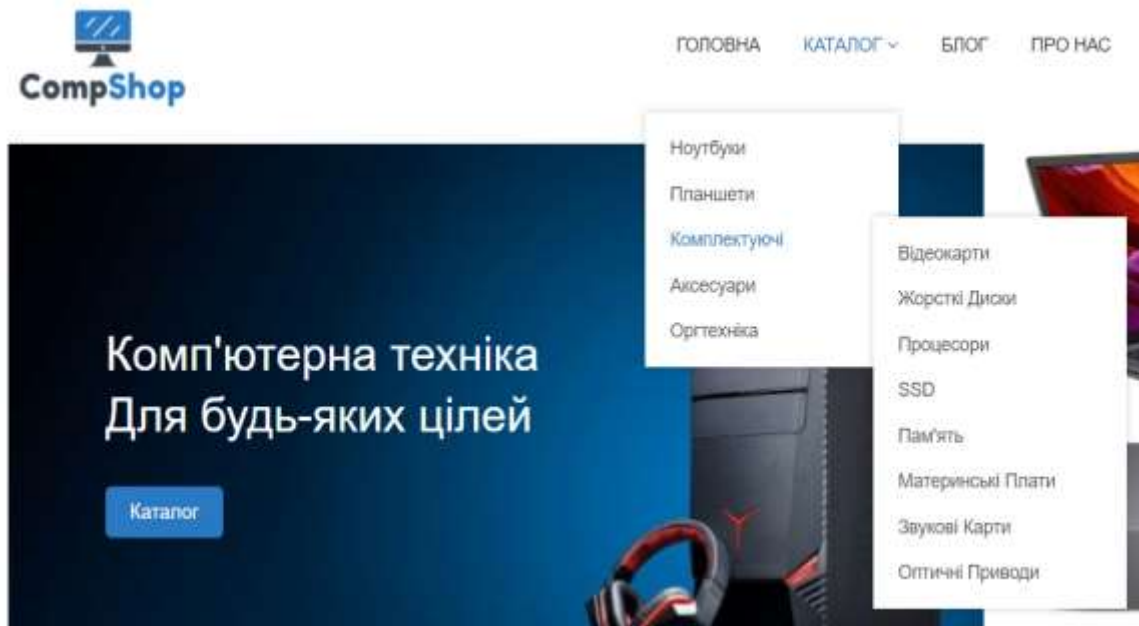


Рисунок 2.6 – Панель навігації

Посилання структуровані в залежності від ієрархії Інтернет-магазину (рис. 2.7).

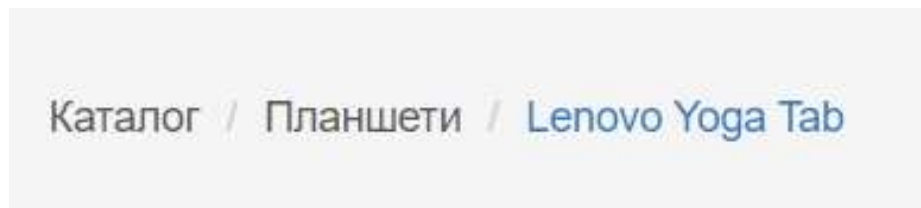


Рисунок 2.7 – Ієрархічна структура посилань на кожній сторінці товару

Основою навігації веб-ресурсу є гіпертекстові посилання, розташовані всередині сайту. Вони допомагають користувачу у будь-який момент з будь-якої сторінки перейти у потрібний розділ сайту. Навігація по сайту доступна з будь-якого розділу і є зручною для користувача. Меню і пункти каталогу з'являються таким чином, щоб у користувача не виникало питань, як перейти з однієї точки сайту в іншу [15].

Сайт обов'язково повинен бути адаптованим під мобільні пристрої. Згідно з дослідженнями, більшість користувачів заходять на веб-ресурси із смартфонів, у 2020 році їхня кількість була приблизно 52%. Головна цінність адаптивної версії в тому, що вона збільшує кількість потенційних клієнтів і підвищує продажі.

Розроблена адаптивна версія повністю відповідає наступним властивостям (рис. 2.8):

- зручність всіх елементів навігації, щоб відвідувачі мали можливість на них з легкістю натискати незалежно від розширення екрану;
- використання тільки вертикального прокручування сторінки;
- структурованість тексту для швидкого сприйняття – наявність заголовків, розділів і візуального контенту;
- читання тексту на будь-якому елементі без труднощів;
- відсутність реклами та спливаючих вікон.
-

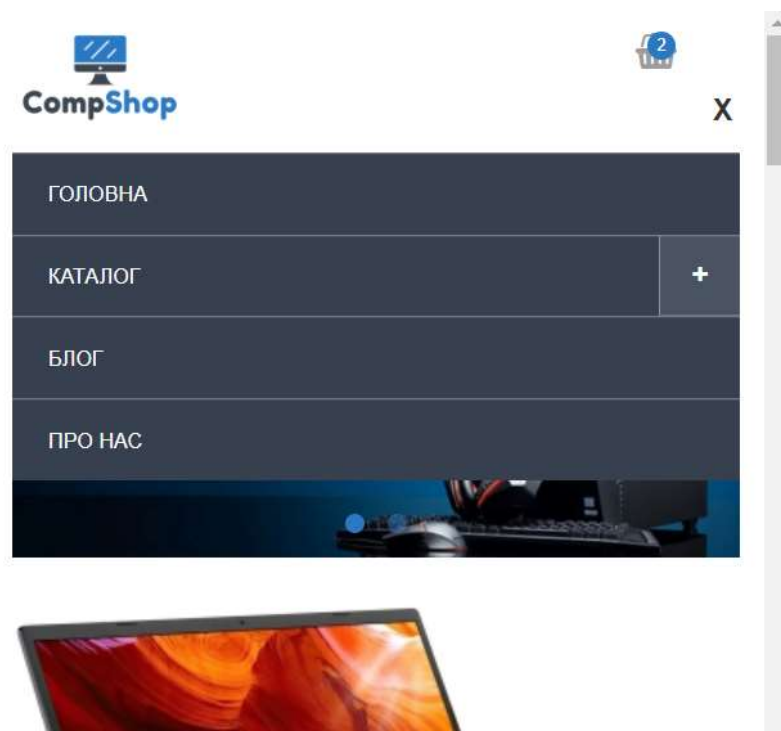


Рисунок 2.8 – Мобільна версія веб-ресурсу з продажу комп'ютерної техніки

Зовнішній вигляд сторінок веб-ресурсу управляється елементами дизайну, що забезпечуються каскадними таблицями стилів (Cascading Style Sheets, CSS). Каскадні таблиці застосовуються для виконання безлічі візуальних ефектів блоків на сторінках, включаючи положення, колір, розмір і т.д.

Нажаль, кожен браузер має свою специфіку, тому необхідно тестувати веб-сторінки з приєднаними таблицями стилів. При розробці Інтернет-магазину застосовувалася зовнішня таблиця стилів.

Під час оптимізації веб-ресурсу для підвищення його видимості в пошукових системах велике значення має вибір параметрів для оптимізації. Один з важливих параметрів - це номер позиції веб-ресурсу за конкретним запитом у пошукових системах. Однак, для досягнення цієї мети, необхідно вдосконалити ряд ключових елементів.

Заголовок сторінки (Title) є одним з найважливіших елементів, які впливають на позицію веб-ресурсу в пошукових системах. Він повинен точно відображати зміст сторінки та містити ключові слова, які важливі для пошукового запиту.

Мета-теги, такі як ключові слова (keywords) і опис (description), також важливі для оптимізації. Ключові слова мають відповідати тематиці сторінки, а опис повинен бути інформативним та привабливим для користувачів.

Зміст сторінки грає ключову роль в оптимізації для пошукових систем. Важливо оптимізувати частоту та вагу використовуваних слів, забезпечуючи їхнє природне та логічне використання. Розташування ключових слів також має важливе значення, зокрема в заголовках, підзаголовках та першому абзаці тексту.

Оптимізація цих параметрів сприяє покращенню рейтингу сайту у пошукових системах та підвищенню його видимості для цільової аудиторії. Зрозуміло та ефективно використані параметри оптимізації сприяють не тільки кращому розташуванню сайту в пошукових результатах, але і поліпшують користувацький досвід від взаємодії з вмістом сайту.

Оптимізація змісту сторінки, без оптимізації заголовка, часто стає мало ефективною. Головною метою оптимізації є підвищення рейтингу сторінки, адже слова в заголовках <title> мають велике значення для пошукових систем. Таким чином, можна зробити висновок: правильно зроблений заголовок сторінки – половина успіху. Для оптимізації сторінок, треба знати найкращі місця для позиціонування ключових слів. Тобто слів, під які ми оптимізуємо веб-сторінку.

Майже усю інформацію із зовнішнього світу ми сприймаємо за допомогою зору. Перше ж відразу враження про об'єкт ми складаємо, оцінюючи його десь на підсвідомості по зовнішньому вигляду, формі, розміру, кольору. Аналогічне відбувається у користувача при вигляді сайту. Дизайн ресурсу – перше, що кидається в очі людині, а його колір – перше, що оцінюється. Саме тому важливо знати, які кольори підібрати для вашого сайту, як їх комбінувати, щоб формувати у користувача правильні асоціації і враження – ті, які будуть спонукати до перегляду сайту і подальшим цільовим діям.

При розробці сайту, основним було обрано білий колір. Допоміжними кольорами було обрано відтінки синього та сірого.

Білий колір знаходиться на протилежному кінці спектра від чорного кольору, через це він чудово поєднується з будь-якими темними тонами і дозволяє за їх допомогою робити акценти. Саме тому було вирішено використати білий в якості основного фону. Даний колір створює ефект мінімалізму, але, разом з тим, робить сайт стильним. На сьогоднішній день подібне рішення – наймодніше і найпопулярніше.

Синій колір виглядає багато і розкішно. На сайті даний колір замальовує усі кнопки, а також деякі елементи навігації, при наведенні на них курсору (рис. 2.9). Кольори тексту, чорний та сірий, було обрано для того, щоб текст було добре видно на світлому фоні.



Монітор 27" Samsung Curved

В КОРЗИНУ



Планшет Lenovo Yoga Tab

9 999.00€ 44 999.00€

Рисунок 2.9 – Застосування синього відтінку для інтерфейсу веб-ресурсу

Важливим елементом будь-якого веб-ресурсу завжди є логотип. Користувач спочатку звертає увагу саме на нього. Тут він виконаний у тих же синьому та сірому тонах, нічого зайвого (рис. 2.10).



Рисунок 2.10 – Логотип веб-ресурсу з продажу комп’ютерної техніки

Детальна сторінка перегляду товару створена в єдиному стилі, що додає єдність та структурованість відображенню інформації. Контент відображається у вигляді компактних блоків, які зручно розташовані та визначені лініями, що чітко визначають межі між різними секціями сторінки.

На цій сторінці надаються головні характеристики товару, що дозволяє користувачеві швидко та зручно отримати ключову інформацію. Кожен блок може містити важливі параметри товару, такі як розміри, вага, технічні характеристики, функціональні особливості та інші важливі деталі.

Дизайн сторінки дозволяє легко визначити та порівняти основні характеристики товару, а використання ліній і блоків створює зручну структуру для організації інформації. Це особливо важливо для користувачів, які шукають конкретну інформацію та хочуть швидко зорієнтуватися у характеристиках товару без зайвого завантаження сторінки.

Крім того, використання єдиного стилю на сторінці допомагає створити єдність інтерфейсу, що робить взаємодію з веб-сайтом приємною та зрозумілою. Однорідність дизайну підсилює враження від користувацького досвіду та може позитивно впливати на сприйняття бренду чи товару користувачами (рис. 2.11).

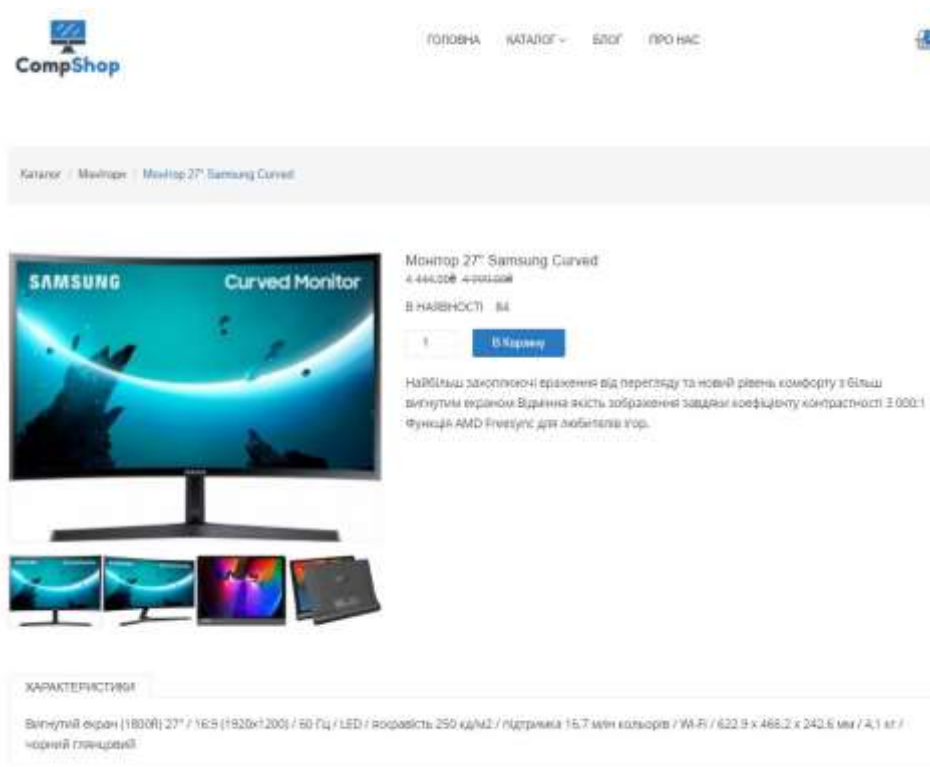


Рисунок 2.11 – Сторінка детального опису товару

2.5 Висновок до розділу 2

У даному розділі були проаналізовані види структур веб-ресурсу з продажу комп'ютерної техніки і було прийнято рішення використовувати блокову. Для даного ресурсу вона підходить найкраще, оскільки є найбільш оптимальною і зрозумілою для користувача.

Також було проведено аналіз інформаційного забезпечення веб-ресурсу. Вирішено, які програмні модулі будуть реалізовувати основний функціонал веб-ресурсу, з яких сторінок буде складатися ресурс і яка інформація буде на них знаходитись.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ

3.1 Обґрунтування вибору мови програмування

Як уже було розглянуто у другому розділі, методи розробки сайтів можна поділити на дві групи – автоматизоване і ручне. До першої групи відносяться спеціальні конструктори і системи керування контентом (CMS), а до другої використання мов розмітки та мов програмування. При розробці застосовувався саме другий спосіб.

Взагалі реалізація даного процесу розділяється на два етапи – фронтенд і бекенд. Фронтенд фахівці займаються клієнтської стороною – тобто тим, що побачить користувач. Бекенд – це програмно-апаратна частина сервісу, те, що працює на сервері. Фронтенд-розробники зазвичай не обходяться без HTML, CSS і JavaScript. Для бекенду обов'язковим є використання PHP, Python, Ruby.

Перед вибором певної мови програмування потрібно врахувати такі речі як – програмна частина, наявність досвіду написання коду, а також БД для серверної частини. Взявши до уваги ці умови, можна визначити для себе, яку мову найкраще використовувати.

PHP є потужною мовою програмування, яка володіє кількома важливими перевагами, особливо коли використовується для розробки веб-сайтів. Одна з ключових переваг цієї мови - відсутність конфліктів з HTML версткою. PHP може інтегруватися безперешкодно в HTML-код, дозволяючи одночасно використовувати HTML-теги для створення зовнішнього вигляду сторінки та вбудовувати PHP-код для реалізації функціональності сторінки.

Легкість освоєння PHP робить її доступною для широкого кола розробників на будь-якому етапі вивчення програмування. Її синтаксис лаконічний та зрозумілий, що полегшує швидке розуміння та використання для створення динамічних веб-сторінок.

PHP також славиться розвинутою підтримкою роботи з даними, що дозволяє легко обробляти інформацію, здійснювати взаємодію з базами даних та виконувати різноманітні операції з обробкою інформації на сервері.

Ще однією вагомою перевагою є широкий спектр апаратних платформ і операційних систем, на яких може працювати PHP. Це робить його універсальним і зручним використовувати в різних середовищах.

Не останнє місце у популярності PHP займає те, що ця мова призначена спеціально для роботи на стороні сервера, що робить її ідеальним вибором для створення динамічних веб-сайтів та взаємодії з базами даних у веб-розробці [16].

Python. Цю мову більшість фахівців вважають ідеальною в DataScience. Одним з головних плюсів Python вважається його простота. При наявності бажання, з його особливостями і тонкощами програмування зможе розібратися кожен бажаючий. До того ж Python сприяє економії часу програміста, так як пропонує велике число спеціальних бібліотек з уже готовими програмними конструкціями.

Ruby. Основне призначення даної мови – формувати і програмувати сайти, а також мобільні додатки. Навколо мови Ruby склалася думка про повільність і неможливість масштабувати великі проекти. З самого початку свого існування на рахунок продуктивності мова Ruby і правда поступалась PHP і Python. Однак численні оновлення мови в корені виправили ситуацію, майбутні апгрейди повинні принести і інші зміни – можливість роботи з паралельними потоками.

Для кращого результату порівняння було створено таблиці 3.1, у якій зазначено деякі критерії оцінки мов.

Таблиця 3.1 – Порівняння сучасних технологій розробки веб-ресурсу

	PHP	Python	Ruby
Багатоплатформеність	+	+	-
Простота вивчення та використання	+	+/-	+
Продуктивність і швидкість виконання коду	+	-	+/-
Наявність доступних програмних бібліотек	+	+	-
Розвинена підтримка даних	+	-	+/-

Проведений аналіз дозволяє зробити висновок, що найбільш доцільним і виправданим буде використання PHP-технологій. Головними перевагами цієї мови програмування є простота вивчення, легкість у застосуванні, ефективність і головне – продуктивність та гнучкість.

Отже, для розробки веб-сайту було обрано технології HTML5, CSS3, JavaScript та PHP. Наступним етапом є вибір середовища розробки, яке підтримує дані технології і дозволяє з максимальною ефективністю використовувати їх для веб-програмування.

3.2 Обґрунтування вибору середовища розробки

Для розробки за допомогою мов програмування PHP та JavaScript є безліч середовищ розробки або текстових редакторів. Головна різниця між ними в тому, що в IDE Ви зможете скомпілювати ваш проект, відлагодити його. Текстові редактори, як правило, цього не вміють робити, вони мають більш широкий підхід – мають можливість редагувати всі типи файлів і не спеціалізуються на конкретній мові програмування. Проте деякі з них дозволяють це виправити встановленням спеціальних розширень. Також важливим є пункт швидкості роботи. IDE зазвичай важкі, складні та повільні, редактори в цьому плані їх випереджають.

У цьому розділі розглянуто найпопулярніші на сьогоднішній день середовища розробки для веб-розробки – Visual Studio Code, Sublime Text 3, PhpStorm.

Visual Studio Code - це високоефективний і універсальний інструмент для розробки, який призначений для створення, редагування та зневадження різноманітних веб-застосунків і програм для хмарних систем. Однією з його ключових переваг є те, що він є безкоштовним та підтримується на різних операційних системах, таких як Windows, Linux і OS X, що робить його доступним для широкого кола розробників.

Унікальність Visual Studio Code полягає в його багатofункціональності та зручності використання. В редакторі вбудовані такі інструменти, як зневаджувач, підсвітка синтаксису, засоби для роботи з Git, а також різні засоби рефакторингу, які

полегшують розробку і забезпечують ефективну роботу з кодом. Навігація по коду, автодоповнення типових конструкцій і контекстні підказки роблять процес розробки більш продуктивним та зручним.

До того ж, Visual Studio Code має вбудований магазин плагінів, через який можна легко завантажувати та встановлювати різноманітні розширення. Це дозволяє розробникам додавати нові функції та адаптувати редактор до своїх потреб, роблячи його ще більш потужним та гнучким інструментом для роботи.

Загалом, Visual Studio Code є важливим інструментом у сфері веб-розробки, який поєднує у собі високу продуктивність і зручність використання, що робить його популярним серед програмістів усього світу.

Sublime Text 3 – це високоефективний, багатоплатформовий текстовий редактор, який відзначається своєю універсальністю та зручністю використання. Розроблений з врахуванням потреб програмістів та веб-розробників, Sublime Text став одним із важливих інструментів для редагування вихідних текстів програм.

Однією з ключових переваг Sublime Text є його розширена підтримка різних мов програмування. Редактор підтримує велику кількість мов, що включає в себе C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, Ruby, SQL, TCL і XML. Це дозволяє розробникам працювати з проектами, написаними на різних мовах, зберігаючи при цьому консистентність та ефективність роботи.

Однією з головних переваг є також можливість підсвічування синтаксису, що полегшує сприйняття коду та виявлення помилок. Sublime Text дозволяє розробникам працювати з великими обсягами коду, забезпечуючи швидке та ефективне редагування, а також інтеграцію з іншими інструментами розробки.

Крім того, Sublime Text відомий своєю гнучкістю та можливістю налаштування різних параметрів відповідно до особистих вподобань користувача. Інтерфейс редактора є простим та зрозумілим, а продуктивність його роботи дозволяє ефективно вирішувати завдання в області розробки програмного забезпечення та веб-розробки.

PhpStorm – це високорівневе інтегроване середовище розробки (IDE), яке спеціалізується на розробці за допомогою мов програмування PHP, а також забезпечує підтримку HTML і JavaScript. Це інтелектуальне середовище надає розробникам широкий набір інструментів для підвищення продуктивності та якості коду. PhpStorm включає в себе аналіз коду на льоту, що дозволяє виявляти потенційні помилки та надає рекомендації для поліпшення кодової бази. Це сприяє покращенню надійності та стабільності програмного забезпечення. Автоматизовані засоби рефакторінгу дозволяють розробникам легко оптимізувати та переписувати свій код для покращення читабельності та ефективності. Однією з важливих функцій PhpStorm є повноцінний SQL-редактор, який надає можливість редагувати та оптимізувати результати SQL-запитів безпосередньо в середовищі розробки. Це дозволяє зручно взаємодіяти з базами даних та виконувати роботу з SQL-запитами без переходу до інших інструментів.

Розроблене на основі платформи IntelliJ IDEA, написаної на Java, PhpStorm володіє високим рівнем продуктивності та стабільності. Крім того, користувачі можуть розширити функціональність середовища розробки за допомогою різноманітних плагінів, розроблених для платформи IntelliJ, або створити свої власні розширення, що робить PhpStorm гнучким інструментом, який можна налаштовувати під конкретні потреби розробника.

Порівняння описаних програм наведено у таблиці 3.2.

Таблиця 3.2 – Порівняльний аналіз програм для веб-розробки

Характеристика	VS Code	Sublime Text 3	PhpStorm
Користування	безкоштовно	безкоштовно	підписка
Підтримка плагінів, які спрощують написання коду	+	+	+
Підсвітка синтаксису	+	+	+
Автодоповнення	+	+-	+
Аналіз коду в реальному часі	+	-	+
Наявність консолі	+	+-	+
Відлагоджувач коду	+	-	+

У кожного редактора, які було проаналізовано, є свої особливості та переваги. Враховуючи порівняння, наведене вище, вибір падає на Visual Studio Code.

Даний продукт з коробки володіє великою кількістю можливостей, інтуїтивно зрозумілий, легко налаштовується, при цьому ще й досить швидко працює. Продукт позиціонується як безкоштовне рішення, яке дозволяє обійтися без повного інтегрованого середовища розробки.

3.3 Робота системи керування базами даних для веб-ресурсу з продажу комп'ютерної техніки

Для роботи динамічного веб-ресурсу потрібно обов'язково використовувати бази даних. Завдяки їм можна зберігати і обробляти інформацію в структурованому вигляді. В основі всієї структури зберігання лежать три поняття: база даних, таблиця, запис. База даних означає об'єднання сукупності даних, що зберігаються для виконання однієї мети. Наприклад: ім'я товару, вартість, знижка, категорія і т.д. Якщо ми робимо сучасний сайт, то всі його дані будуть зберігатися всередині однієї бази даних.

Для керування наявними базами даних використовують СУБД – система управління базами даних. Створювати свої бази даних, таблиці та наповнювати їх даними можна прямо з цієї ж програми, але для виконання цих операцій спочатку доведеться познайомитися з ще однією мовою програмування – SQL.

SQL або Structured Query Language – мова програмування, призначена для управління даними в СУБД. Всі сучасні СУБД підтримують SQL.

За допомогою SQL можна не тільки додавати і зчитувати дані, але і:

- видаляти і оновлювати записи в таблицях;
- створювати і редагувати самі таблиці;
- виробляти операції над даними: вираховувати суму, отримувати найбільше чи найменше значення і так далі;
- налаштовувати роботу сервера СУБД [17].

Існує безліч різноманітних реляційних СУБД. Найвідоміша – це Microsoft Access, що входить до складу офісного пакету додатків Microsoft Office.

Немає ніяких перешкод для використання в якості СУБД MS Access, але для задач веб-програмування набагато краще підходить альтернативна програма – MySQL.

На відміну від MS Access, MySQL абсолютно безкоштовна, має набагато більшу продуктивність і більш безпечна, що робить її ідеальним кандидатом на роль бази даних в веб-розробці. Найголовніше те, що PHP підтримує роботу з базою даних MySQL. Переважна більшість сайтів і додатків на PHP використовують в якості СУБД саме MySQL. PhpStorm включає в себе спеціальні вбудовані функції, спрямовані на ефективну роботу з MySQL, що дозволяє розробникам зручно взаємодіяти з цією системою управління базами даних (СУБД). Ці функції спрощують виконання різноманітних завдань, пов'язаних із роботою з базою даних MySQL.

За допомогою вбудованих функцій, розробники можуть легко виконувати різноманітні SQL-запити безпосередньо в PhpStorm. Це включає в себе читання та записування даних, виконання складних запитів та обробку різноманітних операцій з базою даних.

Окрім цього, вбудовані функції дозволяють ефективно обробляти помилки, що виникають при виконанні SQL-запитів. Це полегшує виявлення та виправлення проблем у взаємодії з базою даних, що є важливим етапом при розробці програмного забезпечення.

Загалом, вбудовані функції для роботи з MySQL в PhpStorm роблять роботу з базою даних більш простою та ефективною для розробників, дозволяючи їм швидко та зручно взаємодіяти з даною СУБД прямо з інтегрованого середовища розробки.

Типовий процес роботи з системою управління базами даних (СУБД) в PHP-сценарії передбачає виконання кількох послідовних кроків для ефективного взаємодії з базою даних. Нижче описані ключові етапи цього процесу:

Встановлення підключення до бази даних: Розпочаток взаємодії з СУБД зазвичай включає встановлення з'єднання між PHP-сценарієм і базою даних. Це може

вимагати використання функції `mysqli_connect()` або PDO для встановлення з'єднання, підключення та передачі необхідних аутентифікаційних даних.

Виконання SQL-запитів: Після встановлення підключення можна виконувати SQL-запити для читання, запису, оновлення або видалення даних в базі. Для цього використовуються функції, такі як `mysqli_query()` або об'єкти `PDOStatement` в PDO. Запити можуть бути параметризованими для забезпечення безпеки та оптимізації.

Обробка результатів: Результати SQL-запитів обробляються для отримання необхідної інформації. Використовуються функції, такі як `mysqli_fetch_assoc()`, `mysqli_fetch_object()` або інші методи об'єкта `PDOStatement` для отримання даних з результатів запиту.

Обробка помилок: Обробка помилок грає важливу роль у взаємодії з базою даних. Використовуються конструкції типу `try...catch` (для PDO) або функції, такі як `mysqli_error()` (для MySQLi), щоб виявляти та обробляти помилки, які можуть виникнути під час виконання SQL-запитів.

Закриття з'єднання: Після завершення взаємодії з базою даних важливо закрити з'єднання, щоб звільнити ресурси. Використовуються функції, такі як `mysqli_close()` або метод `close()` об'єкта PDO, щоб закрити підключення.

Цей процес роботи з СУБД в PHP-сценарії є загальним шаблоном, але може варіюватися в залежності від конкретних вимог проекту та використовуваної системи управління базами даних.

Для управління базами даних буде використовуватися `phpMyAdmin`. `PhpMyAdmin` – веб-додаток написаний на мові PHP, який представляє собою веб-інтерфейс для адміністрування СУБД MySQL. `PhpMyAdmin` дозволяє здійснювати адміністрування сервера MySQL, запускати команди SQL і переглядати вміст таблиць безпосередньо через браузер. Це набагато спрощує роботу з базами даних, оскільки користувачу не потрібно вручну прописувати команди SQL [18].

На рисунку 3.1 зображена таблиця «`product`», у якій зберігаються дані товарів з каталогу. Вона була створена саме за допомогою даної системи.

+ Параметри									
		id	title	text	text2	primary_img	secondary_img	prev_price	price
		1	Монитор 27" Samsung Curved	Найбільш захищений екран від перепаду температури.	Вигнута екран (1000R) 27" / 16:9 (1920x1200) / 60.	img/products/1.jpg	img/products/2.jpg	4 999 000	4 444 000
		2	Планшет Lenovo Yoga Tab	ALL-IN-ONE: вражачою мультимедійний планшет з ролі.	Екран 10.1" (1920x1200) IPS, MultiTouch / Qualcomm.	img/products/3.jpg	img/products/4.jpg	11 999 000	9 999 000
		3	Графічний монітор Pardo13	Контактний інтерактивний дисплей Pardo13 c	Екран 13.3" (1920x1080) електромагнітний сенсор T.	img/products/5.jpg	img/products/6.jpg	17 999 000	16 999 000
		4	Kodak Alaris E102	Сканер Alaris E102 надає вам рішення для	Документ-сканер / A4 / 600x1200 / дисплей (тип mat.	img/products/7.jpg	img/products/8.jpg	15 699 000	14 320 000

Рисунок 3.1 – Таблиця «product» у phpMyAdmin

3.4 Програмна реалізація інтерфейсу інформаційної технології з продажу комп'ютерної техніки

Для збільшення функціональності ресурсу і для підвищення його ефективності було створено декілька програмних модулів.

Модуль бази даних представляється у вигляді класу Db. За його допомогою можна підключатися до бази ресурсу і брати звіди необхідну інформацію для реалізації інших модулів.

```

<?php
class Db
{
public static function getConnection()
{
$paramsPath = ROOT.'./config/db_parameters.php';
$params = include($paramsPath);
$db = new PDO ("mysql:host={$params['host']};dbname={$params['dbname']}",
$params['user'],      $params['password'],      [PDO::ATTR_ERRMODE
PDO::ERRMODE_EXCEPTION]);
$db->exec("set names utf8");
return $db;
} }

```

Модуль клієнтського кабінету призначений для редагування контактних даних, а також для перегляду оформлених замовлень. Для роботи даного контролера використовувався клас User.

```
<?php
include_once ROOT.'/models/Category.php';
include_once ROOT.'/models/User.php';

class AccountController
{
    public function actionIndex()
    {
        $guest = User::guest();
        $categories = array();
        $userId = User::checkLogination();
        $user = User::getUserById($userId);
        require_once(ROOT . '/views/cabinet/index.php');
        return true;
    }
    public function actionEdit()
    {
        $userId = User::checkLogination();
        $user = User::getUserById($userId);
        $name = $user['name'];
        $password = $user['password'];
        $result = false;
        if (isset($_POST['submit'])) {
            $name = $_POST['name'];
            $password = $_POST['password'];
            $errors = false;
            if (!User::checkName($name)) {
```

```

        $errors[] = "Ім'я має містити більше трьох символів";
    }
    if (!User::checkPassword($password)) {
        $errors[] = "Пароль має містити більше семи символів";
    }
    if ($errors == false) {
        $result = User::edit($userId, $name, $password);
    }
}
require_once(ROOT . '/views/account/edit.php');
return true;
}}

```

Модуль адміністратора призначений для керування контентом сайту. Надає можливість додавати, редагувати, видаляти категорії, товари та замовлення. Реалізується за допомогою контролерів AdminController, AdminCategoryController, AdminProductController, AdminOrderController. Нижче для прикладу наведено лістинг класу, який відповідає за управління товарами.

```
<?php
```

```

class AdminProductController extends AdminBase
{
    public function actionIndex()
    {
        self::checkAdmin();
        $productsList = Product::getProductsList();

        require_once(ROOT . '/views/admin_product/index.php');
        return true;
    }
}

```

```
public function actionAdd()
{
    self::checkAdmin();
    $categoriesList = Category::getCategoriesListAdmin();

    if (isset($_POST['submit'])) {
        $options['name'] = $_POST['name'];
        $options['category_id'] = $_POST['category_id'];
        $options['price'] = $_POST['price'];
        $options['prev_price'] = $_POST['prev_price'];
        $options['availability'] = $_POST['availability'];
        $options['brand'] = $_POST['brand'];
        $options['img1'] = $_POST['img1'];
        $options['img2'] = $_POST['img2'];
        $options['text1'] = $_POST['text1'];
        $options['text2'] = $_POST['text2'];
        $options['new'] = $_POST['new'];
        $options['recommended'] = $_POST['recommended'];
        $options['status'] = $_POST['status'];
        $options['color'] = $_POST['color'];

        $errors = false;
        if (!isset($options['name']) || empty($options['name'])) {
            $errors[] = 'Внесіть дані';
        }

        if ($errors == false) {
            $id = Product::createProduct($options);
            if ($id) {
                if (is_uploaded_file($_FILES["image"]["tmp_name"])) {
```

```

        move_uploaded_file($_FILES["img"]["tmp_name"],
$_SERVER['DOCUMENT_ROOT'] . "/upload/images/products/{$id}.jpg");
    }
};
header("Location: /admin/product");
}
}
require_once(ROOT . '/views/admin_product/create.php');
return true;
}

```

```
public function actionEdit($id)
```

```

{
    self::checkAdmin();
    $categoriesList = Category::getCategoriesListAdmin();

    $product = Product::getProductById($id);

    if (isset($_POST['submit'])) {
        $options['name'] = $_POST['name'];
        $options['category_id'] = $_POST['category_id'];
        $options['price'] = $_POST['price'];
        $options['prev_price'] = $_POST['prev_price'];
        $options['availability'] = $_POST['availability'];
        $options['brand'] = $_POST['brand'];
        $options['img1'] = $_POST['img1'];
        $options['img2'] = $_POST['img2'];
        $options['text1'] = $_POST['text1'];
        $options['text2'] = $_POST['text2'];
        $options['new'] = $_POST['new'];
    }
}

```

```

        $options['recommended'] = $_POST['recommended'];
    $options['status'] = $_POST['status'];
    $options['color'] = $_POST['color'];
        if (Product::updateProductById($id, $options)) {
            if (is_uploaded_file($_FILES["image"]["tmp_name"])) {

                move_uploaded_file($_FILES["image"]["tmp_name"],
$_SERVER['DOCUMENT_ROOT'] . "/upload/images/products/{$id}.jpg");
            }
        }
        header("Location: /admin/product");
    }
    require_once(ROOT . '/views/admin_product/update.php');
    return true;
}

public function actionDelete($id)
{
    self::checkAdmin();

    if (isset($_POST['submit'])) {
        Product::deleteProductById($id);

        header("Location: /admin/product");
    }

    require_once(ROOT . '/views/admin_product/delete.php');
    return true;
}
}

```


Модуль авторизації та реєстрації користувача реалізований за допомогою класів User та UserController. Використовується принцип сесії. Нижче поданий лістинг контролера UserController.

```
<?php
include_once ROOT.'/models/Category.php';
include_once ROOT.'/models/User.php';

class UserController
{
    public function actionRegister()
    {
        $guest = User::guest();
        $categories = array();
        $categories = Category::getCategories();

        $name = false;
        $email = false;
        $password = false;
        $result = false;

        if (isset($_POST['submit'])) {
            $name = $_POST['name'];
            $email = $_POST['email'];
            $password = $_POST['password'];
            $errors = false;

            if (!User::verifyName($name)) {
                $errors[] = "Ім'я має містити більше трьох символів";
            }
        }
    }
}
```

```
if (!User::verifyEmail($email)) {
    $errors[] = "Email введений невірно";
}
if (!User::verifyPassword($password)) {
    $errors[] = "Пароль має містити більше семи символів";
}
if (User::checkEmailPresence($email)) {
    $errors[] = "Користувач з таким email вже зареєстрований на сайті";
}
if ($errors == false) {
    $result = User::register($name, $email, $password);
} }
require_once(ROOT . '/views/user/register.php');
return true;
}
```

```
public function actionLogin()
{
    $guest = User::guest();
    $categories = array();
    $categories = Category::getCategories();
    $email = false;
    $password = false;

    if (isset($_POST['submit'])) {
        $email = $_POST['email'];
        $password = $_POST['password'];
        $errors = false;
        if (!User::verifyEmail($email)) {
            $errors[] = "Email введений невірно";
```

```

    }
    if (!User::verifyPassword($password)) {
        $errors[] = "Пароль має містити більше семи символів";
    }
    $user = User::searchUser($email, $password);
    if ($user == false) {
        $errors[] = "Такого користувача не існує";
    } else {
        User::authorization($user);
        header('Location: /account/');
    }
}
require_once(ROOT . '/views/user/login.php');
return true;
}

public function actionLogout()
{
    unset($_SESSION['user']);
    header('Location: /');
}
}

```

Модуль кошика товарів Cart реалізує функції додавання, видалення товару; підрахунок кількості товарів та їх загальну вартість.

```

<?php
class Cart
{
    public static function addProduct($id)

```

```

{
    $id = intval($id);
    $productsInCart = array();
    if (isset($_SESSION['products'])) {
        $productsInCart = $_SESSION['products'];
    }

    if (array_key_exists($id, $productsInCart)) {
        $productsInCart[$id] ++;
    } else {
        $productsInCart[$id] = 1;
    }

    $_SESSION['products'] = $productsInCart;

    return self::countItems();
}

public static function countItems()
{
    if (isset($_SESSION['products'])) {
        $count = 0;
        foreach ($_SESSION['products'] as $id => $quantity) {
            $count = $count + $quantity;
        }
        return $count;
    } else {
        return 0;
    }
}
}

```

```
public static function getProducts()
{
    if (isset($_SESSION['products'])) {
        return $_SESSION['products'];
    }
    return false;
}
```

```
public static function getTotalPrice($products)
{
    $productsInCart = self::getProducts();

    $total = 0;
    if ($productsInCart) {
        foreach ($products as $item) {
            $total += $item['price'] * $productsInCart[$item['id']];
        }
    }

    return $total;
}
```

```
public static function clear()
{
    if (isset($_SESSION['products'])) {
        unset($_SESSION['products']);
    }
}
```

```

public static function deleteProduct($id)
{
    $productsInCart = self::getProducts();

    unset($productsInCart[$id]);

    $_SESSION['products'] = $productsInCart;
}
}

```

Модуль візуалізації контенту відповідає за перенаправлення на сторінку, яку запитує користувач.

```

<?php
class Router
{
    private $routes;
    public function __construct()
    {
        $routesPath = ROOT.'/config/routes.php';
        $this->routes = include($routesPath);
    }

    private function getURI()
    {
        if (!empty($_SERVER['REQUEST_URI'])) {
            return trim($_SERVER['REQUEST_URI'], '/');
        }
    }

    public function run()
    {
        $uri = $this->getURI();
    }
}

```

```

foreach ($this->routes as $uriPattern => $path) {
    if (preg_match("~$uriPattern~", $uri)) {
        $internalRoute = preg_replace("~$uriPattern~", $path, $uri);
        $segments = explode('/', $internalRoute);
        $controllerName = array_shift($segments).'Controller';
        $controllerName = ucfirst($controllerName);

        $actionName = 'action'.ucfirst(array_shift($segments));
        $parameters = $segments;

        $controllerFile = ROOT.'/controllers/'.$controllerName.'.php';
        if (file_exists($controllerFile)) {
            include_once($controllerFile);
        }

        $controllerObject = new $controllerName;
        $result = call_user_func_array(array($controllerObject, $actionName),
        $parameters);

        if ($result != null) {
            break;
        }
    }
}

```

3.5 Тестування програмного забезпечення з продажу комп'ютерної техніки та аналіз результатів його роботи

Процес тестування програмного забезпечення – це чи не найбільш обов'язковий і важливий етап будь-якої розробки, він має величезну роль в процесі створення продукту.

Важливо зазначити, що веб-система повинна пройти повноцінний процес тестування від початку до кінця, перш ніж вона буде готовою до використання користувачами. Тестування включає в себе різні аспекти, які стежать за різними аспектами функціональності та ефективності системи.

Функціональне тестування. Цей вид тестування спрямований на перевірку того, чи виконує система свої функції згідно з визначеними вимогами. Включає тестування функцій, інтерфейсу користувача, обробки даних та інших ключових елементів системи.

Тестування взаємодії та користувацького досвіду. Оцінюється зручність використання системи, привабливість інтерфейсу, а також якість взаємодії користувача з веб-сайтом чи додатком. Зокрема, перевіряється логіка навігації, якість виводу інформації, а також загальний враження від використання.

Тестування продуктивності. Визначається швидкість отримання та відображення інформації на веб-сайті, а також його стійкість до навантаження. Може включати тестування швидкості завантаження сторінок, відгуків від сервера та інші параметри продуктивності.

Тестування безпеки. Важливий аспект, оскільки забезпечення безпеки веб-системи є пріоритетом. Тестування безпеки включає в себе аудит вразливостей, тестування на проникнення, перевірку наявності правильної автентифікації та інші аспекти забезпечення безпеки.

Тестування сумісності. Визначається, як система веде себе на різних платформах, веб-браузерах та пристроях. Важливо переконатися, що веб-система працює ефективно та однаково на різних пристроях із різними характеристиками.

Загальна мета тестування - виявлення проблем або дефектів, які можуть виникнути під час використання веб-сайту чи додатку, та їх виправлення перед впровадженням в експлуатацію. Такий підхід забезпечує надійність, безпеку та високу якість веб-системи.

Тестування є необхідною складовою в розробці веб-сайтів, оскільки воно спрямоване на виявлення та виправлення помилок та недоліків, які можуть виникнути

під час експлуатації. Однак це не лише про пошук і усунення багів, а й про забезпечення зручності та логічності для користувачів.

Основні правила тестування веб-сайтів включають кілька ключових аспектів.

Функціональність. Перевірка, наскільки веб-сайт відповідає функціональним вимогам та завданням, визначеним в технічному завданні. Тестування функцій, кнопок, форм, інтерактивних елементів і т. д.

Зручність використання. Оцінка, наскільки зручно та логічно взаємодіє користувач із веб-сайтом. Врахування логіки навігації, доступності інформації, простоти використання.

Аспекти візуального дизайну. Визначення, наскільки привабливим та естетичним є дизайн веб-сайту, як добре сприймається людським зором.

Тестування на різних пристроях і браузерах. Впевненість, що веб-сайт працює коректно на різних пристроях (комп'ютерах, планшетах, смартфонах) і у різних веб-браузерах. Дотримання технічних вимог і стандартів безпеки. Перевірка відповідності веб-сайту технічним вимогам, а також визначення рівня безпеки та виявлення можливих ризиків. Такий підхід дозволяє не лише виявити і усунути помилки, але і забезпечити високий рівень зручності та задоволення для кінцевих користувачів. В тестуванні веб-сайтів важливо враховувати потреби користувачів і впевнено впроваджувати зміни та вдосконалення, які покращують якість та ефективність веб-системи.

Перевіримо чи правильно працює «Кошик». Для цього додамо до нього два товари, а потім спробуємо один видалити (рис. 3.2, 3.3).



Рисунок 3.2 – Список товарів до редагування

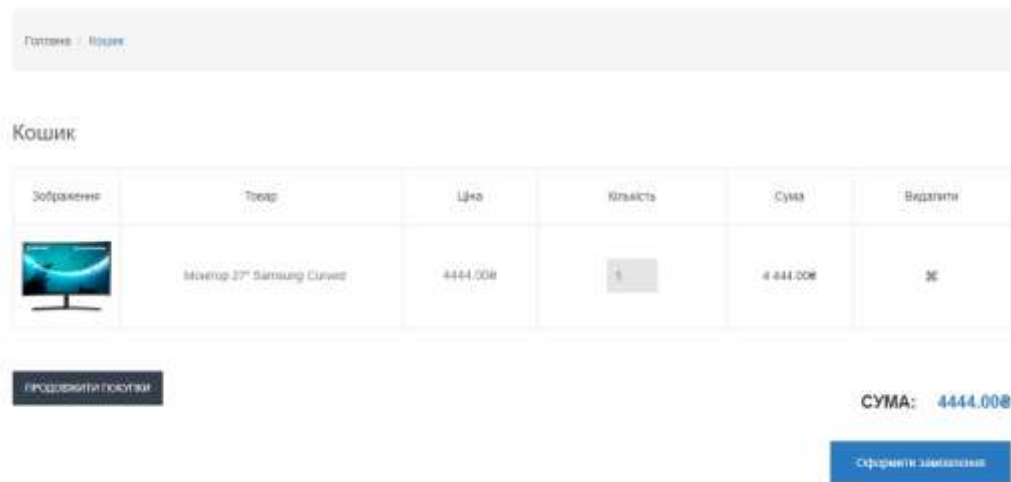


Рисунок 3.3 – Список товарів після редагування

У ході тестування було отримано такі результати:

- Програма працює правильно в останніх стабільних версіях усіх популярних браузерів: Internet Explorer, Google Chrome, Mozilla Firefox, Opera.
- Серед наявних гіперпосилань відсутні неактивні чи неправильні. Переходи на інші сторінки ресурсу здійснюються швидко і точно.
- Після перевірки програми на різних розмірах дисплеїв виявилось, що контент відображається правильно так як і задумав автор.
- Зображення відображаються вірно. Фонове зображення адаптується під розміри дисплея. Шрифт, яким виконано веб-сайт, також відображається правильно.
- Усі заголовки сторінок відображаються правильно. Логотип веб-ресурсу відображається однаково на всіх сторінках.
- В наповнені веб-ресурсу не знайдено синтаксичних та орфографічних помилок.
- Текстові поля веб-ресурсу читаються та сприймаються без труднощів.

На рисунку 3.4 зображена форма реєстрації нового покупця, для занесення відповідної інформації у базу даних. В подальшому ці дані можна змінювати у налаштуваннях акаунта.

Для успішного проходження реєстрації на ресурсі потрібно правильно заповнити реєстраційні поля: Ім'я (більше трьох символів), Email, пароль (більше восьми символів).



Головна / Реєстрація

Реєстрація

* Ім'я

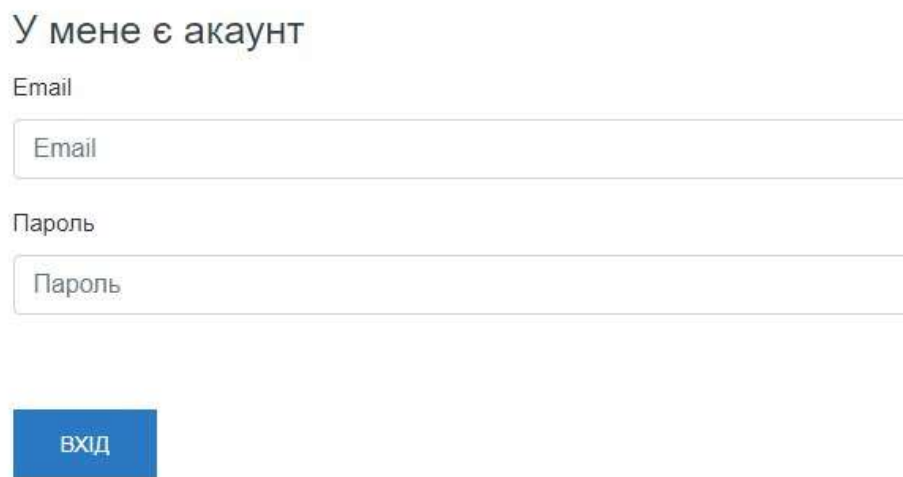
* Email

* Пароль

[ЗАРЕЄСТРУВАТИСЬ](#)

Рисунок 3.4 – Реєстрація користувача

Якщо користувач уже зареєстрований у системі, йому потрібно у формі вказати лише Email та пароль, вказані при реєстрації (рис. 3.5).



У мене є акаунт

Email

Пароль

[ВХІД](#)

Рисунок 3.5 – Авторизація користувача

В асортименті магазину представлено велику кількість товарів, ви можете переглянути будь-який, який Вам сподобався і додати в кошик. Переглянути товарів можна на сторінках «Каталог» і «Головна». На рисунку 3.6 представлена частина асортименту наявних товарів.

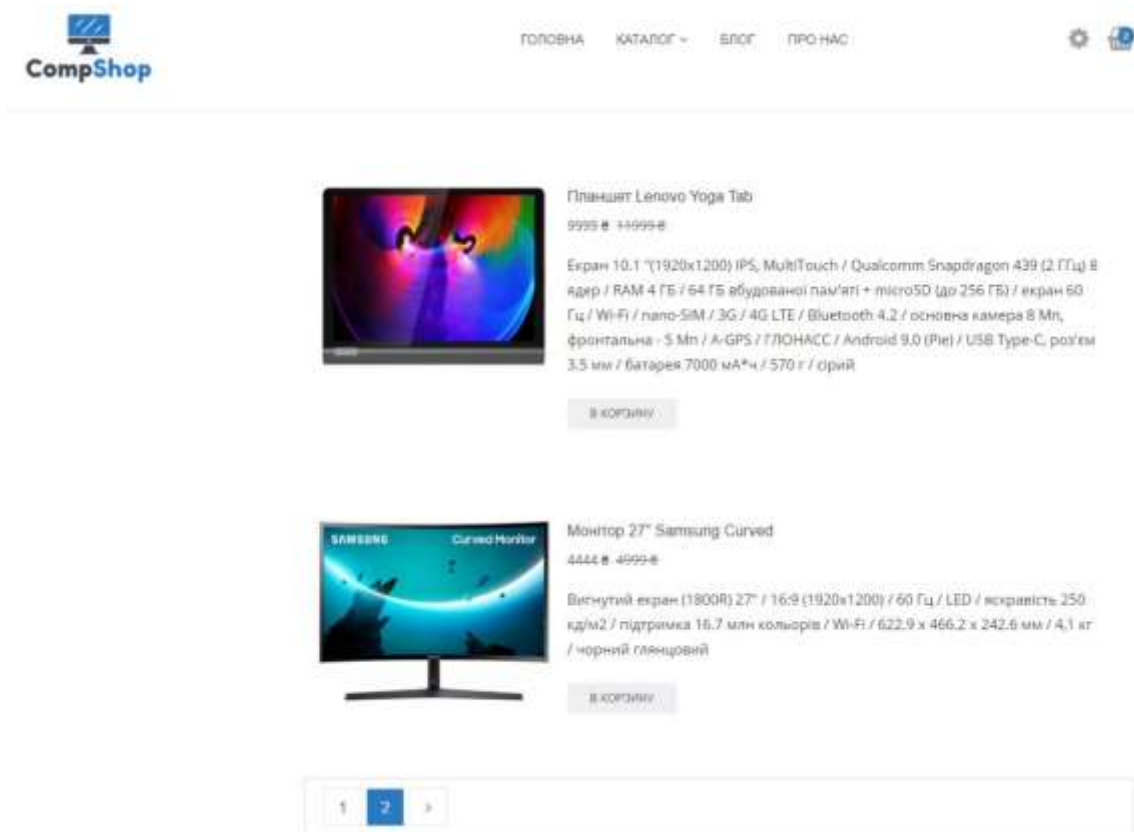


Рисунок 3.6 – Каталог товарів

Список товарів, які додані у кошик можна регулювати за власним бажанням, видаляючи непотрібні та оформлюючи замовлення на ті, які сподобались.

Оформлення замовлення відбувається за декілька кроків.

Якщо користувач увійшов у свій обліковий запис, йому на електронну пошту прийде повідомлення, яке буде містити інформацію про замовлення. В майбутньому продавець зв'яжеться з ним для уточнення даних.

Якщо ж вхід не був здійснений, то клієнту буде запропоновано вказати адресу електронної пошти, по якій з ним можна буде зв'язатись.

Також було здійснено перевірку програмного продукту за допомогою HTML-валідатора. Перевірка показала, що в коді немає помилок. На рисунку 3.7 зображено результат перевірки сторінки index.html.

This document was successfully checked as HTML5!

Result:	Passed, 1 warning(s)	
File :	<input type="text" value="Выберите файл index.html"/> <small>Use the file selection box above if you wish to re-validate the uploaded file index.html</small>	
Encoding :	utf-8	<input type="button" value="(detect automatically)"/>
Doctype :	HTML5	<input type="button" value="(detect automatically)"/>
Root Element:	html	

Рисунок 3.7 – Тестування сторінки index.html

Подібним чином здійснено перевірку програмного продукту за допомогою CSS-валідатора. На рисунку 3.8 зображено результат тестування.

W3C результати перевірки CSS для style.css (CSS3)

Поздравляем! Ошибок не обнаружено.

Этот документ проходит проверку по стандарту [CSS3](#) !

Рисунок 3.8 – Тестування style.css

Розроблений веб-ресурс був протестований серед користувачів в галузі ІТ. Для тестування на достовірність правдивості відповідей було обрано десять спеціалістів, що працюють в ІТ-галузі. Оцінка роботи виставлялась від 1 до 10, в залежності від того була дана програма корисною чи ні. Результати тестування наведені в таблиці 3.3

Таблиця 3.3 – Результати тестування веб-ресурсу з продажу комп'ютерної техніки

Спеціаліст	1	2	3	4	5	6	7	8	9	10
Оцінка	8	8	8	10	8	9	8	7	9	10

Проаналізувавши дані, маємо загальну оцінку 85 із 100. Отже, можна дійти висновку що якість функціонування веб-ресурсу з продажу комп'ютерної техніки

складає 85%. Це свідчить про досить високий рівень підвищення якості процесу он-лайн продажу комп'ютерної техніки.

3.6 Висновок до розділу 3

Під час виконання третього розділу було проаналізовано основні засоби реалізації інформаційної технології з продажу комп'ютерної техніки, обрано середовище розробки Visual Studio Code і обрано оптимальні засоби для розробки Інтернет-магазину. Аналіз порівняння показав, що найефективнішим для візуального відображення буде використання HTML5, CSS3. Для створення бази даних використовується СУБД MySQL. Було розроблено різноманітні програмні модулі, для розробки яких використано скриптові мови програмування JavaScript та PHP.

Проведено повноцінне тестування ресурсу. Помилки при роботі ресурсу не виявлено, під час проведення тестування було виявлено, що сайт відображається однаково в популярних браузерях. Перевірка сайту на usability показала, що сайт є зручним для користувача, має зрозумілу структуру, мобільна версія працює без нарікань. Основні програмні модулі працюють коректно. Також під час перевірки за допомогою HTML та CSS-валідаторів не виявлено помилок коду.

4 ЕКОНОМІЧНА ЧАСТИНА

Для ефективного впровадження науково-технічної розробки важливо, щоб вона відповідала сучасним вимогам науково-технічного прогресу та враховувала економічні аспекти. Оцінка економічної ефективності результатів науково-дослідної роботи є необхідною складовою цього процесу. Представлене в магістерській роботі дослідження, присвячене розробці та вивченню «Інформаційна технологія з продажу комп'ютерної техніки», відноситься до науково-технічних робіт, спрямованих на виведення на ринок, і цей вихід на ринок може бути прийнятий під час проведення самої роботи. Цей напрямок розглядається як комерціалізація науково-технічної розробки, і він вважається пріоритетним, оскільки результати можуть бути корисними для різних зацікавлених сторін, приносячи економічні вигоди. Проте для успішного втілення цього процесу важливо знайти зацікавленого інвестора, який був би зацікавлений у реалізації даного проекту, і переконати його в обґрунтованості вкладання інвестицій у цю розробку.

Для цього визначені наступні етапи виконання робіт:

1. Проведено комерційний аудит науково-технічної розробки, включаючи визначення науково-технічного рівня та комерційного потенціалу.
2. Розраховані витрати на реалізацію науково-технічної розробки.
3. Проведено розрахунок економічної ефективності науково-технічної розробки у разі її впровадження та комерціалізації потенційним інвестором, а також обґрунтовано економічну доцільність комерціалізації для інвестора.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія з продажу комп'ютерної техніки» є розширення функціональних можливостей програмного забезпечення для захисту файлів.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в таблиці 4.1 [20].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція не підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено на працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Продовження таблиці 4.1

Практична здійсненість					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 4.2. Для оцінки науково-технічного рівня і комерційного потенціалу розробки експертами було запрошено трьох незалежних експертів: професор кафедри комп'ютерних наук Вінницького національного технічного університету Яровий А.А.; доцент кафедри комп'ютерних

наук Вінницького національного технічного університету Озеранський В. С., директор ТОВ «ІТІ» Нікітчук В.В.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	Озеранський В.С.	Яровий А.А.	Нікітчук В.В.
	Бали, виставлені експертами:		
1. Технічна здійсненність концепції	5	4	5
2. Ринкові переваги (наявність аналогів)	3	4	3
3. Ринкові переваги (ціна продукту)	4	4	3
4. Ринкові переваги (технічні властивості)	3	3	4
5. Ринкові переваги (експлуатаційні витрати)	3	3	3
6. Ринкові перспективи (розмір ринку)	4	3	3
7. Ринкові перспективи (конкуренція)	3	2	2
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	4	5	5
11. Практична здійсненність (термін реалізації)	5	4	5
12. Практична здійсненність (розробка документів)	5	5	4
Сума балів	СБ ₁ =46	СБ ₂ =45	СБ ₃ =44
Середньоарифметична сума балів $СБ_c$	45,0		

За результатами розрахунків, наведених в табл. 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в таблиці 4.3 [20].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія з продажу комп'ютерної техніки» становить 45 бали, що, відповідно до таблиці 4.3 рівень комерційного потенціалу розробки високий, що свідчить про комерційну важливість проведення даних досліджень.

Магістерська кваліфікаційна робота «Інформаційна технологія з продажу комп'ютерної техніки» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок, тобто при цьому відбувається комерціалізація науково-технічної розробки. Цей напрямок є для нас пріоритетним, оскільки результатами розробки можуть користуватися не тільки самі розробники, а й інші споживачі, отримуючи при цьому суттєвий економічний ефект.

4.2 Визначення рівня конкурентоспроможності розробки

В процесі визначення економічної ефективності науково-технічної розробки також доцільно провести прогноз рівня її конкурентоспроможності за сукупністю параметрів, що підлягають оцінюванню.

В якості аналога для розробки було обрано он-лайн сервіс Telemart.ua

Основними недоліками аналога є відсутність зручної навігації по сайту. Також до недоліків можна віднести відсутність надання рекомендацій по вибору товару. У розробці дана проблема вирішується додаванням рекомендацій по вибору схожих за технічними характеристиками товарів а також наявність зручного і інтуїтивно зрозумілого інтерфейсу.

Також система випереджає аналог за такими параметрами як надання рекомендація під час вибору товарів, підвищеною швидкістю, зручним інтерфейсом.

Одиничний параметричний індекс розраховуємо за формулою [20]:

$$q_i = \frac{P_i}{P_{базі}} \quad (4.1)$$

де q_i – одиничний параметричний індекс, розрахований за i -м параметром;

P_i – значення i -го параметра виробу;

$P_{базі}$ – аналогічний параметр базового виробу-аналога, з яким проводиться порівняння.

Загальні технічні та економічні характеристики розробки представлено в таблиці 4.4.

Таблиця 4.4 – Основні техніко-економічні показники аналога та розробки, що проектується

Показник	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (товар-конкурент)	Новий (інноваційне рішення)		
1	2	3	4	5
Швидкість запуску	7	1,5	4,7	20%
Швидкість доступу до даних	210	140	1,5	25%
Кількість запитів за секунду (qps)	70	231	3,3	25%
Місце на диску	890	240	3,7	30%

Нормативні параметри оцінюємо показником, який отримує одне з двох значень: 1 – пристрій відповідає нормам і стандартам; 0 – не відповідає.

Груповий показник конкурентоспроможності за нормативними параметрами розраховуємо як добуток частинних показників за кожним параметром за формулою [20]:

$$I_{HP} = \prod_{i=1}^n q_i, \quad (4.2)$$

де I_{HP} – загальний показник конкурентоспроможності за нормативними параметрами;

q_i – одиничний (частинний) показник за i -м нормативним параметром;

n – кількість нормативних параметрів, які підлягають оцінюванню.

За нормативними параметрами розроблюваний пристрій відповідає вимогам ДСТУ, тому $I_{HP} = 1$.

Значення групового параметричного індексу за технічними параметрами визначаємо з урахуванням вагомості (частки) кожного параметра [20]:

$$I_{TP} = \sum_{i=1}^n q_i \cdot \alpha_i, \quad (4.3)$$

де I_{TP} – груповий параметричний індекс за технічними показниками (порівняно з виробом-аналогом);

q_i – одиничний параметричний показник i -го параметра;

α_i – вагомість i -го параметричного показника, $\sum_{i=1}^n \alpha_i = 1$;

n – кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

Проведемо аналіз параметрів згідно даних табл. 4.4.

$$I_{mn} = 4,7 \cdot 0,2 + 1,5 \cdot 0,25 + 3,3 \cdot 0,25 + 3,7 \cdot 0,3 = 3,25.$$

Груповий параметричний індекс за економічними параметрами розраховуємо за формулою [20]:

$$I_{EP} = \sum_{i=1}^m q_i \cdot \beta_i, \quad (4.4)$$

де I_{EP} – груповий параметричний індекс за економічними показниками;

q_i – економічний параметр i -го виду;

β_i – частка i -го економічного параметра, $\sum_{i=1}^m \beta_i = 1$;

m – кількість економічних параметрів, за якими здійснюється оцінювання.

Проведемо аналіз параметрів згідно даних таблиці .

$$I_{EP} = 0,75 \cdot 0,5 + 0,84 \cdot 0,5 = 0,80.$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розрахуємо інтегральний показник конкурентоспроможності за формулою [20]:

$$K_{INT} = I_{HP} \cdot \frac{I_{TP}}{I_{EP}}, \quad (4.5)$$

$$K_{INT} = 1 \cdot 3,25 / 0,80 = 4,06.$$

Інтегральний показник конкурентоспроможності $K_{INT} > 1$, отже розробка переважає відомі аналоги за своїми техніко-економічними показниками.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія з продажу комп'ютерної техніки», під час планування,

обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [20]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.6)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=21$ дні.

$$Z_o = 25000 \cdot 30 / 21 = 34091 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	17500	795,5	46	36591
Інженер-програміст	16500	750,0	40	30000
Консультант (аналітик-соціолог)	16500	750,0	7	5250
Технік	17300	786,4	44	34600
Всього				106441

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія з продажу комп'ютерної техніки» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.7)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.8)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $M_M=6500$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [20];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 21$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 6500,00 \cdot 1 \cdot 1,65 / (21 \cdot 8) = 65,8 \text{ грн.}$$

$$Z_{p1} = 65,8 \cdot 5 = 329 \text{ грн.}$$

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.9)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{дод}} = (106441 + 1868,8) \cdot 11 / 100\% = 11914,07 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.6.

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Встановлення обладнання для проведення досліджень	5	1	65,8	329,0
Інсталяція програмного забезпечення розробки програмного забезпечення	1	1	65,8	65,8
Підготовка тестового дослідження	4	3	88,8	355,3
Формування бази даних результатів випробування системи	10	5	111,9	1118,7
Всього				1868,8

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{zn}}{100\%} \quad (4.10)$$

де H_{zn} – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (106441 + 1868,8 + 11914,07) \cdot 22 / 100\% = 26449,24 \text{ грн.}$$

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія з продажу комп'ютерної техніки».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{ej}, \quad (4.11)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

C_{ej} – вартість відходів j -го найменування, грн/кг.

Проведені розрахунки зведемо до таблиці 4.7.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Вартість витраченого матеріалу, грн
Папір А 4	185	1	185
Ручка	25	2	25
Диск оптичний OPTIMA CD	15	2	30
Flesh-пам'ять GOODRAM 64 С10А	380	1	380
Всього			620
З врахуванням коефіцієнта транспортування			682

Витрати на комплектуючі (K_6), які використовують при проведенні НДР на тему «Інформаційна система підтримки підприємств малого бізнесу у сфері послуг» відсутні.

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.}i} \cdot K_i, \quad (4.12)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{\text{пр.}i}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10 \dots 1,12$);

k – кількість найменувань устаткування.

$$B_{\text{спец}} = 11800 \cdot 1 \cdot 1,11 = 12980 \text{ грн.}$$

Отримані результати зведемо до таблиці 4.8.

Таблиця 4.8 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Комп'ютер Core I3-2700	1	11800	12980
Смартфон Iphone X	1	12000	13200
Всього			28798

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для

проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{npz} = \sum_{i=1}^k C_{inprz} \cdot C_{npz.i} \cdot K_i, \quad (4.13)$$

де C_{inprz} – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{npz.i}$ – кількість одиниць програмного забезпечення відповідного найменування,

які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань програмних засобів.

$$B_{npz} = 8400 \cdot 1 \cdot 1,11 = 9240 \text{ грн.}$$

Отримані результати зведемо до таблиці 4.9.

Таблиця 4.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Прикладне середовище розробки ПЗ VSCode	1	8400	9240
Програмне забезпечення SQLite server	1	9300	10230
Всього			19470

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_{об}}{T_е} \cdot \frac{t_{вик}}{12}, \quad (4.14)$$

де C_{σ} – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

T_{σ} – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (23899 \cdot 1) / (2 \cdot 12) = 2987,38 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.10.

Таблиця 4.10 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук Lenovo ThinkPad T480	23899	2	3	2987,38
Printer Xerox Phazer 3040	4500	2	2	375,00
Приміщення лабораторії	240000	20	2	2000,00
Всього				5362,38

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (4.15)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 7,5$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,25 \cdot 290,0 \cdot 7,5 \cdot 0,5 / 0,8 = 339,84 \text{ грн.}$$

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія з продажу комп'ютерної техніки» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (4.16)$$

де H_{cv} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cv} = 20\%$.

$$B_{cv} = (106441 + 1868,8) \cdot 20 / 100\% = 21661,95 \text{ грн.}$$

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ie}}{100\%}, \quad (4.17)$$

де H_{ie} – норма нарахування за статтею «Інші витрати», прийmemo $H_{ie} = 50\%$.

$$I_e = (106441 + 1868,8) \cdot 50 / 100\% = 54154,87 \text{ грн.}$$

До статті «Накладні (загальновиборнічі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиборнічі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.18)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиборнічі) витрати», прийmemo $H_{нзв} = 100\%$.

$$B_{нзв} = (106441 + 1868,8) \cdot 100 / 100\% = 108309,73 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія з продажу комп'ютерної техніки». розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{дод} + Z_n + M + K_e + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сп} + I_e + B_{нзв}. \quad (4.19)$$

$$B_{заг} = 106441 + 1868,8 + 11914,07 + 26449,24 + 682 + 28798 + 19470 + 5362,38 + 339,84 + 21661,95 + 54154,87 + 108309,73 = 385451,8 \text{ грн.}$$

Загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{заг}}{\eta}, \quad (4.20)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta = 0,5$.

$$ЗВ = 385451,8 / 0,5 = 770903,59 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Інформаційна технологія з продажу комп'ютерної техніки» передбачають комерціалізацію протягом 3-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, прийmemo 1 особа

C_o – вартість послуги у році до впровадження інформаційної системи, прийmemo 1100,00 грн;

$\pm \Delta C_o$ – зміна вартості послуги від впровадження результатів, прийmemo зростання на 300,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 3-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [20]:

$$\Delta \Pi_i = (\pm \Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho^3}{100}\right), \quad (4.21)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).
Прийmemo $\rho = 40\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (1 \cdot 300 + 1100 \cdot 3400) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 813185,39 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (1 \cdot 300 + 1100 \cdot (3400 + 2800)) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 1483074 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (1 \cdot 300 + 1100 \cdot (3400 + 2800 + 2600)) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 2104882,5 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.22)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 18\%$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} III &= 813185,39 / (1+0,18)^1 + 1483074 / (1+0,18)^2 + 2104882,5 / (1+0,18)^3 = \\ &= 2931335,89 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (4.23)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв} = 2$;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 770903,59 грн.

$$PV = k_{инв} \cdot 3B = 2 \cdot 770903,59 = 1541807,19 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = III - PV \quad (4.24)$$

де III – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 2931335,89 грн;

PV – теперішня вартість початкових інвестицій, 1541807,19 грн.

$$E_{абс} = III - PV = 2931335,89 - 1541807,19 = 1389528,69 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_e = T_{жс} \sqrt[3]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.25)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_{жс}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 3 роки.

$$E_e = T_{жс} \sqrt[3]{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 1389528,69 / 1541807,19)^{1/3} - 1 = 0,41.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (4.26)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,1$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,25.

$\tau_{мін} = 0,1 + 0,25 = 0,35 < 1,07$ свідчить про те, що внутрішня економічна дохідність інвестицій E_e , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія онтологічного моделювання бази знань з організації бібліотеки» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_e}, \quad (4.27)$$

де E_g – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,41 = 2,4 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновки до розділу 4

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія з продажу комп'ютерної техніки» становить 45 балів, що, свідчить про комерційну важливість проведення даних досліджень оскільки рівень комерційного потенціалу розробки високий.

При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 4,06 рази.

Також термін окупності становить 2,4 роки, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія з продажу комп'ютерної техніки».

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи було розроблено інформаційну технологію з продажу комп'ютерної техніки.

В роботі розглянуто предметну область та актуальність проблеми створення інформаційної технології з продажу комп'ютерної техніки. Електронна комерція активно розвивається і заповнює усі сфери діяльності. Оф-лайн продажі поступово переходять в Інтернет, а це в свою чергу викликає необхідність створення інформаційної технології з продажу комп'ютерної техніки у вигляді інтернет-магазину. Проаналізовані можливі види структур інформаційної технології і вирішено було використовувати блокову. Для реалізації інформаційної технології з продажу комп'ютерної техніки вона підійде найкраще, оскільки є найбільш оптимальною і зрозумілою для користувача. Також було проведено аналіз інформаційного забезпечення сайту. Вирішено, які програмні модулі будуть реалізовувати основний функціонал сайту, з яких сторінок буде складатися ресурс і яка інформація буде на них знаходитись. Проаналізовано основні засоби реалізації інформаційної технології з продажу комп'ютерної техніки, обрано середовище розробки Visual Studio Code і обрано оптимальні засоби для розробки Інтернет-магазину. Аналіз порівняння показав, що найефективнішим для візуального відображення буде використання HTML5, CSS3. Для створення бази даних використовується СУБД MySQL. Було розроблено різноманітні програмні модулі, для розробки яких використано скриптові мови програмування JavaScript та PHP.

Проведено повноцінне тестування ресурсу. Помилки при роботі ресурсу не виявлено, під час проведення тестування було виявлено, що сайт відображається однаково в популярних браузерях. Перевірка сайту на usability показала, що сайт є зручним для користувача, має зрозумілу структуру, мобільна версія працює без нарікань. Основні програмні модулі працюють коректно. Також під час перевірки за допомогою HTML та CSS-валідаторів не виявлено помилок коду.

При тестуванні створеного веб-ресурсу спеціалістами в ІТ-галузі зроблено висновок, що якість функціонування веб-ресурсу з продажу комп'ютерної техніки

складає 85%. Це свідчить про досить високий рівень підвищення якості процесу онлайн продажу комп'ютерної техніки.

Проведено оцінювання комерційного потенціалу розробки інформаційної віртуального ігрового автомата, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, які склали 770903,59 грн.

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія з продажу комп'ютерної техніки» становить 45 балів, що, свідчить про комерційну важливість проведення даних досліджень оскільки рівень комерційного потенціалу розробки високий.

При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 4,06 рази.

Також термін окупності становить 2,4 роки, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже, всі задачі магістерського дослідження виконані, мету роботи досягнуто.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Волков А.О., Озеранський В.С. Структура інформаційної технології з продажу комп'ютерної техніки в Матеріали Всеукраїнської науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2023)» – [Електронний ресурс]. – <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/19741>
2. Що таке електронна комерція? Е-commerce для початківців [електронний ресурс] // Режим доступу: <https://www.interkassa.com/ua/blog/chto-takoe-elektronnaya-kommerciya-e-commerce-dlya-nachinayushchih>
3. Бурбело С.М. Розробка програмних засобів реалізації торговельного майданчика для продажу комп'ютерної техніки / В.В. Войтко, Л.М. Круподьорова, А.О. Волков // Науково-технічна конференція ВНТУ «Молодь в науці: дослідження, проблеми, перспективи (МН-2021)» – Вінниця, 2021. [Електронний ресурс] – Режим доступу до ресурсу: <http://kntu.net.ua/ukr/content/view/full/58984>.
4. Для чого потрібна обчислювальна техніка [електронний ресурс] // Режим доступу: <https://vism.ru/what-is-computing-technology-for-electronic-computers.html>.
5. Інтернет-магазин «Rozetka» [електронний ресурс] // Режим доступу: <https://rozetka.com.ua/> – Назва з екрану.
6. Інтернет-магазин «Telemart.ua» [електронний ресурс] // Режим доступу: <https://www.telemart.ua>
7. Інтернет-магазин «ЗОНА51» [електронний ресурс] // Режим доступу: <https://www.zona51.com>
8. Романюк О.Н. Веб-дизайн і комп'ютерна графіка. Навчальний посібник / О.Н. Романюк, Д.І. Кательніков, О.П. Косоветь. – В.: ВНТУ, 2007. – 147с.
9. PHP [електронний ресурс] // Режим доступу: <https://uk.wikipedia.org/wiki/PHP>

10. Цікаві факти про Інтернет-торгівлю в світі [електронний ресурс] // Режим доступу:<https://lemarbet.com/ua/razvitie-internet-magazina/interesnye-fakty-ob-internet-torgovle>
11. Paul Dubois – MySQL Cookbook Solutions for database developers and administrators. – O’Reilly.
12. Eloquent ORM. [Електронний ресурс]. – Режим доступу: <https://eloquentbyexample.com/>
13. Інформаційні бази [Електронний ресурс]. URL: <https://its.1c/db/metod8dev/content/1591/hdoc>. /(дата звернення: 2023р.).
14. Робота з базою даних [Електронний ресурс]. URL: <https://v8.1c/platforma/rabota-s-bazoy-dannykh/>(дата звернення: 2023р.).
15. PHP, MySQL та інші технології. [Електронний ресурс] – Режим доступу: <http://www.php.su/>.
16. Integrated development environment. [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/Інтегроване середовище розробки](https://uk.wikipedia.org/wiki/Інтегроване_середовище_розробки).
17. Reasons to Consider Angular Framework For Web Development in 2021 [Електронний ресурс]:[Веб-сайт] – Електронні дані. — Режим доступу: <https://www.monocubed.com/angular-framework-for-web-development/>
18. Why You Should Use React.js For Web Development [Електронний ресурс]:[Веб-сайт] – Електронні дані. — Режим доступу: <https://www.freecodecamp.org/news/why-use-react-for-web-development/63>
19. Online Shopping Personalization – Statistics and Trends[Електронний ресурс]:[Веб-сайт] – Електронні дані. — Режим доступу: <https://www.invespcro.com/blog/online-shopping-personalizat>
20. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

ДОДАТКИ

Додаток А (обов'язковий)

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія з продажу комп'ютерної техніки

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 87,3% Схожість 12,7%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи



Волков А.О.

Керівник роботи



Озеранський В.С.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

```

AdminBase.php
<?php
abstract class AdminBase
{
    public static function checkAdmin()
    {
        $userId = User::checkLogination();

        $user = User::getUserById($userId);

        if ($user['role'] == 'admin') {
            return true;
        }
        die('Access denied');
    }
}

Cart.php
<?php
class Cart
{
    public static function addProduct($id)
    {
        $id = intval($id);
        $productsInCart = array();
        if (isset($_SESSION['products'])) {
            $productsInCart = $_SESSION['products'];
        }
        if (array_key_exists($id, $productsInCart)) {
            $productsInCart[$id] ++;
        } else {
            $productsInCart[$id] = 1;
        }
        $_SESSION['products'] = $productsInCart;

        return self::countItems();
    }

    public static function countItems()
    {
        if (isset($_SESSION['products'])) {
            $count = 0;
            foreach ($_SESSION['products'] as $id => $quantity) {
                $count = $count + $quantity;
            }
            return $count;
        } else {
            return 0;
        }
    }

    public static function getProducts()
    {
        if (isset($_SESSION['products'])) {
            return $_SESSION['products'];
        }
        return false;
    }
}

```

```

    }
    public static function getTotalPrice($products)
    {
        $productsInCart = self::getProducts();
        $total = 0;
        if ($productsInCart) {
            foreach ($products as $item) {
                $total += $item['price'] * $productsInCart[$item['id']];
            }
        }
        return $total;
    }
    public static function clear()
    {
        if (isset($_SESSION['products'])) {
            unset($_SESSION['products']);
        }
    }
    public static function deleteProduct($id)
    {
        $productsInCart = self::getProducts();
        unset($productsInCart[$id]);
        $_SESSION['products'] = $productsInCart;
    }
}
}
Db.php
<?php
class Db
{
    public static function getConnection()
    {
        $paramsPath = ROOT.'/config/db_parameters.php';
        $params = include($paramsPath);
        $db = new PDO("mysql:host={$params['host']};dbname={$params['dbname']}", $params['user'],
$params['password'], [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]);
        $db->exec("set names utf8");
        return $db;
    }
}
}

Pagination.db
<?php
class Pagination
{
    private $max = 10;
    private $index = 'page';
    private $current_page;
    private $total;
    private $limit;
    public function __construct($total, $currentPage, $limit, $index)
    {
        $this->total = $total;
        $this->limit = $limit;
        $this->index = $index;
        $this->amount = $this->amount();
        $this->setCurrentPage($currentPage);
    }
}

    public function get()
        $links = null;
        $limits = $this->limits();
        $html = '<ul class="blog-pagination">';
        for ($page = $limits[0]; $page <= $limits[1]; $page++) {

```

```

    if ($page == $this->current_page) {
        $links .= '<li class="active"><a href="#">' . $page . '</a></li>';
    } else {
        $links .= $this->generateHtml($page);
    }
}
if (!is_null($links)) {
    if ($this->current_page > 1)
        $links = $this->generateHtml(1, '&lt;');
    if ($this->current_page < $this->amount)
        $links .= $this->generateHtml($this->amount, '&gt;');
}
$html .= $links . '</ul>';
return $html;
}
private function generateHtml($page, $text = null)
{
    if (!$text)
        $text = $page;
    $currentURI = rtrim($_SERVER['REQUEST_URI'], '/') . '/';
    $currentURI = preg_replace('~page[0-9]+~', "", $currentURI);
    return
        '<li><a href="' . $currentURI . $this->index . $page . '">' . $text . '</a></li>';
}
private function limits()
{
    $left = $this->current_page - round($this->max / 2);
    $start = $left > 0 ? $left : 1;
    if ($start + $this->max <= $this->amount)
        $end = $start > 1 ? $start + $this->max : $this->max;
    else {
        $end = $this->amount;
        $start = $this->amount - $this->max > 0 ? $this->amount - $this->max : 1;
    }
    return
        array($start, $end);
}
private function setCurrentPage($currentPage)
{
    $this->current_page = $currentPage;
    if ($this->current_page > 0) {
        if ($this->current_page > $this->amount)
            $this->current_page = $this->amount;
        } else
            $this->current_page = 1;
}
private function amount()
{
    return ceil($this->total / $this->limit);
}
}

```

Router.php

```

<?php
class Router
{
    private $routes;
    public function __construct()
    {
        $routesPath = ROOT.'/config/routes.php';
        $this->routes = include($routesPath);
    }
}

```

```

private function getURI()
{
    if (!empty($_SERVER['REQUEST_URI'])) {
        return trim($_SERVER['REQUEST_URI'], '/');
    }
}
public function run()
{
    $uri = $this->getURI();
    foreach ($this->routes as $uriPattern => $path) {
        if (preg_match("~$uriPattern~", $uri)) {
            $internalRoute = preg_replace("~$uriPattern~", $path, $uri);
            $segments = explode('/', $internalRoute);
            $controllerName = array_shift($segments).'Controller';
            $controllerName = ucfirst($controllerName);
            $actionName = 'action'.ucfirst(array_shift($segments));
            $parameters = $segments;
            $controllerFile = ROOT.'/controllers/'.$controllerName.'.php';
            if (file_exists($controllerFile)) {
                include_once($controllerFile);
            }
            $controllerObject = new $controllerName;
            $result = call_user_func_array(array($controllerObject, $actionName), $parameters);
            if ($result != null) {
                break;
            }
        }
    }
}
}
}
}

```

AdminController.php

```

<?php
class AdminCategoryController extends AdminBase
{
    public function actionIndex()
    {
        self::checkAdmin();
        $categoriesList = Category::getCategoriesListAdmin();
        require_once(ROOT . '/views/admin_category/index.php');
        return true;
    }
    public function actionCreate()
    {
        self::checkAdmin();
        if (isset($_POST['submit'])) {
            $name = $_POST['name'];
            $sortOrder = $_POST['sort_order'];
            $status = $_POST['status'];
            $errors = false;
            if (!isset($name) || empty($name)) {
                $errors[] = 'Введіть дані у поля';
            }
            if ($errors == false) {
                Category::createCategory($name, $sortOrder, $status);
                header("Location: /admin/category");
            }
        }
        require_once(ROOT . '/views/admin_category/create.php');
        return true;
    }
    public function actionUpdate($id)
    {
        self::checkAdmin();
        $category = Category::getCategoryById($id);
        if (isset($_POST['submit'])) {
            $name = $_POST['name'];

```

```

        $sortOrder = $_POST['sort_order'];
        $status = $_POST['status'];
        Category::updateCategoryById($id, $name, $sortOrder, $status);
        header("Location: /admin/category");
    }
    require_once(ROOT . '/views/admin_category/update.php');
    return true;
}
public function actionDelete($id)
{
    self::checkAdmin();
    if (isset($_POST['submit'])) {
        Category::deleteCategoryById($id);
        header("Location: /admin/category");
    }
    require_once(ROOT . '/views/admin_category/delete.php');
    return true;
}
}

```

AdminController.php

```

<?php
class AdminController extends AdminBase
{
    public function actionIndex()
    {
        self::checkAdmin();
        require_once(ROOT . '/views/admin/index.php');
        return true;
    }
}

```

AdminOrderController.php

```

<?php
class AdminOrderController extends AdminBase
{
    public function actionIndex()
    {
        self::checkAdmin();
        $ordersList = Order::getOrdersList();
        require_once(ROOT . '/views/admin_order/index.php');
        return true;
    }
    public function actionUpdate($id)
    {
        self::checkAdmin();
        $order = Order::getOrderById($id);
        if (isset($_POST['submit'])) {
            $userName = $_POST['userName'];
            $userPhone = $_POST['userPhone'];
            $userComment = $_POST['userComment'];
            $date = $_POST['date'];
            $status = $_POST['status'];
            Order::updateOrderById($id, $userName, $userPhone, $userComment, $date, $status);
            header("Location: /admin/order/view/$id");
        }
        require_once(ROOT . '/views/admin_order/update.php');
        return true;
    }
}

```

```

public function actionView($id)
{
    self::checkAdmin();
    $order = Order::getOrderById($id);
    $productsQuantity = json_decode($order['products'], true);
}

```

```

        $productsIds = array_keys($productsQuantity);
        $products = Product::getProductsByIds($productsIds);
        require_once(ROOT . '/views/admin_order/view.php');
        return true;
    }
    public function actionDelete($id)
    {
        self::checkAdmin();
        if (isset($_POST['submit'])) {
            Order::deleteOrderById($id);
            header("Location: /admin/order");
        }
        require_once(ROOT . '/views/admin_order/delete.php');
        return true;
    }
}

```

AdminProductController.php
<?php

```

class AdminProductController extends AdminBase
{
    public function actionIndex()
    {
        self::checkAdmin();
        $productsList = Product::getProductsList();

        require_once(ROOT . '/views/admin_product/index.php');
        return true;
    }
    public function actionCheckout()
    {
        $productsInCart = Cart::getProducts();
        if ($productsInCart == false) {
            header("Location: /");
        }
        $categories = Category::getCategoriesList();
        $productsIds = array_keys($productsInCart);
        $products = Product::getProductsByIds($productsIds);
        $totalPrice = Cart::getTotalPrice($products);
        $totalQuantity = Cart::countItems();
        $userName = false;
        $userPhone = false;
        $userComment = false;
        $result = false;
        if (!User::isGuest()) {
            $userId = User::checkLogged();
            $user = User::getUserById($userId);
            $userName = $user['name'];
        } else {
            $userId = false;
        }
        if (isset($_POST['submit'])) {
            $userName = $_POST['userName'];
            $userPhone = $_POST['userPhone'];
            $userComment = $_POST['userComment'];
            $errors = false;
            if (!User::checkName($userName)) {
                $errors[] = "Неправильне ім'я";
            }
            if (!User::checkPhone($userPhone)) {
                $errors[] = "Неправильний телефон";
            }
        }
    }
}

```



```

if ($errors == false) {
    $result = Order::save($userName, $userPhone, $userComment, $userId, $productsInCart);
    if ($result) {
        $adminEmail = 'volkov989989@gmail.com';
        $subject = 'Нове замовлення!';
        mail($adminEmail, $subject, $message);
        Cart::clear();
    }
}
}
require_once(ROOT . '/views/cart/checkout.php');
return true;
}}

```

CatalogController.php

```

<?php
include_once ROOT . '/models/User.php';
include_once ROOT . '/models/Category.php';
include_once ROOT . '/models/Product.php';
include_once ROOT . '/components/Pagination.php';
class CatalogController
{
    public function actionIndex()
    {
        $guest = User::guest();
        $categories = array();
        $categories = Category::getCategories();
        $latestProducts = array();
        $latestProducts = Product::getLatestProducts(20);
        require_once(ROOT . '/views/catalog/index.php');
        return true;
    }
    public function actionCategory($categoryId, $page = 1)
    {
        $guest = User::guest();
        $categories = array();
        $categories = Category::getCategories();
        $categoryProducts = array();
        $categoryProducts = Product::getProductsByCategory($categoryId, $page);
        $total = Product::getAllProductsFromCategory($categoryId);
        $pagination = new Pagination($total, $page, Product::DEFAULT, 'page');
        require_once(ROOT . '/views/catalog/category.php');
        return true;
    }
}
}

```

ProductController.php

```

<?php
include_once ROOT . '/models/Category.php';
include_once ROOT . '/models/Product.php';
include_once ROOT . '/models/User.php';
class ProductController
{
    public function actionView($productId)
    {
        $guest = User::guest();
        $categories = array();
        $categories = Category::getCategories();
        $product = Product::getProductById($productId);
        require_once(ROOT . '/views/product/view.php');
        return true;
    }
}
}

```

```

SiteController.php
<?php
include_once ROOT.'/models/Category.php';
include_once ROOT.'/models/Product.php';
include_once ROOT.'/models/User.php'
class SiteController
{
public function actionIndex()
{
    $guest = User::guest();
    $categories = array();
    $categories = Category::getCategories();
    $latestProducts = array();
    $latestProducts = Product::getLatestProducts(4);
    require_once(ROOT.'/views/site/index.php');
    return true;
}
public function actionContact()
{
    $userEmail = false;
    $userText = false;
    $result = false;
    if (isset($_POST['submit'])) {
        $userEmail = $_POST['userEmail'];
        $userText = $_POST['userText'];
        $errors = false;
        if (!User::checkEmail($userEmail)) {
            $errors[] = 'Неправильний email';
        }
        if ($errors == false) {
            $adminEmail = 'volkov989989@gmail.com';
            $message = "Текст: {$userText}. Від {$userEmail}";
            $subject = 'Тема';
            $result = mail($adminEmail, $subject, $message);
            $result = true;
        }
    }
    require_once(ROOT . '/views/site/contact.php');
    return true;
}
public function actionAbout()
{
    require_once(ROOT . '/views/site/about.php');
    return true;
}
}
}

```

```

UserController.php
<?php
include_once ROOT.'/models/Category.php';
include_once ROOT.'/models/User.php';
class UserController
{
public function actionRegister()
{
    $guest = User::guest();
    $categories = array();
    $categories = Category::getCategories();
    $name = false;
    $email = false;
    $password = false;
    $result = false;

```

```

if (isset($_POST['submit'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $errors = false;
    if (!User::verifyName($name)) {
        $errors[] = "Ім'я має містити більше трьох символів";
    }
    if (!User::verifyEmail($email)) {
        $errors[] = "Email введений невірно";
    }
    if (!User::verifyPassword($password)) {
        $errors[] = "Пароль має містити більше семи символів";
    }
    if (User::checkEmailPresence($email)) {
        $errors[] = "Користувач з таким email вже зареєстрований на сайті";
    }
    if ($errors == false) {
        $result = User::register($name, $email, $password);
    }
}
require_once(ROOT . '/views/user/register.php');
return true;
}
public function actionLogin()
{
    $guest = User::guest();
    $categories = array();
    $categories = Category::getCategories();
    $email = false;
    $password = false;
    if (isset($_POST['submit'])) {
        $email = $_POST['email'];
        $password = $_POST['password'];
        $errors = false;
        if (!User::verifyEmail($email)) {
            $errors[] = "Email введений невірно";
        }
        if (!User::verifyPassword($password)) {
            $errors[] = "Пароль має містити більше семи символів";
        }
        $user = User::searchUser($email, $password);
        if ($user == false) {
            $errors[] = "Такого користувача не існує";
        } else {
            User::authorization($user);
            header('Location: /account/');
        }
    }
    require_once(ROOT . '/views/user/login.php');
    return true;
}
}
public static function deleteCategoryById($id)
{
    $db = Db::getConnection();
    $sql = 'DELETE FROM category WHERE id = :id';
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    return $result->execute();
}
public static function getStatusText($status)
{

```

```

switch ($status) {
    case '1':
        return 'Відображається';
        break;
    case '0':
        return 'Приховано';
        break;
    }
}

public static function createCategory($name, $sortOrder, $status)
{
    $db = Db::getConnection();
    $sql = 'INSERT INTO category (name, sort_order, status) '
        . 'VALUES (:name, :sort_order, :status)';
    $result = $db->prepare($sql);
    $result->bindParam(':name', $name);
    $result->bindParam(':sort_order', $sortOrder);
    $result->bindParam(':status', $status);
    return $result->execute();
}
}

```

Order.php

```

<?php
class Order
{
    public static function save($userName, $userPhone, $userComment, $userId, $products)
    {
        $db = Db::getConnection();
        $sql = 'INSERT INTO product_order (user_name, user_phone, user_comment, user_id, products) '
            . 'VALUES (:user_name, :user_phone, :user_comment, :user_id, :products)';
        $products = json_encode($products);
        $result = $db->prepare($sql);
        $result->bindParam(':user_name', $userName, PDO::PARAM_STR);
        $result->bindParam(':user_phone', $userPhone, PDO::PARAM_STR);
        $result->bindParam(':user_comment', $userComment, PDO::PARAM_STR);
        $result->bindParam(':user_id', $userId, PDO::PARAM_STR);
        $result->bindParam(':products', $products, PDO::PARAM_STR);
        return $result->execute();
    }
    public static function getOrdersList()
    {
        $db = Db::getConnection();
        $result = $db->query('SELECT id, user_name, user_phone, date, status FROM product_order ORDER BY id DESC');
        $ordersList = array();
        $i = 0;
        while ($row = $result->fetch()) {
            $ordersList[$i]['id'] = $row['id'];
            $ordersList[$i]['user_name'] = $row['user_name'];
            $ordersList[$i]['user_phone'] = $row['user_phone'];
            $ordersList[$i]['date'] = $row['date'];
            $ordersList[$i]['status'] = $row['status'];
            $i++;
        }
        return $ordersList;
    }
    public static function getStatusText($status)
    {
        switch ($status) {
            case '1':
                return 'Нове замовлення';
                break;

```

```

        case '2':
            return 'В обробці';
            break;
        case '3':
            return 'Доставляється';
            break;
        case '4':
            return 'Завершено';
            break;
    }
}
public static function getOrderById($id)
{
    $db = Db::getConnection();
    $sql = 'SELECT * FROM product_order WHERE id = :id';
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->setFetchMode(PDO::FETCH_ASSOC);
    $result->execute();
    return $result->fetch();
}
public static function deleteOrderById($id)
{
    $db = Db::getConnection();
    $sql = 'DELETE FROM product_order WHERE id = :id';
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    return $result->execute();
}
public static function updateOrderById($id, $userName, $userPhone, $userComment, $date, $status)
{
    $db = Db::getConnection();
    $sql = "UPDATE product_order
    SET
        user_name = :user_name,
        user_phone = :user_phone,
        user_comment = :user_comment,
        date = :date,
        status = :status
    WHERE id = :id";
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id);
    $result->bindParam(':user_name', $userName);
    $result->bindParam(':user_phone', $userPhone);
    $result->bindParam(':user_comment', $userComment);
    $result->bindParam(':date', $date);
    $result->bindParam(':status', $status);
    return $result->execute();
}
}
}

```

Product.php

```

<?php
class Product
{
    const SHOW_BY_DEFAULT = 6;
    public static function getLatestProducts($count = self::SHOW_BY_DEFAULT)
    {
        $db = Db::getConnection();
        $sql = 'SELECT id, name, price, is_new FROM product '
            . 'WHERE status = "1" ORDER BY id DESC '
            . 'LIMIT :count';
        $result = $db->prepare($sql);
    }
}

```

```

$result->bindParam(':count', $count);
$result->setFetchMode(PDO::FETCH_ASSOC);
$result->execute();
$i = 0;
$productsList = array();
while ($row = $result->fetch()) {
    $productsList[$i]['id'] = $row['id'];
    $productsList[$i]['name'] = $row['name'];
    $productsList[$i]['price'] = $row['price'];
    $productsList[$i]['is_new'] = $row['is_new'];
    $i++;
}
return $productsList;
}
public static function getProductsListByCategory($categoryId, $page = 1)
{
    $limit = Product::SHOW_BY_DEFAULT;
    $offset = ($page - 1) * self::SHOW_BY_DEFAULT;
    $sql = 'SELECT id, name, price, is_new FROM product '
        . 'WHERE status = 1 AND category_id = :category_id '
        . 'ORDER BY id ASC LIMIT :limit OFFSET :offset';
    $result = $db->prepare($sql);
    $result->bindParam(':category_id', $categoryId);
    $result->bindParam(':limit', $limit);
    $result->bindParam(':offset', $offset);
    $result->execute();
    $i = 0;
    $products = array();
    while ($row = $result->fetch()) {
        $products[$i]['id'] = $row['id'];
        $products[$i]['name'] = $row['name'];
        $products[$i]['price'] = $row['price'];
        $products[$i]['new'] = $row['new'];
        $i++;
    }
    return $products;
}
public static function getProductById($id)
{
    $db = Db::getConnection();
    $sql = 'SELECT * FROM product WHERE id = :id';
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id);
    $result->setFetchMode(PDO::FETCH_ASSOC);
    $result->execute();
    return $result->fetch();
}
public static function getTotalProductsInCategory($categoryId)
{
    $db = Db::getConnection();
    $sql = 'SELECT count(id) AS count FROM product WHERE status="1" AND category_id = :category_id';
    $result = $db->prepare($sql);
    $result->bindParam(':category_id', $categoryId);
    $result->execute();
    $row = $result->fetch();
    return $row['count'];
}
}
public static function getProductsByIds($idsArray)
{
    $db = Db::getConnection();
    $idsString = implode(',', $idsArray);
    $sql = "SELECT * FROM product WHERE status='1' AND id IN ($idsString)";

```

```

$result = $db->query($sql);
$result->setFetchMode(PDO::FETCH_ASSOC);
$i = 0;
$products = array();
while ($row = $result->fetch()) {
    $products[$i]['id'] = $row['id'];
    $products[$i]['name'] = $row['name'];
    $products[$i]['price'] = $row['price'];
    $i++;
}
return $products;
}
public static function getRecommendedProducts()
{
    $db = Db::getConnection();
    $result = $db->query('SELECT id, name, price, new FROM product '
        . 'WHERE status = "1" AND is_recommended = "1" '
        . 'ORDER BY id DESC');
    $i = 0;
    $productsList = array();
    while ($row = $result->fetch()) {
        $productsList[$i]['id'] = $row['id'];
        $productsList[$i]['name'] = $row['name'];
        $productsList[$i]['price'] = $row['price'];
        $productsList[$i]['new'] = $row['new'];
        $i++;
    }
    return $productsList;
}
public static function getProductsList()
{
    $db = Db::getConnection();
    $result = $db->query('SELECT id, name, price FROM product ORDER BY id ASC');
    $productsList = array();
    $i = 0;
    while ($row = $result->fetch()) {
        $productsList[$i]['id'] = $row['id'];
        $productsList[$i]['name'] = $row['name'];
        $productsList[$i]['price'] = $row['price'];
        $i++;
    }
    return $productsList;
}
public static function deleteProductById($id)
{
    $db = Db::getConnection();
    $sql = 'DELETE FROM product WHERE id = :id';
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    return $result->execute();
}
public static function updateProductById($id, $options)
{
    $db = Db::getConnection();
    $sql = "UPDATE product
    SET
        name = :name,
        code = :code,
        price = :price,
        category_id = :category_id,
        brand = :brand,
        availability = :availability,
        text = :text,

```

```

        new = :new,
        recommended = :recommended,
        status = :status
    WHERE id = :id";
$result = $db->prepare($sql);
$result->bindParam(':id', $id);
$result->bindParam(':name', $options['name']);
$result->bindParam(':code', $options['code']);
$result->bindParam(':price', $options['price']);
$result->bindParam(':category_id', $options['category_id']);
$result->bindParam(':brand', $options['brand']);
$result->bindParam(':availability', $options['availability']);
$result->bindParam(':description', $options['description']);
$result->bindParam(':new', $options['new']);
$result->bindParam(':recommended', $options['recommended']);
$result->bindParam(':status', $options['status']);
return $result->execute();
}
public static function createProduct($options)
{
    $db = Db::getConnection();
    $sql = 'INSERT INTO product '
        . '(name, code, price, category_id, brand, availability,'
        . 'text1, new, recommended, status)'
        . 'VALUES '
        . '(:name, :code, :price, :category_id, :brand, :availability,'
        . ':description, :new, :recommended, :status)';
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id);
    $result->bindParam(':name', $options['name']);
    $result->bindParam(':code', $options['code']);
    $result->bindParam(':price', $options['price']);
    $result->bindParam(':category_id', $options['category_id']);
    $result->bindParam(':brand', $options['brand']);
    $result->bindParam(':availability', $options['availability']);
    $result->bindParam(':description', $options['description']);
    $result->bindParam(':new', $options['new']);
    $result->bindParam(':recommended', $options['recommended']);
    $result->bindParam(':status', $options['status']);
    if ($result->execute()) {
        return $db->lastInsertId();
    }
    return 0;
}
public static function getAvailabilityText($availability)
{
    switch ($availability) {
        case '1':
            return 'В наявності';
            break;
        case '0':
            return 'Під замовлення';
            break;
    }
}
public static function getImage($id)
{
    $noImage = 'no-image.jpg';
    $path = '/upload/images/products/';
    $pathToProductImage = $path . $id . '.jpg';
    if (file_exists($_SERVER['DOCUMENT_ROOT'].$pathToProductImage)) {
        return $pathToProductImage;
    }
}

```



```

        return $path . $noImage;
    }
}

```

User.php

```

<?php
include_once ROOT.'/components/Db.php';
class User
{
    public static function register($name, $email, $password)
    {
        $db = Db::getConnection();
        $result = $db->prepare('INSERT INTO user (name, email, password) VALUES (:name, :email, :password)');
        $result->bindParam(':name', $name);
        $result->bindParam(':email', $email);
        $result->bindParam(':password', $password);
        return $result->execute();
    }
    public static function verifyName ($name)
    {
        if (strlen($name) >= 4) {
            return true;
        }
        return false;
    }
    public static function verifyEmail ($email)
    {
        if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
            return true;
        }
        return false;
    }
    public static function verifyPassword ($password)
    {
        if (strlen($password) >= 8) {
            return true;
        }
        return false;
    }
    public static function checkEmailPresence($email)
    {
        $db = Db::getConnection();
        $result = $db->prepare('SELECT COUNT(*) FROM user WHERE email = :email');
        $result->bindParam(':email', $email);
        $result->execute();

        if ($result->fetchColumn())
            return true;

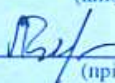
        return false;
    }
    public static function searchUser($email, $password)
    {
        $db = Db::getConnection();
        $result = $db->prepare('SELECT * FROM user WHERE email = :email AND password =
:password');
        $result->bindParam(':email', $email);
        $result->bindParam(':password', $password);
        $result->execute();
        $user = $result->fetch();
        if ($user) {
            return $user['id'];
        }
    }
}

```

```
        return false;
    }
    public static function authorization($user)
    {
        $_SESSION['user'] = $user;
    }
    public static function checkLogination()
    {
        if (isset($_SESSION['user'])) {
            return $_SESSION['user'];
        }
        header('Location: /user/login');
    }
    public static function guest()
    {
        if (isset($_SESSION['user'])) {
            return false;
        }
        return true;
    }
}
```

Додаток В (обов'язковий)**ІЛЮСТРАТИВНА ЧАСТИНА****ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ**

Виконав: студент 2 курсу, групи 2КН-22м
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

 Волков А.О.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

 Озеранський В.С.
(прізвище та ініціали)

«07» 12 _____ 2023 р.

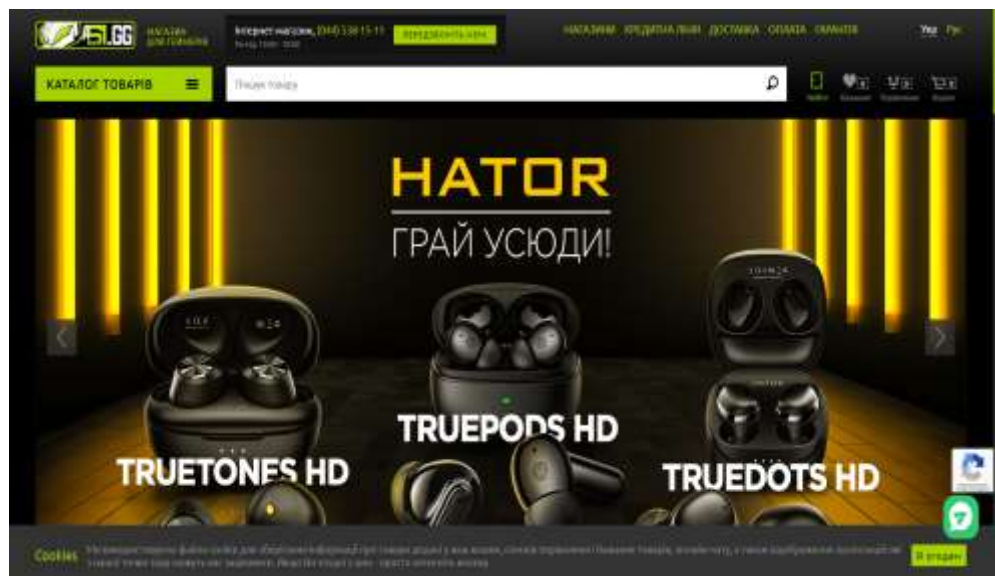
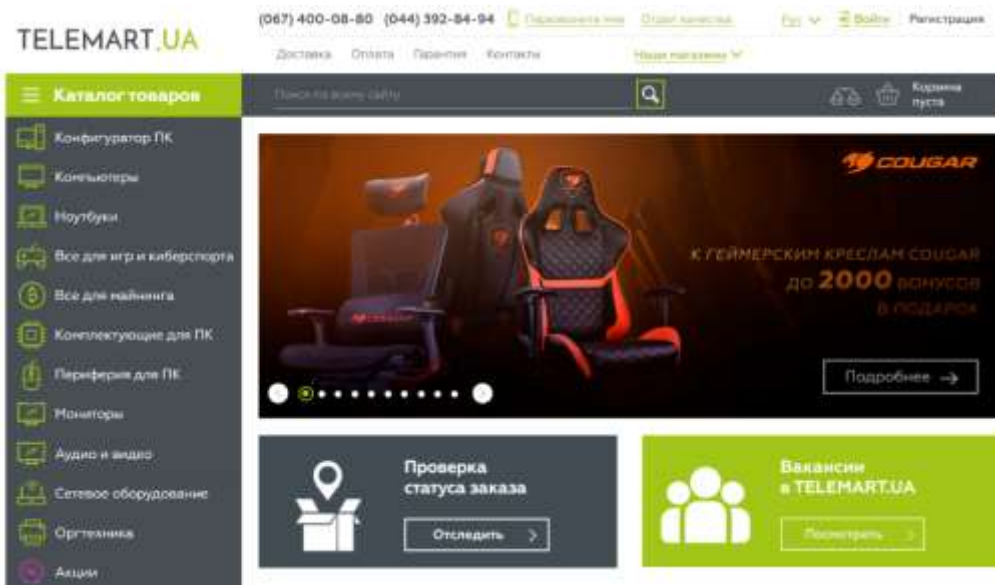
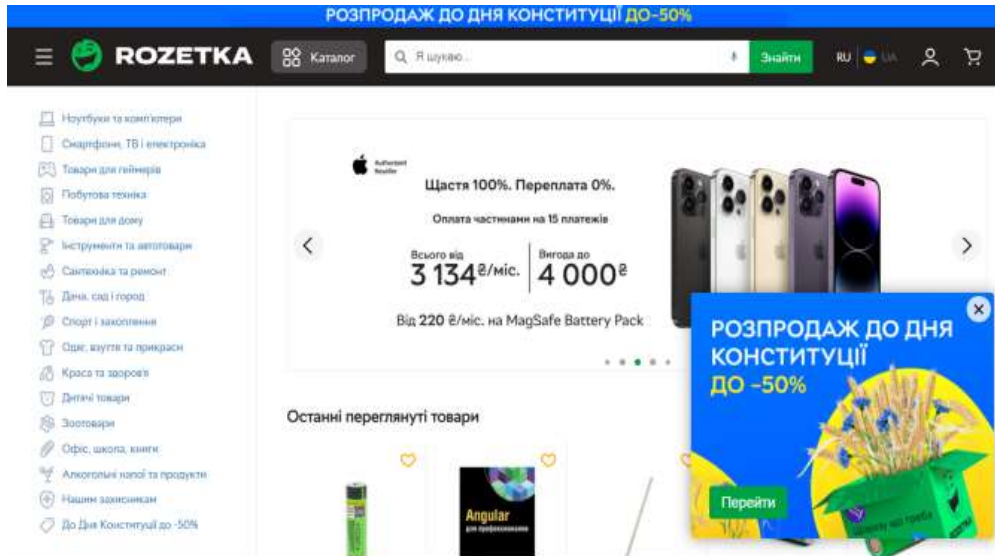


Рисунок В.1 – Порівняльний аналіз аналогів

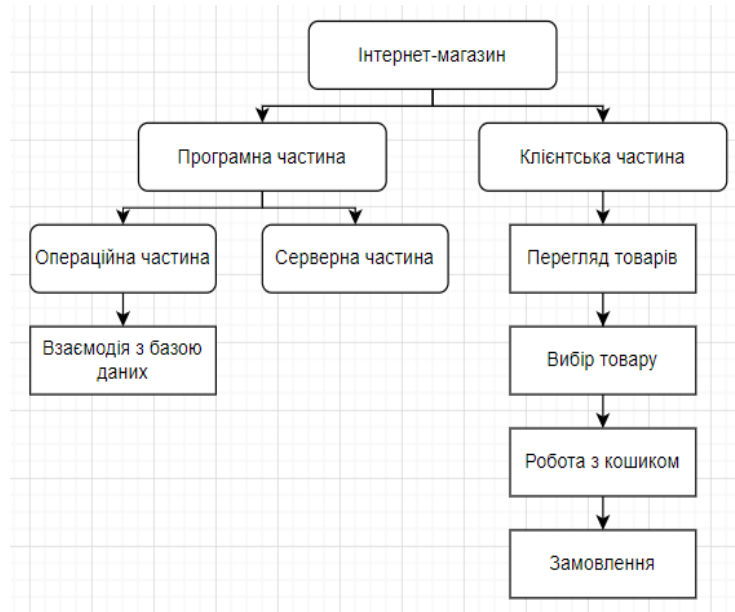


Рисунок В.2 – Структура інформаційної технології з продажу комп'ютерної техніки

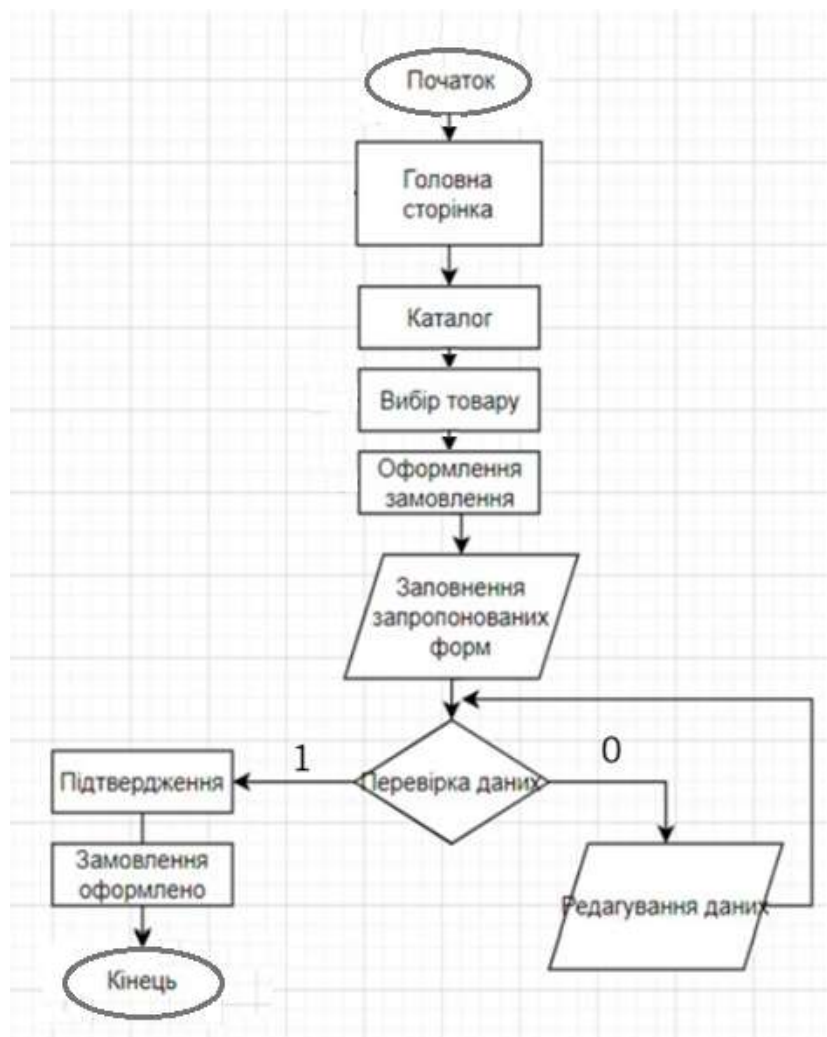


Рисунок В.3 – Алгоритм функціонування інформаційної технології з продажу комп'ютерної техніки

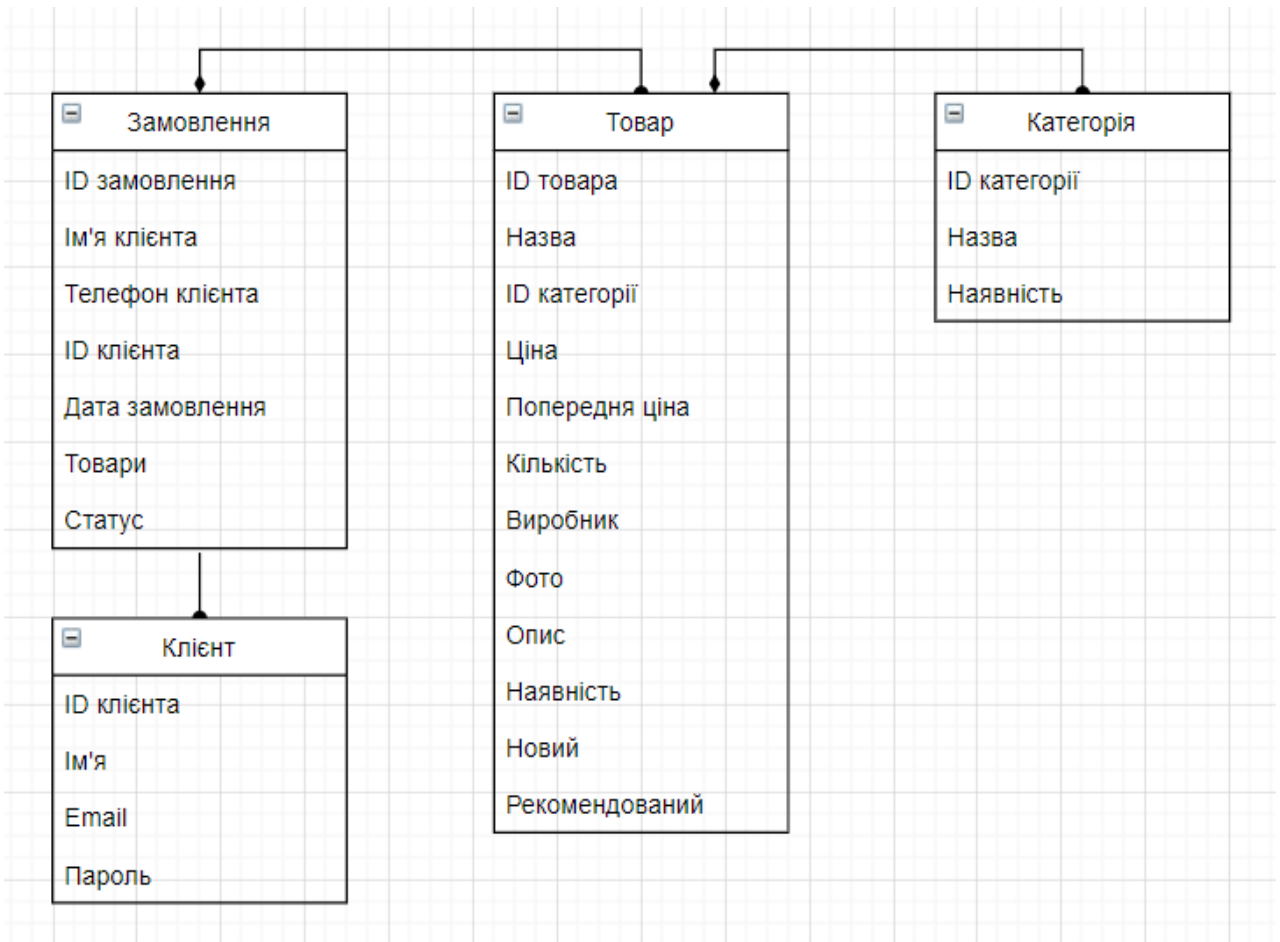


Рисунок В.4 – Узагальнена схема бази даних інформаційної технології з продажу комп'ютерної техніки

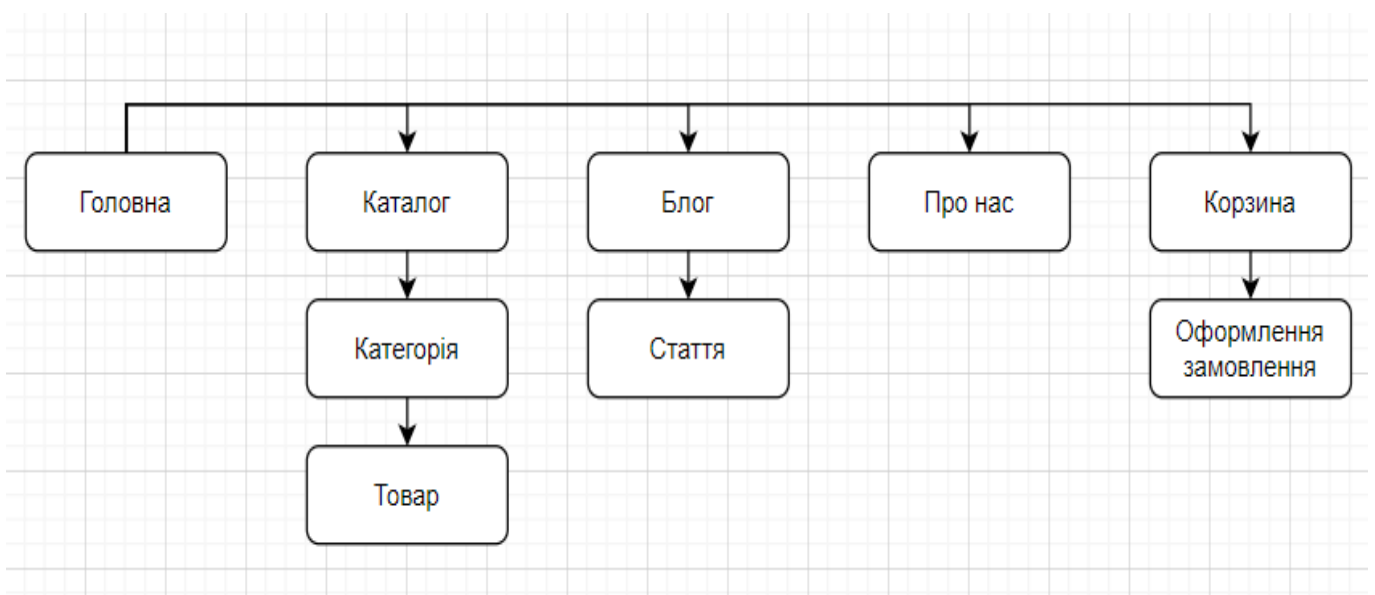


Рисунок В.5 – Структура інтерфейсу інформаційної технології з продажу комп'ютерної техніки



Рисунок В.6 – Інтерфейс головної сторінки веб-ресурсу з продажу комп'ютерної техніки

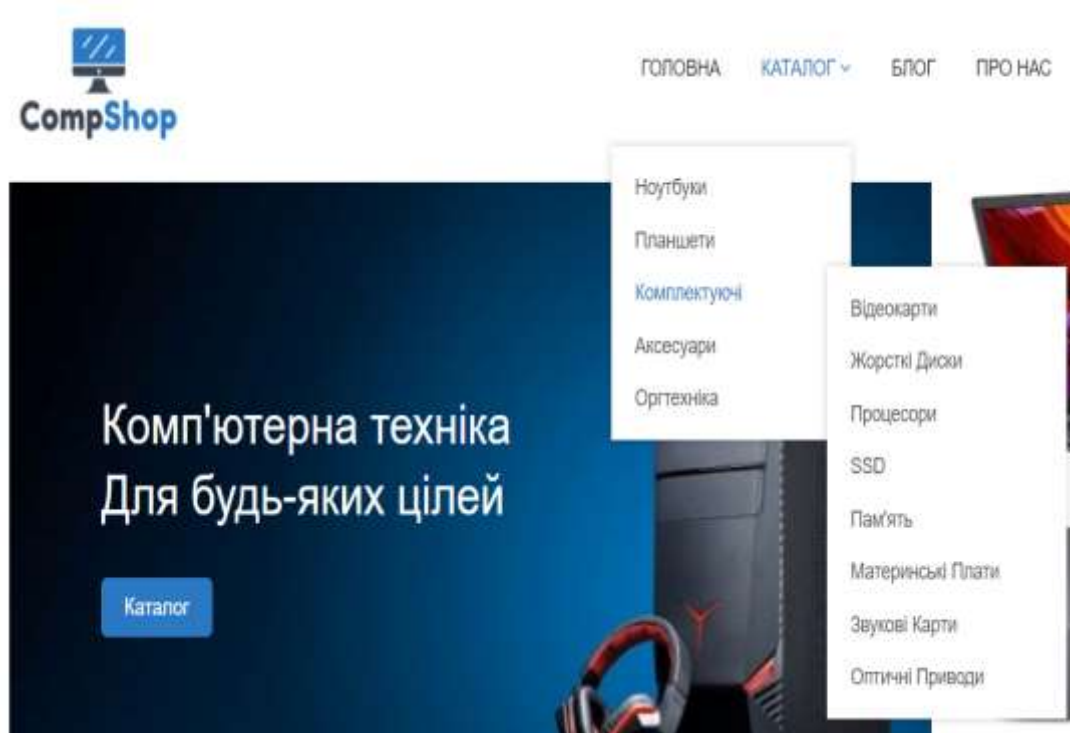


Рисунок В.7 – Графічний інтерфейс веб-ресурсу з продажу комп'ютерної техніки

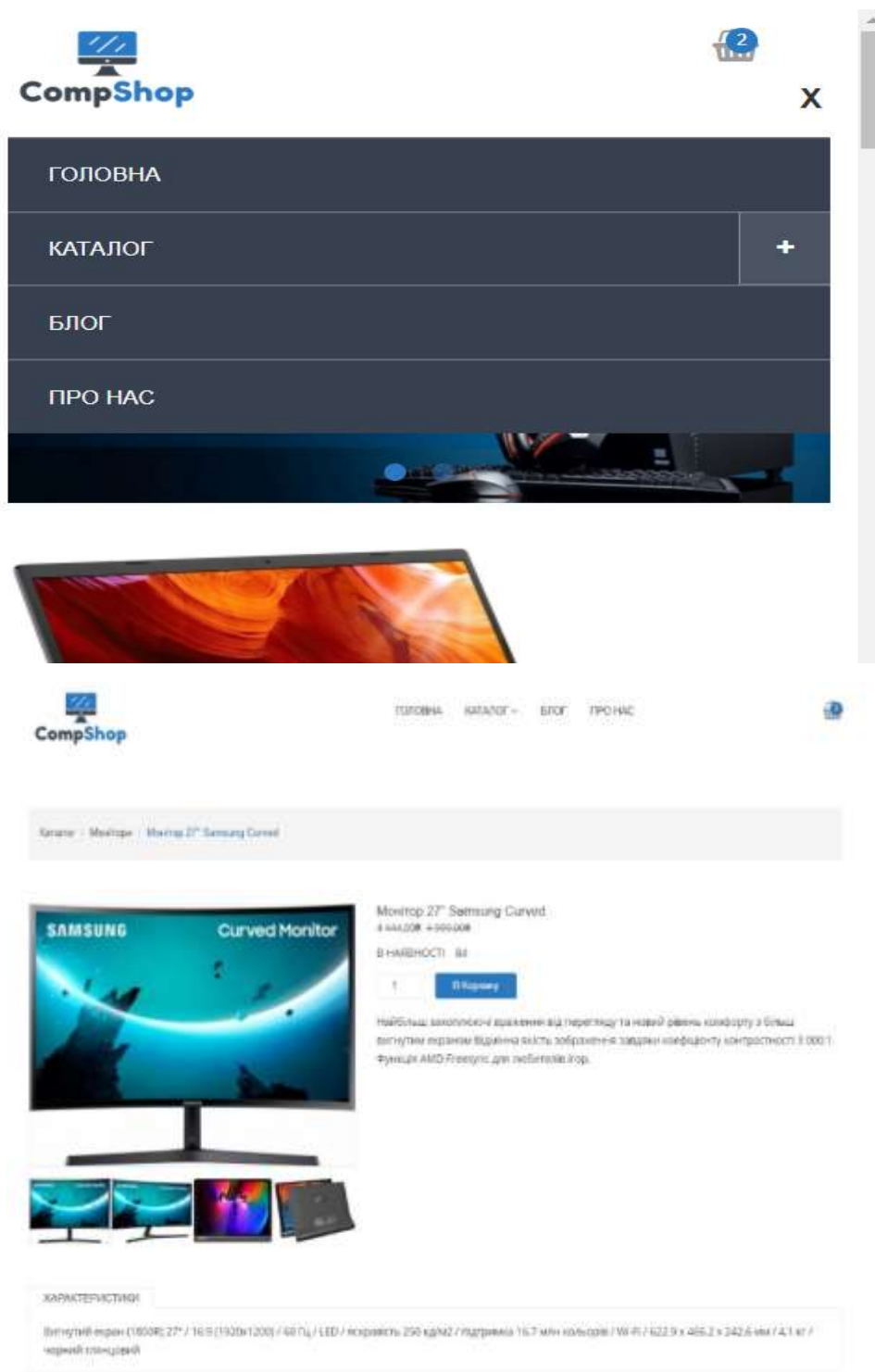


Рисунок В.8 – Графічний інтерфейс веб-ресурсу з продажу комп'ютерної техніки

Головна / Реєстрація




Реєстрація


* Ім'я

* Email

* Пароль

[ЗАРЕЄСТРУВАТИСЬ](#)


 [ГОЛОВНА](#) [КАТАЛОГ](#) [ВХОД](#) [ПРО НАС](#)  



Планшет Lenovo Yoga Tab
 8899 \$ 11000 \$

Екран 10.1" (1920x1200) IPS, MultiTouch / Qualcomm Snapdragon 439 Q / 8 Гб / RAM 4 Гб / 64 Гб вбудованої пам'яті + microSD (до 256 Гб) / екран 60 Гц / Wi-Fi / nano-SIM / 3G / 4G LTE / Bluetooth 4.2 / основна камера 8 Мп, фронтальна - 5 Мп / A-GPS / ГЛОНАСС / Android 9.0 (Pie) / USB Type-C, роз'єм 3.5 мм / батарея 7000 мАч / 570 г / срий

[В ОФОРМЛЕННІ](#)



Монитор 27" Samsung Curved
 8899 \$ 4999 \$

Вигнутий екран (1900R) 27" / 16:9 (1920x1200) / 60 Гц / LED / яскравість 250 кд/м2 / підтримка 16.7 млн кольорів / Wi-Fi / 622.8 x 466.2 x 342.6 мм / 4.1 кг / чорний глянцевий

[В ОФОРМЛЕННІ](#)

1 2 3

Рисунок В.9 – Графічний інтерфейс веб-ресурсу з продажу комп'ютерної техніки

Додаток Г (довідниковий) Інструкція користувача

Після запуску програми необхідно пройти процес реєстрації нового користувача (рис. Г.1).

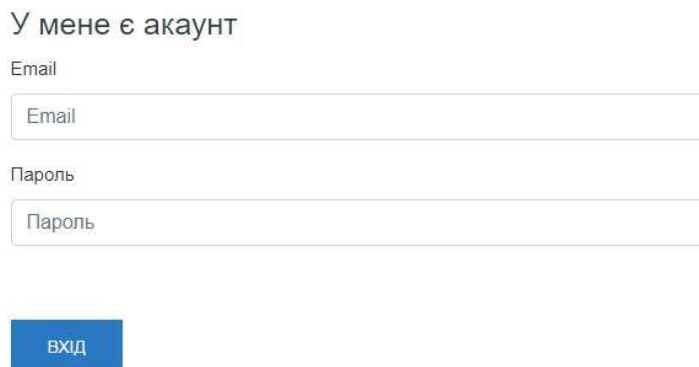
Для успішного проходження реєстрації на ресурсі потрібно правильно заповнити реєстраційні поля: Ім'я (більше трьох символів), Email, пароль (більше восьми символів).



The screenshot shows a web interface for user registration. At the top, there is a navigation bar with the text "Головна" and "Реєстрація". Below this, the title "Реєстрація" is displayed. The form consists of three input fields, each with a label and an asterisk indicating it is required: "Ім'я", "Email", and "Пароль". Below the input fields is a blue button with the text "ЗАРЕЄСТРУВАТИСЬ".

Рисунок Г.1 – Реєстрація користувача

Якщо користувач уже зареєстрований у системі, йому потрібно у формі вказати лише Email та пароль, вказані при реєстрації (рис. Г.2).



The screenshot shows a web interface for user login. The title "У мене є акаунт" is displayed. Below this, there are two input fields: "Email" and "Пароль". Below the input fields is a blue button with the text "ВХІД".

Рисунок Г.2 – Авторизація користувача

В асортименті магазину представлено велику кількість товарів, ви можете переглянути будь-який, який Вам сподобався і додати в кошик. Переглянути товарів можна на сторінках «Каталог» і «Головна». На рис. Г.3 представлена частина асортименту наявних товарів.

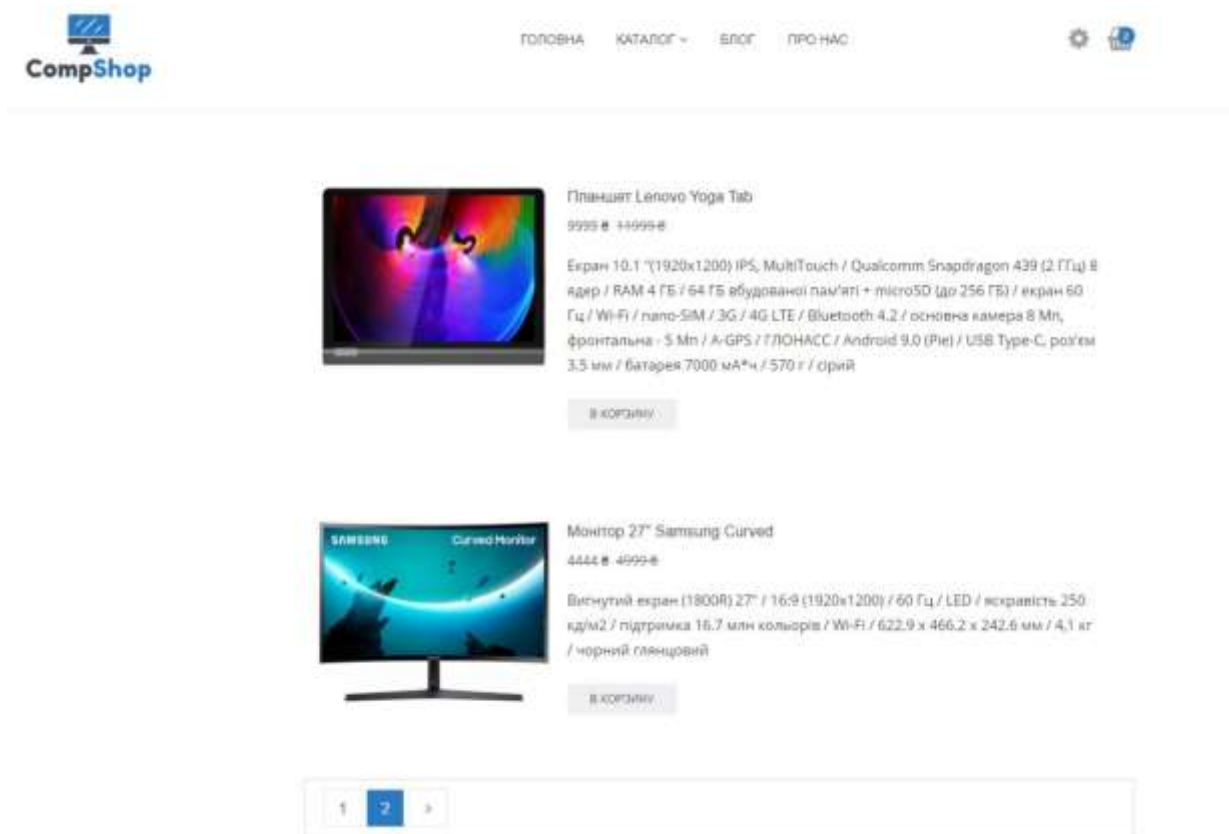


Рисунок Г.3 – Каталог товарів

Список товарів, які додані у кошик можна регулювати за власним бажанням, видаляючи непотрібні та оформлюючи замовлення на ті, які сподобались.

Оформлення замовлення відбувається за декілька кроків.

Якщо користувач увійшов у свій обліковий запис, йому на електронну пошту прийде повідомлення, яке буде містити інформацію про замовлення. В майбутньому продавець зв'яжеться з ним для уточнення даних.

Якщо ж вхід не був здійснений, то клієнту буде запропоновано вказати адресу електронної пошти, по якій з ним можна буде зв'язатись.