

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

### КВАЛІФІКАЦІЙНА РОБОТА

Веб-застосунок для продажу

Назва теми

комп'ютерних периферійних пристроїв

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного  
забезпечення»

Шифр КвРІПЗ.190134.01.10.ПЗ

Виконав студент IV курсу, група ПЗ-19-1

Підпис

Каспруков А. В.

Ініціали, прізвище

Керівник д. ф.-м. н., проф.

Науковий ступінь, звання

Підпис

Бедратюк Л. П.

Ініціали, прізвище

Нормоконтролер канд.тех.наук,доцент

Науковий ступінь, звання

Підпис

Гурман І.В

Ініціали, прізвище

До захисту допускаю:

Завідувач кафедри інженерії  
програмного забезпечення

Підпис

Л. П. Бедратюк

Ініціали, прізвище

5 червня 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій  
Кафедра Інженерії програмного забезпечення  
Рівень вищої освіти Перший (бакалаврський)  
Галузь знань 12 «Інформаційні технології»  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ  
Завідувач кафедри 173  
Л. П. Бедратюк  
02 01 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Каспруков Артем Віталійович

Прізвище, ім'я, по батькові студента

1. Тема кваліфікаційної роботи Веб-застосунок для продажу  
комп'ютерних периферійних пристроїв

Керівник кваліфікаційної роботи Бедратюк Леонід Петрович д.ф.-м.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 5





2. Строк подання студентом роботи на кафедру 01.06.2023 р.

3. Вихідні дані до роботи Матеріали переддипломної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)  
Дослідження предметної області,  
Проектування програмного забезпечення,  
Програмна реалізація,

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)  
Три креслення у форматі А3: блок схема авторизації та редагування  
профілю користувача, функціональна діаграма варіантів використання,  
ER-діаграма бази даних

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Гурман І. В., канд.тех.наук. доцент	3.06.23 	3.06.23 
Антиплагіат	Гурман І. В., канд.тех.наук. доцент	2.06.23 	2.06.23 

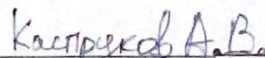
7. Дата видачі завдання « 02 » січня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1 Ознайомлення з тематикою дипломного проєктування, визначення та узгодження індивідуальних тем кваліфікаційних робіт (КвР)	01.12– 31.12.2022	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2023	
3 Проєктування програмного забезпечення	01.02 – 28.02 2023	
4 Програмна реалізація з використанням відповідних засобів розробки	01.03 – 10.04.2023	
5 Тестування програмного забезпечення	11.04 – 30.04.2023	
6 Написання вступу, загальних висновків, оформлення переліку джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог	01.05 – 25.05.2023	
7 Попередній захист КвР	травень 2023	
8 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки.	26.05 – 30.05.2023	
9 Здача КвР на кафедру; підготовка КвР для розміщення у репозитарії ХНУ; підготовка до захисту та захист КвР	з 01.06.2023	

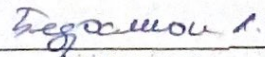
Студент

  
Підпис

  
Ініціали, прізвище

Керівник роботи

  
Підпис

  
Ініціали, прізвище

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Веб-застосунок для продажу комп'ютерних периферійних пристроїв».

Автор роботи: Каспруков А.В.

Керівник роботи: Бедратюк Л.П.

Обсяг – 64 с., 25 рис., 2 таблиці, 3 додатки, 46 джерел.

Графічна частина: 3 креслення.

Веб-застосунок, інтернет-магазин, ASP.NET MVC, C#, SQL SERVER.

Мета кваліфікаційної роботи полягає в розробці концепції та створенні веб-застосунку комп'ютерної периферії. Робота включає в себе вивчення потреб та побажань споживачів, аналіз ринку та конкурентів, вибір технологічної платформи та створення дизайну інтерфейсу магазину. Також в роботі буде розглянуто питання забезпечення безпеки операцій та даних споживачів при здійсненні покупок у веб-застосунку.

Очікуваним результатом кваліфікаційної роботи є розробка та запуск функціонуючого веб-застосунку комп'ютерної периферії, який зможе забезпечити споживачів широким вибором якісних товарів за доступними цінами. Також, магазин буде мати зручний та простий інтерфейс для замовлення та оплати, що забезпечить комфортний та швидкий процес покупок.

1.06.2023

(дата)



(підпис)

## ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.190134.01.10.ПЗ	Пояснювальна записка	64		
2	A4		Завдання на кваліфікаційну роботу	1		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
5	A3	КвРІПЗ.190134.01.10.E8	Блок схема авторизації та редагування профілю користувача	1		
6	A3	КвРІПЗ.190134.01.10.E8	Функціональна діаграма варіантів використання	1		
7	A3	КвРІПЗ.190134.01.10.E8	ER-діаграма бази даних	1		





КвРІПЗ. 190134.01.10.ВД

Змн.	Арк.	№ докум.	Підпис	Дата	Літ.	Арк.	Аркуші
Виконав		Каспруков А.В.		5.06			
Керівник		Бедратюк Л.П.		5.06		1	1
Н. контр.		Гурман І.В.		5.06			
Зав. каф.		Бедратюк Л.П.		5.06			

ХНУ, ІПЗ-19-1

## ЗМІСТ

<b>ВСТУП</b> .....	6
<b>1. Дослідження предметної області та постановка задачі</b> .....	8
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей.....	8
1.2 Аналіз наявного програмно-технічного забезпечення предметної області	12
1.3 Визначення функціональних вимог до програмного забезпечення.....	18
1.4 Постановка задачі.....	20
<b>2. Проектування програмного забезпечення</b> .....	22
2.1 Архітектурне проектування .....	22
2.2 Детальне проектування.....	27
2.3 Проектування інтерфейсу користувача.....	35
2.4 Аналіз та вибір технологій для розробки клієнтської частини веб-застосунку .....	38
2.5 Висновок .....	40
<b>3. Програмна реалізація</b> .....	41
3.1 Розробка бази даних .....	41
3.2 Розробка програмних модулів.....	43
3.3 Керівництво користувача.....	47
3.4 Технічні характеристики веб-додатка.....	52
3.5 Аналіз методів тестування веб-додатка.....	53
3.6 Розробка тестів .....	55
3.7 Висновок .....	57
<b>ВИСНОВКИ</b> .....	59
<b>СПИСОК ДЖЕРЕЛ</b> .....	60
<b>ДОДАТОК А</b> .....	65
<b>ДОДАТОК Б</b> .....	84
<b>ДОДАТОК В</b> .....	96

КВРПЗ. 190134.01.10.ПЗ											
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>							
		<i>Виконав Каспруков А.В.</i>		<i>5.06</i>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;"><i>Літ.</i></td> <td style="width: 10%;"><i>Арк.</i></td> <td style="width: 10%;"><i>Аркушів</i></td> </tr> <tr> <td style="text-align: center;">5</td> <td style="text-align: center;">5</td> <td style="text-align: center;">64</td> </tr> </table>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>	5	5	64
<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>									
5	5	64									
		<i>Керівник Бедратюк Л.П.</i>		<i>5.06</i>							
		<i>Н. контр. Гурман І.В.</i>		<i>5.06</i>							
		<i>Зав. каф. Бедратюк Л.П.</i>		<i>5.06</i>							

## ВСТУП

В сучасному світі, де комп'ютерна техніка та Інтернет зайняли значне місце в житті людини, веб-застосунки стали популярними серед споживачів. Веб-застосунки дозволяють купувати товари не виходячи з дому та швидко зробити замовлення. Розвиток інтернет-технологій та зростання їх використання в сучасному світі привели до значного збільшення популярності онлайн-торгівлі. За останні кілька років кількість веб-застосунків в Україні значно зросла, а обсяги продажів через інтернет постійно збільшуються.

Комп'ютерна периферія є одним з найбільш важливих компонентів сучасних комп'ютерних систем, тому попит на неї постійно зростає. Крім того, зростання кількості роботи з дому, дистанційної освіти та інших змін у способі життя людей вимагає більшої кількості комп'ютерних пристроїв та аксесуарів, що також сприяє зростанню популярності веб-застосунків комп'ютерної периферії.

Отже, створення функціонуючого веб-застосунку комп'ютерної периферії є актуальним завданням, що відповідає потребам ринку та споживачів. Дана кваліфікаційна робота має на меті не тільки розробити концепцію веб-застосунку, але і дослідити його можливості та переваги порівняно з конкурентами, що забезпечить ефективне впровадження проекту на ринок.

Мета кваліфікаційної роботи полягає в розробці концепції та створенні веб-застосунку комп'ютерної периферії. Робота включає в себе вивчення потреб та побажань споживачів, аналіз ринку та конкурентів, вибір технологічної платформи та створення дизайну інтерфейсу магазину. Також в роботі буде розглянуто питання забезпечення безпеки операцій та даних споживачів при здійсненні покупок у веб-застосунку.

Очікуваним результатом кваліфікаційної роботи буде розробка та запуск функціонуючого веб-застосунку комп'ютерної периферії, який зможе забезпечити споживачів широким вибором якісних товарів за доступними

					КВРІПЗ.190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		6

цінами. Також, магазин буде мати зручний та простий інтерфейс для замовлення та оплати, що забезпечить комфортний та швидкий процес покупок.

Дана кваліфікаційна робота має практичне значення для підприємців, які планують розпочати власний бізнес в сфері інтернет-торгівлі. Результати дослідження можуть бути використані для побудови ефективної маркетингової стратегії, а також для розробки зручного та функціонального інтерфейсу веб-застосунку.

Отже, кваліфікаційна робота містить в собі аналіз та розробку веб-застосунку комп'ютерної периферії, яка буде корисною для підприємців, викладачів, студентів та всіх зацікавлених у розвитку інтернет-торгівлі.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		7

## 1. Дослідження предметної області та постановка задачі

### 1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Аналізування предметної області є важливим етапом при розробці будь-якого проекту, включаючи створення у для продажу комп'ютерної периферії. Цей етап допомагає з'ясувати ключові особливості та характеристики даної галузі, що в свою чергу дозволяє розробити та впровадити необхідний функціонал у проекті з максимальною ефективністю.

Комп'ютерна периферія - це набір пристроїв, які призначені для взаємодії з комп'ютером, такі як клавіатури, миші, монітори, друкуючі пристрої, сканери, акустичні системи та інші. Ці пристрої стали невід'ємною складовою комп'ютерної техніки та широко використовуються в різних сферах, таких як робота, навчання та розваги.

Веб-застосунки з комп'ютерною периферією дуже поширені серед користувачів, оскільки вони надають зручну можливість придбати необхідний пристрій без виходу з дому. Крім того, вони дозволяють знайти широкий вибір товарів, які часто буває важко знайти в звичайних магазинах.

Існує кілька типів сайтів для продажу товарів, кожен з яких має свої переваги та недоліки. Ось декілька з них:

– Інтернет-магазин - це веб-сайт, на якому доступний широкий вибір товарів для покупки в Інтернеті. Він має функціонал для пошуку, вибору та замовлення товарів, а також оплати та доставки. Інтернет-магазин може бути спеціалізованим на певному виді товару (наприклад, електроніка, одяг або косметика), або пропонувати різні категорії товарів.

– Маркетплейс - це сайт, на якому кілька продавців можуть продавати свої товари. Він має функціонал для пошуку товарів та продавців, а також оплати та доставки. Маркетплейс може бути спеціалізованим на певному виді товару

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		8

(наприклад, рукоділля або харчові продукти), або пропонувати різні категорії товарів.

– Сайт-візитка - це сайт, що має за мету просування певного бренду або продукту. Він може містити інформацію про товари, але не має функціоналу для їх замовлення або оплати. Сайт-візитка може бути використаний для створення онлайн-присутності бренду та залучення клієнтів до фізичного магазину.

У цій галузі характерними рисами є швидкий технологічний прогрес та постійні зміни в асортименті товарів. За останні роки на ринку з'явилися нові типи периферійних пристроїв, такі як віртуальні реальності, для яких важливо забезпечити сумісність та зручну взаємодію з комп'ютером.

Як і будь-який інший магазин, інтернет-магазин має власну структуру та функціонал, що робить його унікальним у порівнянні з іншими веб-сайтами. Зазвичай, на сайті інтернет-магазину можна знайти основні розділи, такі як головну сторінку, каталог товарів, корзину замовлень, сторінку профілю користувача, сторінку контактів та розділ інформації. Головна сторінка зазвичай містить інформацію про акції та новини, пошукову систему, а також основні категорії товарів.

Основна функція каталогу товарів на сайті інтернет-магазину полягає у тому, щоб надати користувачеві можливість переглянути всі доступні товари та знайти потрібний за різними параметрами, такими як бренд, модель, ціновий діапазон та інші характеристики. Щоб забезпечити зручний пошук, сайт повинен мати ефективну систему фільтрації та пошуку, що дозволить користувачам швидко знайти необхідний товар.

Корзина замовлень є розділом на сайті, де користувач може переглянути свій вибір товарів та оформити замовлення. Сторінка профілю користувача дозволяє зареєстрованим користувачам керувати своїм обліковим записом.

Було прийнято рішення використовувати технологію програмування ASP.NET MVC для розробки веб-застосунку, що спеціалізується на продажу комп'ютерної периферії. Ця технологія дозволяє створювати масштабовані та

					КВРІПЗ.190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		9

стійкі до великих навантажень веб-застосунки. Було обрано програмне середовище Visual Studio 2022 для розробки, яке надає доступ до вбудованих інструментів для швидкої та ефективної розробки програмного забезпечення, що допомагає розв'язувати складні задачі.

Інтернет-магазини мають особливість інтеграції з електронними платіжними системами, які дозволяють клієнтам здійснювати онлайн-оплату замовлень. Для цього використовується платіжна система - PayPal. Важливо забезпечити безпеку платежів та захист від шахраїв, а також надати користувачам можливість вибору найбільш зручного способу оплати.

Інтернет-магазини мають ще одну важливу особливість - систему управління запасами. Ця система дозволяє контролювати кількість товарів на складі та автоматично оновлювати інформацію про наявність товарів на сайті. Таким чином, запобігається ситуації, коли товар вже розпроданий, але все ще доступний для замовлення на сайті.

У цьому розділі було досліджено предметну область веб-застосунків та їх структурні та функціональні особливості. Ми також обрали технологію програмування ASP.NET MVC та програмне середовище Microsoft Visual Studio 2022 для створення веб-застосунку для продажу комп'ютерної периферії.

Окрім цього, для уявлення та узагальнення складних процесів, які потребують обробки багатьох потоків даних, була використана методологія функціонального моделювання та графічного опису процесів IDEF-0. Застосування такої методології дозволяє створювати контекстні діаграми складних процесів. Головна перевага використання цих діаграм - у спрощенні розуміння ієрархічної структури об'єктів, шляхом чіткого визначення зв'язків між складовими системи та вхідними даними.

Для створення візуальних моделей та діаграм було використано безкоштовне онлайн-середовище <https://creately.com/home/>, яке дозволяє створювати високоякісні та ефективні діаграми та моделі.

					КВРІПЗ.190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		10

На зображенні 1.1 наведено контекстну діаграму онлайн-магазину, де потоки показують, звідки надходять дані та основну інформацію, яка виходить. Вхідні дані включають інформацію, що надходить від користувачів, їх реєстраційні дані, запити до системи щодо даних про конкретні товари або загальну інформацію.

Система включає в себе адміністративний функціонал, який дозволяє контролювати розміщення даних на платформі, спілкуватися з клієнтами та обробляти замовлення. Вихідними даними є замовлення на покупку товарів, які залишають користувачі. Цей процес є головною причиною розробки програмного забезпечення.



Рисунок 1.1 – контекстна IDEF0-діаграма роботи програмного забезпечення

Задля детального зображення процесів дії користувача у системі, була використана діаграма декомпозиції першого рівня (рис.1.2). Це рішення, створити діаграму декомпозиції першого рівня, допомогло отримати один із основних сценаріїв використання сайту.

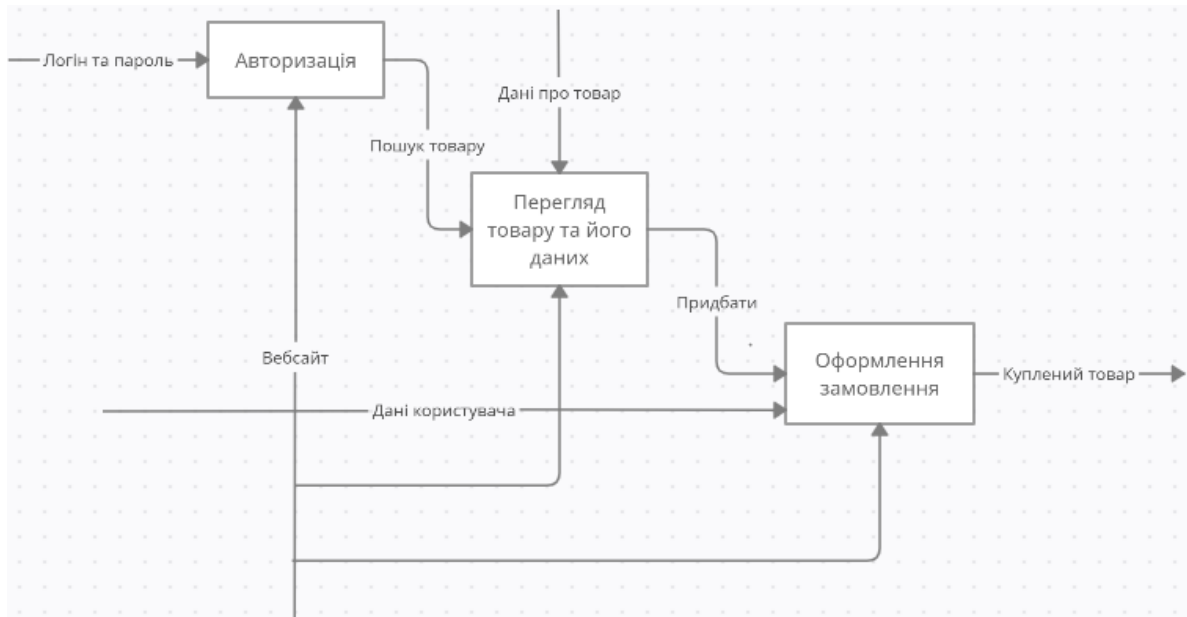


Рисунок 1.2 – IDEF0-діаграма декомпозиції першого рівня

## 1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Для створення веб-застосунку для продажу комп'ютерної периферії, необхідно провести аналіз наявного програмно-технічного забезпечення, що використовується в даній предметній області.

На сучасному ринку існує значна кількість інтернет-магазинів, які спеціалізуються на продажу комп'ютерних пристроїв та периферійних пристроїв, таких як монітори, принтери, клавіатури, миші тощо.

Для того, щоб провести детальний аналіз існуючого програмного та технічного забезпечення в цій галузі, були досліджені різні інтернет-магазини, зокрема лідери цього ринку – Rozetka та Foxtrot. Було проаналізовано їхні сайти, інтерфейси, функціонал, методи оплати та доставки, а також рівень підтримки клієнтів. В результаті отримано важливі висновки щодо найкращих практик у цій галузі, які можуть бути використані для покращення функціоналу та

користувачького досвіду інтернет-магазинів, які спеціалізуються на продажу комп'ютерних пристроїв та периферійних пристроїв.

Один з найпопулярніших інтернет-магазинів у цій області є «Rozetka», який використовує свою власну платформу для електронної комерції – «ROZETKA.UA». Ця платформа є зручною та легкою у використанні, з багатим функціоналом та можливістю персоналізації для користувачів. Компанія була заснована у 2005 році і стала лідером у сфері електронної комерції в Україні завдяки своєму високому рівню обслуговування та величезному асортименту товарів. У магазині клієнти можуть знайти широкий вибір комп'ютерних пристроїв, таких як ноутбуки, комп'ютери, монітори, клавіатури та миші. Крім того, тут пропонуються різноманітні периферійні пристрої, такі як принтери, сканери, маршрутизатори та інші.

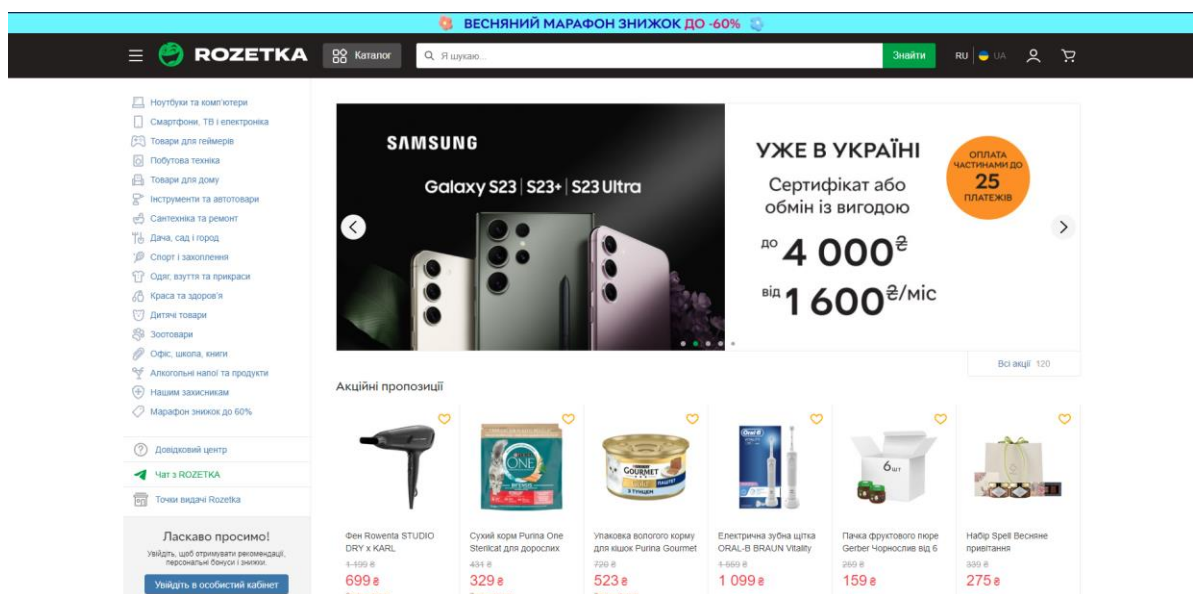


Рисунок 1.3 – сторінка веб-сервісу Rozetka - <https://rozetka.com.ua/ua/>

Сайт має інтуїтивно зрозумілий та простий інтерфейс, що сприяє швидкому та зручному пошуку товарів. Крім того, на сайті доступні фільтри, які дозволяють користувачам швидко знайти потрібний товар за різними критеріями, такими як ціна, бренд, розмір тощо.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		13

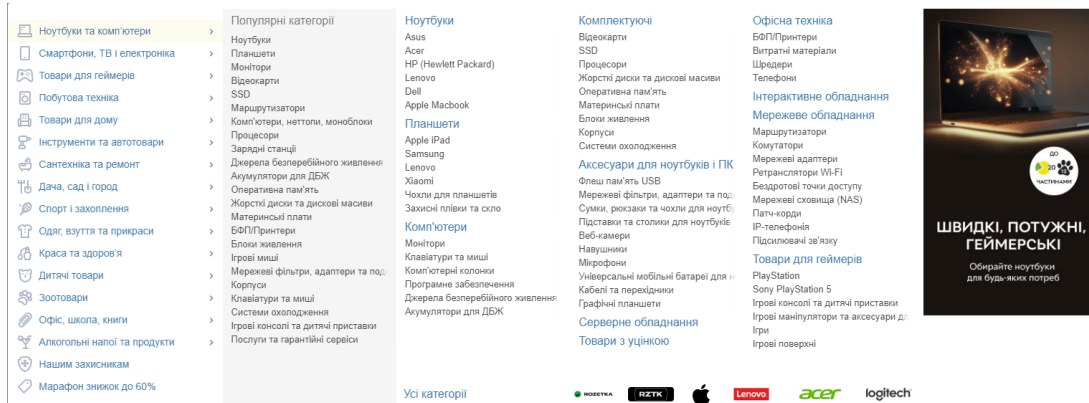


Рисунок 1.4 – каталог веб-сервісу Rozetka - <https://rozetka.com.ua/ua/>

Однією з найважливіших переваг Rozetka є наявність системи відгуків та коментарів, що дозволяє клієнтам отримувати більш детальну інформацію про товари від інших покупців. Ця функція не тільки надає можливість ознайомитись з враженнями та думками інших користувачів, але і допомагає зробити обґрунтований вибір, враховуючи переваги та недоліки товарів.

Така система дозволяє знизити ризик покупки товару низької якості та забезпечує більшу довіру та впевненість в покупцях. Крім того, це ще один спосіб, яким Rozetka намагається забезпечити високий рівень обслуговування та задоволення своїх клієнтів.

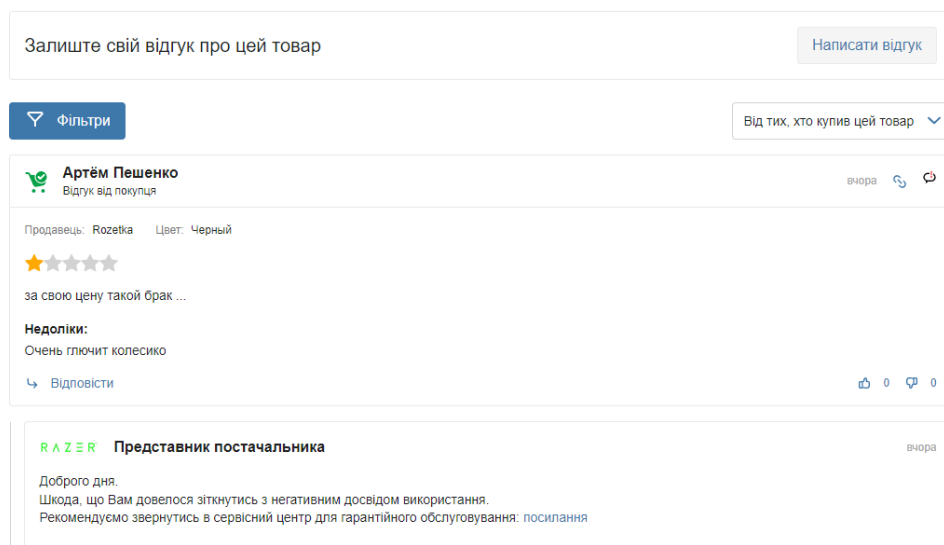


Рисунок 1.5 – відгук товару веб-сервісу Rozetka - <https://rozetka.com.ua/ua/>

На сайті Rozetka також є можливість взаємодії з підтримкою сервісу через чат. Ця функція дозволяє клієнтам отримати оперативну та якісну підтримку щодо питань, пов'язаних з товаром чи оплатою. Якщо клієнт зіштовхнеться з проблемою, він може звернутися до фахівців підтримки через чат і отримати допомогу в режимі реального часу. Це дуже зручно, оскільки клієнт може отримати відповіді на свої запитання швидко та без зайвих зусиль. Також це дозволяє уникнути зайвих розмов з операторами підтримки по телефону або електронній пошті.

Незважаючи на переваги магазину Rozetka, цей веб-сервіс має свої недоліки. Ось декілька недоліків веб-застосунку Rozetka:

- Висока ціна товарів порівняно з конкурентами: на деякі товари ціни в Rozetka можуть бути значно вищими, ніж у інших інтернет-магазинах.
- Доставка товарів може займати більше часу, ніж заявлено: хоча Rozetka зазвичай здійснює доставку протягом кількох днів, в окремих випадках затримки можуть виникнути.
- Проблеми з обробкою повернень та замін товарів: деякі клієнти скаржаться на те, що процес повернення або заміни товару в Rozetka може бути досить складним і займати багато часу.
- Невідповідність товарів опису: іноді товари в Rozetka можуть не відповідати опису на сайті, що може призвести до розчарування у покупців.
- Погана якість обслуговування: деякі клієнти відзначають низьку якість обслуговування в Rozetka, зокрема, неефективність роботи з питань повернення, заміни та гарантій.

Ще одним інтернет-магазином, який заслуговує уваги, є Foxtrot, який використовує платформу Magento. Ця платформа є надійною та стабільною, має велику кількість розширень та можливостей для налаштування.

Foxtrot є одним з провідних інтернет-магазинів в Україні, який спеціалізується на продажу комп'ютерної периферії та інших компонентів,

					КвРІПЗ.190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		15

необхідних для комп'ютера. В асортименті магазину представлені такі товари, як клавіатури, миші, навушники, монітори, принтери, сканери, маршрутизатори, модеми та багато іншого, що дозволяє клієнтам знайти все необхідне для своєї роботи або розваг. Foxtrot співпрацює зі всесвітньо відомими виробниками, такими як HP, Dell, Lenovo, Acer, Samsung та інші, що гарантує високу якість продукції та забезпечує її надійність.

Foxtrot магазин може похвалитися не лише широким асортиментом комп'ютерної периферії, а й різноманітністю вибору. На їх сайті клієнти можуть знайти все необхідне для свого комп'ютера: від мишей та клавіатур до геймпадів та джойстиків для ігор, а також навушників, колонок, мікрофонів, веб-камер і навіть дисків. Незалежно від того, чи ви професійний гравець, чи просто шукаєте надійну і якісну периферію для роботи.

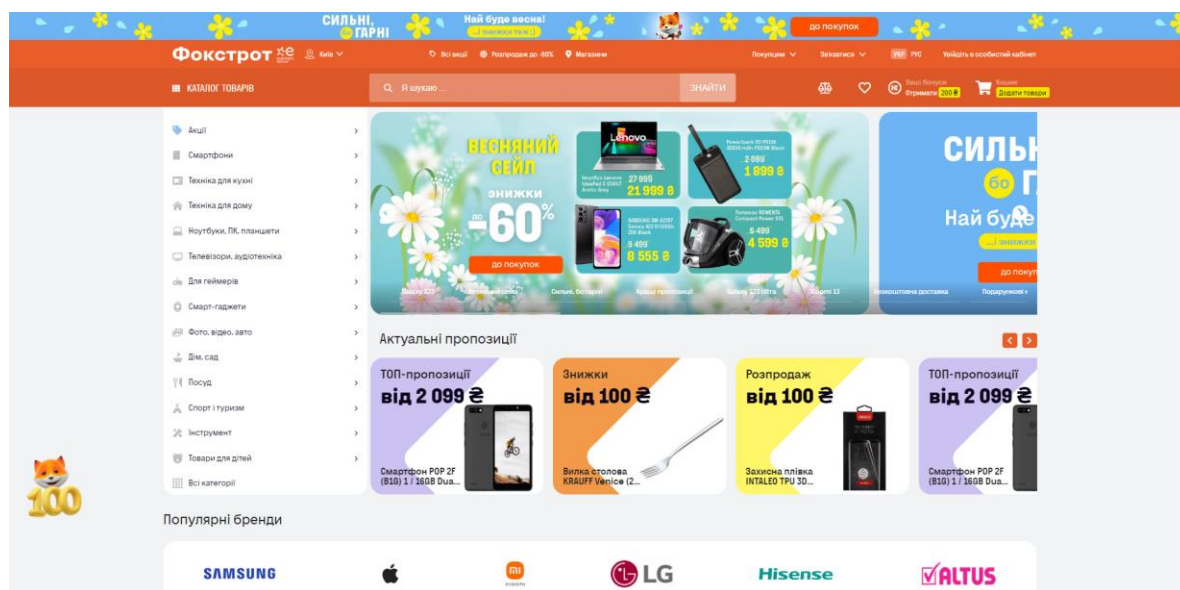


Рисунок 1.6 – головна сторінка магазину Foxtrot - <https://www.foxtrot.com.ua>

В інтернет-магазині Foxtrot представлені відомі бренди комп'ютерної периферії, такі як Logitech, Microsoft, A4Tech, Canyon, Defender, Razer, HyperX та інші. Крім того, в магазині можна знайти як нові моделі, так і б/у товари з гарантією від магазину.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		16

На сайті Foxtrot можна ознайомитися з характеристиками товарів та прочитати відгуки та огляди клієнтів, що допомагає зробити правильний вибір при покупці товарів. Крім того, Foxtrot пропонує безкоштовну доставку до будь-якого з їх магазинів по всій території України, а також різні варіанти оплати, що робить процес покупки зручним та доступним для клієнтів.

Характеристики Мышь HyperX Pulsefire Surge (HX-MC002B)					
Гарантия .....	24 мес.	Тип сенсора .....	оптический	Функции и особенности .....	подсветка
<b>Основные характеристики</b>		Разрешение сенсора .....	16000dpi	Размеры .....	120.24 x 62.85 x 40.7мм
Тип подключения .....	проводное	Количество кнопок .....	6	Вес .....	130г
Проводное подключение .....	USB	Радиус действия/длина кабеля .....	1.8м	▲ Комплектация, внешний вид и характеристики товара могут изменяться производителем без уведомления. Рекомендуем проверять их на сайте производителя перед покупкой	

Рисунок 1.7 – характеристики товару – «Миша HyperX Pulsefire Surge» -

<https://www.foxtrot.com.ua>

Як і будь-який інтернет-магазин, Foxtrot має недоліки, ось декілька з них:

- час доставки може бути тривалим, особливо якщо товар знаходиться на іншому складі магазину.
- Іноді на сайті можуть бути неточності щодо наявності товару, тому є ризик, що замовлення може бути скасовано через відсутність товару на складі.
- Обмін або повернення товару може бути складним і часом затриматися.
- Іноді ціни на товари в магазині можуть бути трохи вищими, ніж в інших магазинах.

Висновуючи з усього вищевказаного, можна стверджувати, що Foxtrot – це провідний та популярний магазин в Україні, який має великий асортимент товарів та послуг у різних категоріях, зокрема, комп’ютерної периферії, електроніки, побутової техніки та іншого. Крім того, Foxtrot надає зручні опції доставки та оплати, а також ставить на перший план задоволення потреб клієнтів. Тому, якщо ви шукаєте надійного партнера для покупки товарів та послуг, Foxtrot може бути гарним вибором.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		17

Один з найпопулярніших програмних продуктів для розробки інтернет-магазинів, які продають комп'ютерну техніку та електроніку, є Magento. Він має багато функціональних можливостей, що дозволяє налаштовувати магазин відповідно до потреб користувача, але використання Magento потребує значних витрат на розробку та підтримку проекту, що може бути неефективним для невеликих підприємств. Інші програмні продукти, такі як OpenCart, PrestaShop та WooCommerce, мають свої переваги та недоліки, але можуть бути вигідними для менших підприємств з обмеженим бюджетом на розробку та підтримку магазину. Існують спеціалізовані програмні продукти для управління складами та продажами, які дозволяють вести облік товарів, приймати та відвантажувати замовлення, вести звіти та аналізувати продажі. Також є різні CRM-системи, які допомагають вести взаємодію з клієнтами та контролювати продажі. Однак, зважаючи на високу вартість деяких з цих програмних продуктів та їх обмежені функціональні можливості, розробка власного веб-застосунку для продажу комп'ютерної техніки може бути більш ефективним та вигідним рішенням для підприємства.

### 1.3 Визначення функціональних вимог до програмного забезпечення

Для вдалого функціонування веб-застосунку комп'ютерної периферії необхідне належне програмне забезпечення, що у подальшому забезпечить належну надійність, ефективність роботи магазину та задоволення потреб клієнтів.

Функції які повинні бути виконанні ПЗ включають:

- Можливість перегляду товарів у магазині, їх характеристики, фотографій та цін.
- Можливість оформлення замовлення.
- Можливість сплати замовлення.

					КвРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		18

– Можливість виконувати вхід користувачу, або реєстрацію, якщо користувач вперше на сайт.

– Можливість додавання товару до кошика.

Програмне забезпечення використовує різні варіанти використання, у кожній ролі користувачів свої можливості – таблиця 1.1.

Таблиця 1.1 – варіанти використання усіх акторів.

Актор	Функції
Гість (користувач без реєстрації на сайті)	Перегляд товарів, їх характеристик, фотографій та ціни; використання категорій для пошуку потрібних товарів; виконання реєстрації для повноцінного використання сайту.
Клієнт	Перегляд товарів, їх характеристик, фотографій та ціни; використання категорій для пошуку потрібних товарів; авторизація; редагування профілю, яка включає змінну пароллю, пошти, ім'я та прізвища;
Адміністратор сайту	Керування контентом сайту (додавання/редагування текстів, фото, тощо.); підтримка та оновлення ПЗ магазину; налаштування безпеки сайту та захист від взломів, DDOS-атак та зловмисного програмного забезпечення.
Банк (система оплати)	Перевірка та обробка платіжних операцій з банківських карток та інших електронних способів оплати; забезпечення безпеки платіжних операцій та попередження шахрайства.

Сайт повинен бути доступний на таких пристроях: телефони, планшети, ноутбуки та комп'ютери. Та з можливістю переходу на сайт з різних браузерів.

У цьому розділі було оглянуто основні функціональні вимоги до програмного забезпечення веб-застосунку для продажу комп'ютерної периферії. Було визначено різноманітні можливості, які має надавати програма для клієнтів, гостей, адміністраторів сайту та системи оплати.

Отже, визначення функціональних вимог до програмного забезпечення є важливим етапом у створенні веб-застосунку для продажу комп'ютерної периферії, що дозволяє забезпечити повноцінне та зручне функціонування магазину для користувачів.

#### 1.4 Постановка задачі

Основна мета цієї кваліфікаційної роботи – розробити веб-застосунок, який задовольнятиме потреби користувачів у зручному, швидкому та безпечному процесі покупки комп'ютерної периферії. Для досягнення цієї мети необхідно вирішити ряд завдань та проблем, пов'язаних з проектуванням, розробкою та тестуванням веб-застосунку.

Для досягнення поставленої мети було сформульовано такі завдання:

1. Аналіз потреб та вимог користувачів: Перед розробкою веб-застосунку необхідно провести детальний аналіз потенційних користувачів, їхніх потреб та вимог до функціональності та зручності веб-застосунку. Це допоможе визначити ключові функції та особливості, які повинні бути реалізовані в програмному продукті.

2. Вибір технологій та інструментів: На основі аналізу ринку та сучасних технологій необхідно вибрати найбільш підходящі технології та інструменти для розробки веб-застосунку. Це включає вибір веб-фреймворка, бази даних, мов програмування та інших компонентів, які забезпечать оптимальну роботу програмного продукту.

					КВРІПЗ.190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		20

3. Проектування архітектури та інтерфейсу: На основі зібраних вимог та вибраних технологій, необхідно розробити архітектуру веб-застосунку, яка буде забезпечувати його ефективну та масштабовану роботу. Також потрібно розробити зручний та інтуїтивно зрозумілий інтерфейс, який дозволить користувачам з легкістю здійснювати покупки та керувати своїми обліковими записами.

4. Реалізація та тестування: На цьому етапі необхідно розробити веб-застосунок, використовуючи обрані технології та архітектуру. Після реалізації провести тестування, щоб переконатися у правильній роботі всіх функцій та відповідності програмного продукту поставленим вимогам.

5. Впровадження та експлуатація: Після успішного завершення розробки та тестування, необхідно впровадити веб-застосунок у роботу та забезпечити його стабільну та безперебійну роботу. Також важливо забезпечити підтримку та обслуговування програмного продукту, щоб користувачі мали доступ до актуальних версій та вирішення можливих проблем.

Виконання поставлених завдань дозволить створити веб-застосунок, який буде задовольняти вимоги користувачів та пропонувати зручну та надійну платформу для придбання комп'ютерної периферії.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		21

## 2. Проектування програмного забезпечення

### 2.1 Архітектурне проектування

У цьому розділі проведений аналіз використання клієнт-серверного типу архітектури в сучасних веб-застосунках з фокусом на кваліфікаційну роботу – «Веб-застосунок для продажу комп'ютерних периферійних пристроїв», який розробляється з використанням мови програмування C# та фреймворку ASP.NET MVC.

Клієнт-серверна архітектура відіграє важливу роль у розробці сучасних веб-додатків, забезпечуючи ефективний обмін даними між клієнтською та серверною частинами. У контексті цього проекту, клієнтська частина буде представлена веб-сторінками, які будуть відображати інтерфейс користувача, а серверна частина буде відповідати за обробку запитів, доступ до бази даних та бізнес-логіку.

ASP.NET MVC є потужним фреймворком для розробки веб-додатків, що надає зручність у реалізації клієнт-серверної архітектури. Модель-представлення-контролер (MVC) є основним шаблоном проектування, що використовується в ASP.NET MVC, дозволяючи розділити логіку додатку на моделі (бізнес-логіка та доступ до даних), представлення (відображення даних користувачу) та контролери (обробка запитів та управління потоком додатку).

Однією з переваг використання клієнт-серверної архітектури в цьому проекті є розділення обов'язків між клієнтом та сервером. Клієнтська частина дозволяє користувачам зручно взаємодіяти з магазином, переглядати товари, розміщувати замовлення та здійснювати оплату. Вона відповідає за візуальне оформлення інтерфейсу, реалізацію функцій взаємодії з користувачем та передачу даних на сервер для подальшої обробки.

Серверна частина, заснована на ASP.NET MVC, забезпечує обробку запитів від клієнта, взаємодію з базою даних, виконання бізнес-логіки та формування

					КвРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		22

відповідей, які надсилаються клієнту. Вона забезпечує безпеку, надійність та ефективність функціонування інтернет-магазину.

Клієнт-серверна архітектура також дозволяє підтримувати розширюваність і масштабованість додатку. Наприклад, можна додати нові функції на стороні сервера, які будуть доступні всім клієнтам без необхідності зміни їхнього інтерфейсу. Також, можна підвищити продуктивність інтернет-магазину шляхом розподілу завдань між різними серверами, наприклад, використовуючи балансувальник навантаження (load balancer).

У сучасних веб-застосунках широко використовується асинхронна взаємодія між клієнтом та сервером з використанням технологій, таких як Ajax. Це дозволяє оновлювати лише окремі частини сторінки без перезавантаження всієї сторінки, що забезпечує швидку та зручну роботу з додатком.

У цій кваліфікаційній роботі, використання клієнт-серверного типу архітектури на базі C# ASP.NET MVC дозволить розробити потужний та функціональний веб-застосунок для продажу комп'ютерних периферійних пристроїв з покращеними можливостями для користувачів. На основі аналізу сучасних веб-застосунків та їхніх використаних клієнт-серверних архітектур, було виявлено деякі переваги та можливості, які можуть бути застосовані у цьому проекті:

1. Розділення логіки: Використання шаблону MVC дозволяє чітко розділити бізнес-логіку, представлення та контролери. Це спрощує розробку, тестування та підтримку проекту, оскільки різні аспекти додатку знаходяться у відокремлених компонентах.

2. Зручність інтерфейсу користувача: Використання клієнт-серверної архітектури дозволяє створити інтуїтивно зрозумілий та зручний інтерфейс для користувачів. Завдяки можливості асинхронної взаємодії, можна оновлювати лише необхідні елементи сторінки, що покращує швидкість та забезпечує більш плавний досвід користувача.

					КвРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		23

3. Масштабованість та продуктивність: Клієнт-серверна архітектура дозволяє легко масштабувати систему, додавати нові функції та розподіляти завдання між різними серверами. Це робить систему більш гнучкою та забезпечує збільшення продуктивності навіть при зростанні обсягу даних та навантаження.

4. Безпека: Використання клієнт-серверної архітектури дозволяє застосовувати різні механізми безпеки, такі як аутентифікація та авторизація, для захисту даних та обмеження доступу до функцій системи.

Узагальнюючи, аналіз використання клієнт-серверного типу архітектури в сучасних веб-застосунках підтверджує його значення та ефективність у розробці інтернет-магазинів та подібних проєктів. Використання мови програмування C# та фреймворку ASP.NET MVC надає багато переваг, зокрема швидкість розробки, стабільність та безпеку.

Далі, на основі проведеного аналізу, буде розроблено та реалізовано клієнт-серверну архітектуру в цій кваліфікаційній роботі – «Веб-застосунок для продажу комп'ютерних периферійних пристроїв». Веб-сторінки, що будуть створені для інтерфейсу користувача, будуть базуватись на мові розмітки HTML, з використанням CSS для стилізації та JavaScript для взаємодії з клієнтом.

На серверній стороні, за допомогою C# та ASP.NET MVC, буде реалізовано обробку запитів, управління базою даних, аутентифікацію та авторизацію користувачів, а також виконання бізнес-логіки. Буде використовувати Entity Framework для роботи з базою даних та розроблю логіку, що дозволить користувачам переглядати товари, розміщувати замовлення, відстежувати їх статус та здійснювати оплату.

Для забезпечення безпеки та захисту даних, буде використаний механізми аутентифікації та авторизації, зокрема використання ролей та прав доступу. Будуть реалізовані заходи для запобігання атак, таким як SQL-ін'єкції, перехоплення даних та інші.

Загалом, аналіз використання клієнт-серверного типу архітектури вказує на його важливість та застосунок у сфері розробки сучасних веб-застосунків,

					КвРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		24

зокрема для веб-застосунків з продажу комп'ютерної периферії. Використання мови програмування C# та фреймворку ASP.NET MVC дозволяє забезпечити ефективну розробку, гнучкість та безпеку системи.

Далі в розділі проводиться аналіз різних підходів до проектування серверної архітектури для цієї кваліфікаційної роботи на тему «Веб-застосунок для продажу комп'ютерних периферійних пристроїв». Враховуючи вимоги до системи та особливості фреймворку ASP.NET MVC, розглянемо декілька підходів, які можуть бути використані для розробки ефективної та масштабованої серверної архітектури.

### 1. Монолітний підхід:

Монолітна архітектура є традиційним підходом до розробки веб-застосунків, де весь функціонал системи розташований в одному додатку. У випадку веб-застосунку для продажу комп'ютерної периферії, цей підхід може бути доцільним, оскільки система має обмежений функціонал і не потребує складної розподіленої архітектури. Програмний код може бути організований у вигляді модулів або шарів, які відповідають за різні частини функціоналу, такі як обробка замовлень, керування продуктами і аутентифікація.

### 2. Сервісно-орієнтований підхід:

У сервісно-орієнтованій архітектурі система розбивається на окремі сервіси, які незалежно виконують свої функції і взаємодіють за допомогою відкритих протоколів, наприклад, REST або SOAP. Кожен сервіс може мати свою власну базу даних та логіку, що дозволяє розподілити завдання та підвищити масштабованість системи. Наприклад, окремі сервіси можуть відповідати за каталог продуктів, обробку замовлень, аутентифікацію та інші функціональні складові системи. Комунікація між сервісами може здійснюватися за допомогою API, що дозволяє їм взаємодіяти і обмінюватися даними.

### 3. Мікросервісна архітектура:

Мікросервісна архітектура є розширеним варіантом сервісно-орієнтованого підходу, де система поділяється на набір невеликих

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		25

індивідуальних мікро сервісів. Кожен мікросервіс має власну базу даних та функціональність і може бути незалежно розгорнутий, масштабований та оновлюваний. Кожен мікросервіс може бути написаний на різних мовах програмування та використовувати власні технології. Цей підхід дозволяє забезпечити гнучкість, розширюваність та незалежність різних компонентів системи.

#### 4. Контейнеризація:

Контейнеризація є технологією, що дозволяє пакувати програмне забезпечення та всі його залежності у виконуваний контейнер, що забезпечує ізольоване та переносним середовище для виконання додатків. Одним з популярних інструментів контейнеризації є Docker. За допомогою контейнерів можна ефективно розгортати та масштабувати окремі компоненти системи, забезпечуючи їх незалежність та розділення ресурсів.

#### 5. Клієнт-серверна архітектура:

Клієнт-серверна архітектура є однією з найпоширеніших та добре зарекомендованих архітектурних підходів у веб-розробці. Застосовуючи цей підхід, ми розділяємо систему на дві основні складові: клієнтську частину, що відповідає за інтерфейс користувача та взаємодію з користувачем, і серверну частину, що забезпечує обробку запитів, доступ до даних та бізнес-логіку.

Переваги клієнт-серверної архітектури включають:

1. Розділення відповідальностей: Цей підхід дозволяє нам чітко розподілити завдання між клієнтом та сервером. Клієнт відповідає за представлення даних та взаємодію з користувачем, тоді як сервер забезпечує обробку даних, бізнес-логіку та збереження інформації.

2. Масштабованість: Клієнт-серверна архітектура сприяє масштабуванню системи, оскільки можна масштабувати окремі компоненти, такі як сервери баз даних або сервери додатків, відповідно до навантаження.

3. Більша безпека: Цей підхід дозволяє нам застосовувати різні рівні безпеки. Серверна частина може забезпечити автентифікацію, авторизацію та

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		26

контроль доступу до даних, що допомагає забезпечити безпеку та конфіденційність інформації.

Більша гнучкість: Клієнт-серверна архітектура дає нам можливість використовувати різні технології та платформи для розробки клієнтської та серверної частин. Це дозволяє нам використовувати оптимальні інструменти та технології для кожної складової системи.

У випадку цього інтернет магазину комп'ютерної периферії, клієнт-серверна архітектура дозволить ефективно керувати взаємодією користувачів з системою, забезпечуючи швидку та надійну обробку запитів, збереження даних та зручний інтерфейс для користувачів.

При виборі підходу до проектування серверної архітектури також варто враховувати свої навички та досвід роботи з певними технологіями. Оскільки розробник програмного проекту володіє навичками програмування на мові C# та фреймворку ASP.NET MVC, монолітний підхід або сервісно-орієнтований підхід може бути більш доречним.

## 2.2 Детальне проектування

У даному розділі буде розглянуте детальне проектування системи « Веб-застосунок для продажу комп'ютерних периферійних пристроїв» з використанням мови програмування C# та фреймворку ASP.NET MVC. Цей розділ має на меті визначити архітектурні складові, моделі даних, функціональні модулі та інші ключові елементи проекту.

### 1. Архітектурна концепція:

У рамках проекту буде використана клієнт-серверна архітектура, де клієнтська частина представлена веб-інтерфейсом для користувачів, а серверна частина відповідає за обробку запитів, доступ до бази даних та бізнес-логіку.

					КвРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		27

Буде використано паттерн MVC (Model-View-Controller), що дозволяє розділити логіку програми на модель даних, представлення та контролери.

## 2. Модель даних:

Основними сутностями системи будуть товари, користувачі, кошик замовлень, замовлення та інші необхідні елементи. Для зберігання даних буде використана реляційна база даних, така як SQL Server або MySQL. Необхідно провести детальний аналіз вимог щодо моделі даних та визначити таблиці, поля, зв'язки та індекси.

## 3. Функціональні модулі:

Система « Веб-застосунок для продажу комп'ютерних периферійних пристроїв» буде містити наступні функціональні модулі:

– Аутентифікація та авторизація користувачів: дозволить користувачам реєструватися, входити в систему та отримувати доступ до персонального кабінету.

– Каталог товарів: відображення списку товарів з можливістю фільтрації, пошуку та сортування.

– Кошик замовлень: додавання товарів до кошика, зміна кількості товар в кошику, видалення товарів з кошика та оформлення замовлення.

– Управління замовленнями: відображення списку замовлень, їх статусів та можливість зміни статусу замовлення.

– Адміністративний розділ: можливість для адміністратора додавати, редагувати та видаляти товари, керувати користувачами, відстежувати статистику продажів та здійснювати інші адміністративні функції.

## 4. Взаємодія з іншими системами:

Система «Веб-застосунок для продажу комп'ютерних периферійних пристроїв» може взаємодіяти з різними зовнішніми системами, наприклад, платіжними шлюзами для оплати замовлень, системою доставки для розрахунку

					КвРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		28

вартості доставки та відстеження посилок, системою електронної пошти для надсилання сповіщень користувачам.

#### 5. Забезпечення безпеки:

Одним із ключових аспектів проектування є забезпечення безпеки системи. Будуть використані заходи для захисту від атак, такі як валідація введених даних, запобігання SQL-ін'єкціям та злому аутентифікації. Також буде реалізована політика доступу до функцій і ресурсів залежно від ролі користувача.

#### 6. Технічні аспекти:

Система буде розроблена з використанням мови програмування C# та фреймворку ASP.NET MVC. Для зберігання даних буде використана реляційна база даних, а для взаємодії з клієнтом – технології AJAX та JavaScript. Буде забезпечена адаптивність та доступність системи на різних пристроях та браузерах.

На основі проведеного детального проектування буде створена діаграма класів, що деталізує взаємодію компонентів системи. На діаграмі будуть відображені класи системи разом з їхніми взаємозв'язками та властивостями. Це дозволить краще розуміти структуру системи та взаємодію її компонентів.

Також будуть розглянуті питання масштабування та оптимізації системи. Зокрема, будуть проаналізовані можливості горизонтального та вертикального масштабування, розглянуті підходи до оптимізації продуктивності та швидкодії системи, включаючи хешування, оптимізацію запитів до бази даних та інші техніки.

Загалом, детальне проектування системи «Веб-застосунок для продажу комп'ютерних периферійних пристроїв» на основі мови програмування C# та фреймворку ASP.NET MVC дозволить визначити архітектурні рішення, створити необхідні діаграми та деталізувати функціональні модулі системи. Це забезпечить якісне та ефективне реалізацію проекту та відповідність його вимогам та цілям.

					КвРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		29

## 2.2.2 Вибір системи керування базами даних

У даному розділі проведений аналіз різних типів баз даних з метою вибору найбільш підходящого варіанту для реалізації системи «Веб-застосунок комп'ютерної периферії». Враховуючи вимоги проекту та особливості розробки на мові програмування C# з використанням фреймворку ASP.NET MVC, були розглянуті такі типи баз даних:

1. Реляційні бази даних:
  - MySQL: відкрита реляційна база даних, яка володіє великим набором функціональних можливостей та широко використовується у веб-розробці.
  - Microsoft SQL Server: комерційна реляційна система керування базами даних, що підтримує багатofункціональну розробку на платформі Microsoft.
2. NoSQL бази даних:
  - MongoDB: документоорієнтована база даних, яка забезпечує гнучку схему та швидкий доступ до даних.
  - Redis: ключ-значення база даних, яка володіє високою продуктивністю та підтримує хешування та сесії.

Після проведеного аналізу та порівняння цих типів баз даних, було прийнято рішення використовувати реляційну базу даних Microsoft SQL Server для реалізації системи « Веб-застосунок комп'ютерної периферії». Це рішення було обґрунтовано такими факторами:

1. Стабільність та надійність: Microsoft SQL Server відома своєю стабільністю та надійністю, що є важливим для операційної діяльності інтернет-магазину.
2. Підтримка мови програмування C#: Microsoft SQL Server має добру інтеграцію з мовою програмування C#, що спрощує розробку та взаємодію з

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		30

базою даних у проекті, зокрема з використанням Entity Framework, що є популярним ORM-фреймворком для роботи з реляційними базами даних у середовищі ASP.NET MVC.

3. Функціональні можливості: Microsoft SQL Server надає широкий набір функціональних можливостей, які можуть бути корисними для інтернет-магазину, таких як транзакційна безпека, підтримка засобів реплікації та масштабування, можливості роботи зі складними запитамми тощо.

4. Екосистема та підтримка: Microsoft SQL Server має розгалужену екосистему, яка включає інструменти розробки, документацію та активну спільноту користувачів. Це забезпечує зручну роботу з базою даних та можливість отримання підтримки у разі потреби.

Таким чином, вибір реляційної бази даних Microsoft SQL Server є обґрунтованим для проекту «Веб-застосунок комп'ютерної периферії» з урахуванням його потреб та особливостей реалізації на мові програмування C# з використанням фреймворку ASP.NET MVC. Обрана база даних надасть потрібний рівень стабільності, функціональності та підтримку для успішної реалізації проекту.

### 2.2.3 Проектування функціональної архітектури

У цьому розділі буде розглянута функціональна архітектура системи «Веб-застосунок комп'ютерної периферії». Функціональна архітектура визначає структуру та організацію функцій, модулів та компонентів системи, а також взаємозв'язки між ними.

#### 1. Модуль «Користувачі»:

– Реєстрація та авторизація: цей модуль відповідає за реєстрацію нових користувачів, авторизацію вже зареєстрованих користувачів та керування їх обліковими записами.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		31

– Користувацький профіль: модуль, що дозволяє користувачам керувати своїми персональними даними, налаштуваннями облікового запису та переглядати історію замовлень.

## 2. Модуль «Каталог товарів»:

– Перегляд товарів: модуль, який надає можливість користувачам переглядати доступний асортимент товарів.

– Фільтрація та сортування: функціональність, яка дозволяє користувачам фільтрувати товари за різними критеріями (ціна, виробник, категорія) та сортувати їх за популярністю або ціною.

– Пошук: модуль, що забезпечує можливість шукати товари за ключовими словами або описом.

## 3. Модуль «Кошик»:

– Додавання та видалення товарів: функція, що дозволяє користувачам додавати товари до кошика та видаляти їх.

– Кількість товарів: модуль, який дозволяє змінювати кількість товарів в кошику.

– Розрахунок вартості: функція, що розраховує загальну вартість товарів у кошику.

## 4. Модуль «Оформлення замовлення»:

– Вибір способу доставки: функція, що надає користувачам можливість обрати бажаний спосіб доставки з доступних варіантів.

– Вибір способу оплати: модуль, який дозволяє користувачам обрати бажаний спосіб оплати, такий як оплата готівкою, переказом або онлайн-платіжною системою.

– Підтвердження замовлення: функція, яка дозволяє користувачам підтвердити своє замовлення та здійснити остаточну оплату.

## 5. Модуль «Адміністрування»:

					КВРІПЗ.190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		32

– Керування товарами: функція, яка надає адміністратору можливість додавати, видаляти та редагувати товари в системі.

– Керування замовленнями: модуль, що дозволяє адміністратору переглядати, обробляти та відстежувати статус замовлень.

6. Модуль «Огляди та рейтинги»:

– Додавання оглядів: функція, яка дозволяє зареєстрованим користувачам залишати огляди про куплені товари.

– Відображення рейтингів: модуль, що відображає рейтинги товарів на основі оцінок користувачів.

Функціональна архітектура системи «Веб-застосунок комп'ютерної периферії» розроблена з урахуванням потреб та вимог, які виникають у процесі створення та управління інтернет-магазином. Вона дозволяє користувачам зручно взаємодіяти з системою, забезпечує ефективну роботу з каталогом товарів, кошиком, оформленням замовлення та іншими функціональними можливостями. Крім того, функціональна архітектура передбачає окремий модуль для адміністрування, що дозволяє адміністратору здійснювати управління товарами, замовленнями та іншими аспектами системи. Це забезпечує можливість ефективного контролю за роботою магазину з боку адміністратора.

Одним із ключових аспектів функціональної архітектури є розділення логіки серверної та клієнтської частин. ASP.NET MVC використовує розділену архітектуру Model-View-Controller (Модель-Вид-Контролер), що дозволяє ефективно розділити логіку додатку між різними компонентами.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		33

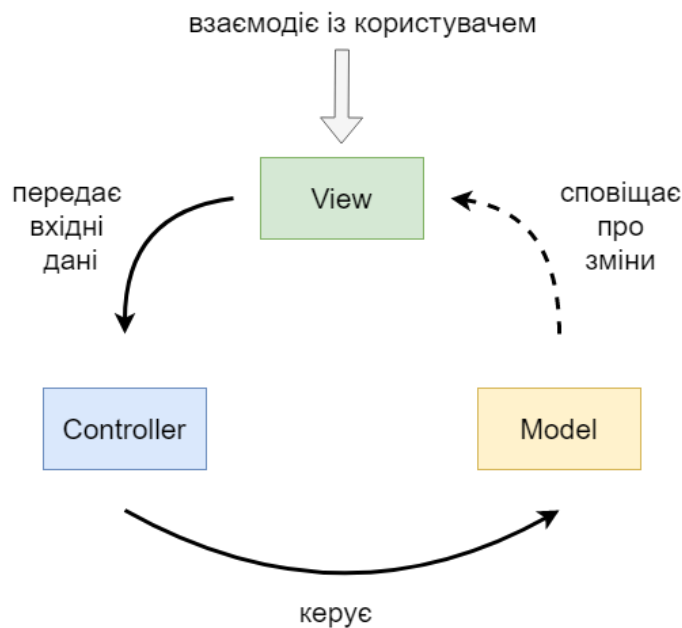


Рисунок 2.2.1 – діаграма взаємодії між компонентами шаблону MVC

Модель (Model) відповідає за обробку даних, включаючи роботу з базою даних, об'єктами доменної моделі та бізнес-логікою. Вона забезпечує доступ до даних та виконання операцій з ними.

Вид (View) відображає інформацію користувачам та обробляє їх взаємодію з системою. Він відповідає за створення та відображення веб-сторінок, інтерфейсів користувача, форм, таблиць тощо. Вид взаємодіє з моделлю та контролером для отримання та передачі даних.

Контролер (Controller) приймає запити від користувача та керує потоком даних між моделлю та видом. Він відповідає за обробку запитів, виконання відповідних дій та відправку відповідей користувачеві.

Завдяки використанню архітектури MVC, розробка та підтримка функціональної архітектури системи «Веб-застосунок для продажу комп'ютерних периферійних пристроїв» стає більш організованою та керованою. Розділення логіки додатку на компоненти дозволяє забезпечити легку модифікацію окремих частин системи без впливу на інші компоненти.

Наприклад, зміни в бізнес-логіці моделі не потребують втручання в інтерфейс користувача або контролери.

Крім того, використання C# та ASP.NET MVC дозволяє використовувати різноманітні фреймворки та бібліотеки для розширення функціональності системи. Наприклад, можна використовувати Entity Framework для роботи з базою даних, Bootstrap для стилізації інтерфейсу користувача, а також різні пакети для роботи з аутентифікацією, авторизацією та іншими аспектами безпеки.

### 2.3 Проектування інтерфейсу користувача

Розділ «Проектування інтерфейсу користувача» є важливою складовою кваліфікаційної роботи на тему «Веб-застосунок для продажу комп'ютерних периферійних пристроїв». У цьому розділі описується процес проектування інтерфейсу, його складові елементи та особливості реалізації за допомогою мови програмування C# та фреймворку ASP.NET MVC.

Один з найважливіших аспектів при проектуванні інтерфейсу користувача є забезпечення зручності та ефективності взаємодії користувача з системою. Інтерфейс повинен бути інтуїтивно зрозумілим, забезпечувати швидкий доступ до необхідної інформації та функціоналу, а також мати привабливий та сучасний дизайн.

Один з підходів до проектування інтерфейсу користувача, який буде використаний, - це концепція «мобільного першого» (mobile-first design). Застосування цієї концепції дозволяє забезпечити адаптивність та оптимальний вигляд інтерфейсу на різних пристроях, включаючи мобільні пристрої, планшети та настільні комп'ютери.

При проектуванні інтерфейсу системи «Веб-застосунок комп'ютерної периферії» було звернута особлива увага на наступні складові:

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		35

1. Навігація: Забезпечення зручної навігації по сайту для користувачів. Меню зрозуміле та легко доступне, з можливістю швидко перейти до основних розділів, таких як каталог товарів, кошик, профіль користувача та інші.

2. Домашня сторінка: Створення привабливої та інформативної домашньої сторінки з переліком акційних товарів і пропозицій, новин та рекомендацій. На домашній сторінці також можна розмістити пошукову панель для швидкого пошуку товарів, а також фільтри для зручного сортування та обмеження результатів.

3. Каталог товарів: Розробка структурованого та зручного каталогу товарів, де користувачі зможуть швидко знайти потрібний товар за категоріями, підкатегоріями, характеристиками та іншими фільтрами. Кожна сторінка товару повинна містити детальну інформацію про товар, його фотографії, ціну, наявність та відгуки користувачів.

4. Кошик покупок: Розробка функціонального та зручного кошика покупок, де користувачі зможуть додавати товари, видаляти їх, змінювати кількість, а також оформляти замовлення. Кошик повинен відображати підсумкову інформацію про товари, загальну суму замовлення та опції доставки та оплати.

5. Авторизація та реєстрація: Розробка системи авторизації та реєстрації користувачів, де вони зможуть створити свій обліковий запис, увійти до системи, переглядати та редагувати свій профіль, а також виконувати різні дії, пов'язані з покупками та замовленнями.

6. Оформлення замовлення: Розробка простого та зрозумілого процесу оформлення замовлення, де користувачі зможуть вказати необхідну інформацію, таку як адреса доставки, спосіб оплати, коментарі до замовлення тощо.

7. Адаптивний дизайн: Забезпечення адаптивного дизайну, щоб інтерфейс користувача належним чином відображався та функціонував на різних пристроях та розмірах екранів, включаючи мобільні пристрої, планшети та

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		36

настільні комп'ютери. Це забезпечить зручне користування системою незалежно від пристрою, який використовує користувач.

8. Валідація даних: Реалізація механізмів валідації даних на клієнтській та серверній сторонах. Це дозволить запобігти введенню некоректних або недійсних даних та забезпечить правильну обробку та збереження інформації.

9. Стилзація інтерфейсу: Використання стилів та CSS для стилізації інтерфейсу користувача, що відповідають сучасним трендам та надають привабливий зовнішній вигляд. Можна використовувати CSS-фреймворки, такі як Bootstrap, для швидкого та простого застосування стилів.

10. Взаємодія з сервером: Реалізація механізмів взаємодії з сервером за допомогою технології AJAX, що дозволяє оновлювати контент без перезавантаження сторінки. Це може бути корисно для додаткової функціональності, такої як додавання товарів до кошика без переходу на нову сторінку.

11. Тестування інтерфейсу: Важлива частина проектування інтерфейсу – це проведення тестування, щоб переконатися, що всі елементи та функціональність працюють правильно та задовольняють вимоги користувачів. Можна використовувати автоматизовані тести, а також залучати користувачів для проведення випробувань та збору фідбеку.

В заключенні розділу «Проектування інтерфейсу користувача» для кваліфікаційної роботи на тему «Веб-застосунок комп'ютерної периферії» на мові програмування C# ASP.NET MVC, буде розроблений зручний, привабливий та функціональний інтерфейс, що відповідає потребам користувачів та стандартам сучасного веб-дизайну. Цей розділ зосереджений на розробці навігації, структури, елементів взаємодії та дизайну інтерфейсу, які сприятимуть позитивному користувацькому досвіду та успішному функціонуванню веб-застосунку з продажу комп'ютерної периферії.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		37

## 2.4 Аналіз та вибір технологій для розробки клієнтської частини веб-застосунку

Розділ «Аналіз та вибір технологій для розробки клієнтської частини веб-застосунку» для кваліфікаційної роботи на тему «Веб-застосунок для продажу комп'ютерних периферійних пристроїв» на мові програмування C# ASP.NET MVC присвячений оцінці різних технологій та інструментів, які можуть бути використані для розробки клієнтської частини веб-застосунку. У цьому розділі буде проведено аналіз доступних технологій та порівняння їх переваг та обмежень, буде зроблений вибір тих, що найбільше відповідають вимогам проекту.

Однією з ключових технологій, яка була розглянута, є HTML5, яка забезпечує потужні можливості для створення розкішного та інтерактивного інтерфейсу. Було досліджено різні можливості HTML5, такі як семантичні теги, форми, аудіо та відео, графіка та анімація, щоб використовувати їх належним чином у проекті.

Крім HTML5, буде розглянуто використання CSS3 для стилізації та оформлення веб-сторінок. CSS3 надає розширені можливості для керування виглядом та макетом елементів, що дозволяє створити привабливий та сучасний дизайн інтерфейсу.

Також, буде проаналізовано JavaScript і його фреймворки для веб-розробки, зокрема Angular та React. Ці фреймворки дозволяють розробляти потужні та інтерактивні клієнтські додатки з високою продуктивністю та швидкодією. Буде проведено порівняння їх функціональних можливостей, буде обрано найбільш слушний фреймворк для потреб проекту та описатиму переваги використання вибраного фреймворку у розробці клієнтської частини веб-застосунку.

Крім того, буде розглянуто використання Ajax (асинхронний JavaScript і XML) для забезпечення безперервної та динамічної взаємодії між клієнтом і

					КвРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		38

сервером. Ajax дозволяє оновлювати вміст сторінки без перезавантаження сторінки, що забезпечує більш плавну та зручну користувацьку взаємодію.

Окрім основних технологій, також буде розглянуто використання різних бібліотек та інструментів, які спрощують розробку клієнтської частини. Наприклад, Bootstrap – популярний фреймворк для розробки адаптивного та зручного для використання на різних пристроях інтерфейсу.

Завершуючи цей розділ, вибір правильних технологій для розробки клієнтської частини є критично важливим для успішної реалізації проекту «Веб-застосунок для продажу комп'ютерних периферійних пристроїв». Подальша робота буде спрямована на реалізацію цих технологій та створення функціональної, естетичної та зручної для користувачів клієнтської частини веб-застосунку. Будуть використані передові технології, такі як HTML5, CSS3, JavaScript, фреймворки Angular або React, а також інші потрібні бібліотеки та інструменти.

Важливо врахувати потреби та очікування користувачів веб-застосунку, забезпечивши їм зручний інтерфейс для швидкого пошуку товарів, оформлення замовлень та спрощеної оплати. Розробка клієнтської частини повинна також бути адаптивною, щоб забезпечити коректне відображення та функціональність на різних пристроях, включаючи комп'ютери, планшети та смартфони.

У підсумку, проектування клієнтської частини веб-застосунку є важливим етапом у реалізації веб-застосунку комп'ютерної периферії. Вибір правильних технологій та інструментів, їхня інтеграція та налагодження, а також розробка естетичного та зручного інтерфейсу допоможуть створити веб-застосунок, який забезпечить задоволення та зручність користувачів у процесі взаємодії з магазином. У наступних розділах будуть розглянуті деталі реалізації цих технологій та розробка функціоналу магазину, що дозволить створити повноцінний та ефективний веб-застосунок.

## 2.5 Висновок

У другому розділі кваліфікаційної роботи було проведено аналіз та проектування архітектури та інтерфейсу веб-застосунку для продажу комп'ютерної периферії. Було розглянуто використання клієнт-серверного типу архітектури в сучасних веб-застосунках, а також аналізовано підходи до проектування серверної архітектури.

У розділі "Архітектурне проектування" було виявлено, що клієнт-серверна архітектура є ефективним та широко використовуваним підходом у сучасних веб-застосунках. Цей тип архітектури дозволяє розділити функціональність між клієнтською та серверною частинами, забезпечуючи більшу масштабованість та надійність системи.

У розділі "Детальне проектування" було розроблено детальну архітектуру веб-застосунку, включаючи складові компоненти, модулі та зв'язки між ними. Було приділено увагу розподілу функціональності та забезпеченню ефективного взаємодії між компонентами системи.

Також було проведено проектування інтерфейсу користувача, де було визначено вигляд та функціональність веб-застосунку. Розроблений інтерфейс враховує потреби користувачів та найкращі практики дизайну інтерфейсів.

У розділі "Аналіз та вибір технологій для розробки клієнтської частини веб-застосунку" були проаналізовані різні технології, що можуть бути використані для розробки клієнтської частини веб-застосунку. Після уважного вивчення було зроблено вибір оптимальних технологій, які забезпечать зручний та ефективний інтерфейс для користувачів.

В результаті виконання другого розділу було проведено детальний аналіз та проектування архітектури та інтерфейсу веб-застосунку. Результати цього розділу є важливою основою для подальшої розробки та реалізації веб-застосунку у наступних етапах кваліфікаційної роботи.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		40

### 3. Програмна реалізація

#### 3.1 Розробка бази даних

Розділ «Розробка бази даних» є ключовим етапом у створенні веб-застосунку для продажу комп'ютерної периферії. Для ефективної роботи та забезпечення потрібного функціоналу магазину, необхідно ретельно спроектувати структуру та взаємозв'язки таблиць у базі даних.

Під час розробки бази даних для веб-застосунку, буде використовуватись мова програмування C# та фреймворк ASP.NET MVC, що надає потужні інструменти для роботи з базами даних. Одним з основних завдань полягає у виборі підходящої системи управління базами даних (СУБД), яка задовольнятиме потреби магазину з точки зору швидкості, масштабованості, надійності та зручності роботи.

Однією з відомих СУБД, яку можна використовувати в контексті цього проекту, є Microsoft SQL Server. Вона надає широкі можливості для створення та управління базами даних, підтримує мову запитів SQL і інтегрується з фреймворком ASP.NET MVC.

Microsoft SQL Server є однією з найпопулярніших та потужних реляційних баз даних, розроблених компанією Microsoft. Він надає широкий спектр функцій та можливостей для зберігання, керування та обробки даних. SQL Server є частиною сімейства продуктів Microsoft Data Platform, яке включає інструменти для аналізу, інтеграції та візуалізації даних.

Основні сутності бази даних цього веб-застосунку включатимуть інформацію про товари, клієнтів, замовлення та інші супутні дані. Наприклад, для таблиці «Товари» можна використовувати поля, які включатимуть унікальний номер товару, назву, опис, ціну та інші характеристики (рис. 3.1.1).

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		41

dbo.tblProducts [Design] ✕				
		Update	Script File: dbo.tblProducts.sql	
	Name	Data Type	Allow Nulls	Default
PK	Id	int	<input type="checkbox"/>	
	Name	nvarchar(50)	<input checked="" type="checkbox"/>	
	Slug	nvarchar(50)	<input checked="" type="checkbox"/>	
	Description	nvarchar(MAX)	<input checked="" type="checkbox"/>	
	Price	numeric(18,2)	<input checked="" type="checkbox"/>	
	CategoryName	nvarchar(50)	<input checked="" type="checkbox"/>	
	CategoryId	int	<input checked="" type="checkbox"/>	
	ImageName	nvarchar(50)	<input checked="" type="checkbox"/>	
			<input type="checkbox"/>	

Рисунок 3.1.1. – таблиця товарів в БД

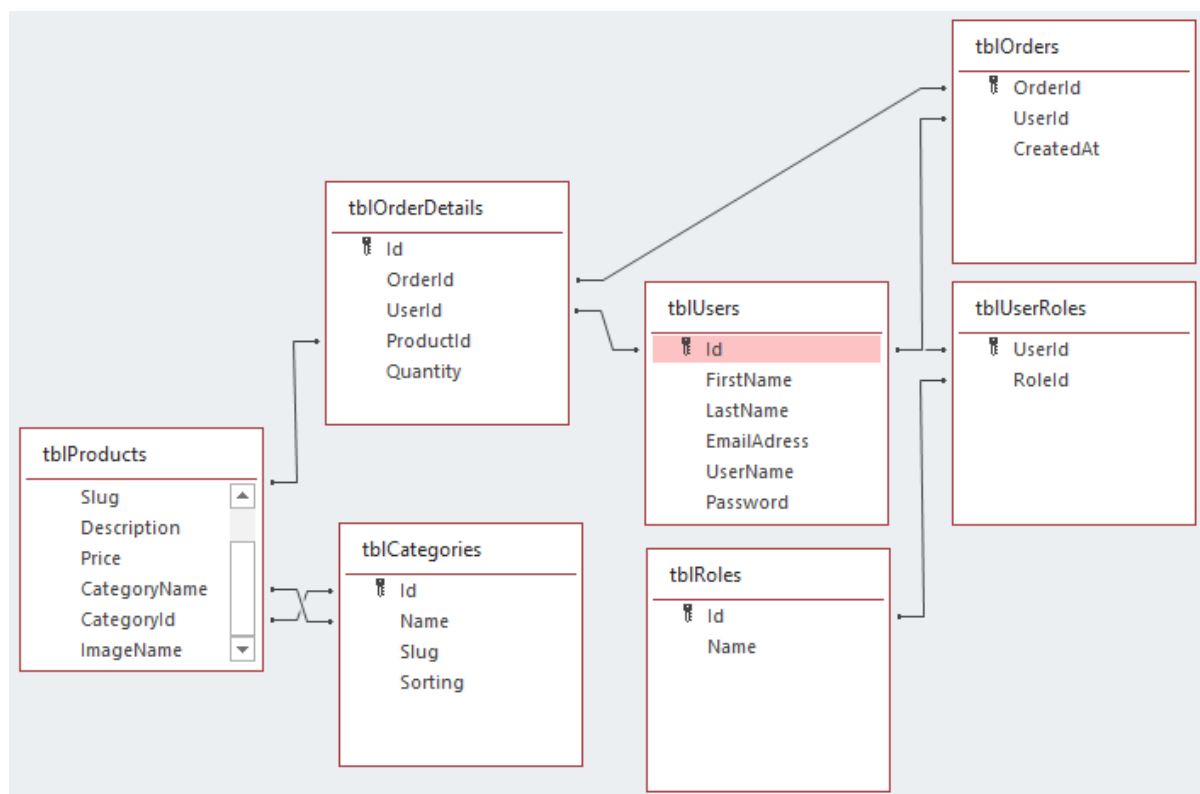


Рисунок 3.1.2. – зв'язки між таблицями БД

У підсумку, розробка бази даних є необхідною складовою процесу створення веб-застосунку для продажу комп'ютерної периферії. Вибір відповідної СУБД, такої як Microsoft SQL Server, забезпечить нам потужність, надійність та зручність у роботі з базою даних.

### 3.2 Розробка програмних модулів

Цей розділ зосереджений на розробці програмних модулів для веб-застосунку з продажу комп'ютерної периферії. Використовуючи мову програмування C# та фреймворк ASP.NET MVC, це надає змогу створити потужні та функціональні модулі, які забезпечать коректну роботу та зручний інтерфейс для користувачів.

Один з основних модулів, який потрібно розробити, це модуль авторизації та реєстрації, який забезпечить безпеку та обмеження доступу до деяких функціональних можливостей. Користувачі зможуть створювати облікові записи, увійти до свого профілю, змінювати дані та керувати своїми особистими налаштуваннями.

```
// GET: Account/Login
[HttpGet]

0 references
public ActionResult Login()
{
    string userName = User.Identity.Name;

    if (!string.IsNullOrEmpty(userName))
        return RedirectToAction("user-profile");

    return View();
}
```

Рисунок 3.2.1 – реалізація авторизації користувача.

```

// POST: Account/Login
[HttpPost]
0 references
public ActionResult Login(LoginUserVM model)
{
    // Перевірка моделі на валідність
    if (!ModelState.IsValid)
        return View(model);

    // Перевірка користувача на валідність
    bool isValid = false;

    using (Db db = new Db())
    {
        if (db.Users.Any(x => x.Username.Equals(model.Username) && x.Password.Equals(model.Password)))
            isValid = true;

        if (!isValid)
        {
            ModelState.AddModelError("", "Невірний логін або пароль.");
            return View(model);
        }
        else
        {
            FormsAuthentication.SetAuthCookie(model.Username, model.RememberMe);
            return Redirect(FormsAuthentication.GetRedirectUrl(model.Username, model.RememberMe));
        }
    }
}

```

Рисунок 3.2.2 – реалізація авторизації користувача.

```

// GET: /account/logout
0 references
public ActionResult Logout()
{
    FormsAuthentication.SignOut();
    return RedirectToAction("Login");
}

```

Рисунок 3.2.3 – реалізація виходу з аккаунту користувача.

Інший важливий модуль – модуль управління товарами. Цей модуль дозволить адміністратору додавати нові товари, оновлювати існуючі, встановлювати ціни. Крім того, користувачі зможуть переглядати доступні товари, шукати за категоріями, додавати їх до кошика та здійснювати покупки. Реалізація додавання товару до категорій зображений на рисунках 3.2.4-3.2.5.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		44

```

// POST: Admin/Shop/AddProduct
[HttpPost]
0 references
public ActionResult AddProduct(ProductVM model, HttpPostedFileBase file)
{
    if (!ModelState.IsValid)
    {
        using (Db db = new Db())
        {
            model.Categories = new SelectList(db.Categories.ToList(), "Id", "Name");
            return View(model);
        }
    }

    using (Db db = new Db())
    {
        if (db.Products.Any(x => x.Name == model.Name))
        {
            model.Categories = new SelectList(db.Categories.ToList(), "Id", "Name");
            ModelState.AddModelError("", "Ця назва вже використовується!");
            return View(model);
        }
    }
}

```

Рисунок 3.2.4 – реалізація додавання товару до категорії.

```

int id;

using (Db db = new Db())
{
    ProductDTO product = new ProductDTO();

    product.Name = model.Name;
    product.Slug = model.Name.Replace(" ", "-").ToLower();
    product.Description = model.Description;
    product.Price = model.Price;
    product.CategoryId = model.CategoryId;

    CategoryDTO catDTO = db.Categories.FirstOrDefault(x => x.Id == model.CategoryId);
    product.CategoryName = catDTO.Name;

    db.Products.Add(product);
    db.SaveChanges();

    id = product.Id;
}

TempData["SM"] = "Ви додали продукт!";



return RedirectToAction("AddProduct");
}

```

Рисунок 3.2.5 – реалізація додавання товару до категорії.

Також для авторизованого користувача було розроблено модуль особистого кабінету користувача. Цей модуль є доволі важливий для веб-

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		45

застосунку, оскільки він надає користувачу можливість редагувати свій профіль, а саме, змінювати ім'я користувача, прізвище, електронну адресу, логін та пароль.(рис 3.2.6 – 3.2.8)

```
// GET: /account/user-profile
[HttpGet]
[ActionName("user-profile")]
0 references
public ActionResult UserProfile()
{
    string userName = User.Identity.Name;

    UserProfileVM model;

    using (Db db = new Db())
    {
        UserDTO dto = db.Users.FirstOrDefault(x => x.Username == userName);

        model = new UserProfileVM(dto);
    }
    return View("UserProfile", model);
}
```

Рисунок 3.2.6 – реалізація модуля редагування профілю користувача.

```
// POST: /account/user-profile
[HttpPost]
[ActionName("user-profile")]
0 references
public ActionResult UserProfile(UserProfileVM model)
{
    bool userNameIsChanged = false;

    if (!ModelState.IsValid)
    {
        return View("UserProfile", model);
    }

    if (!string.IsNullOrEmpty(model.Password))
    {
        if (!model.Password.Equals(model.ConfirmPassword))
        {
            ModelState.AddModelError("", "Пароль не підходить.");
            return View("UserProfile", model);
        }
    }
}
```

Рисунок 3.2.7 – реалізація модуля редагування профілю користувача.

```

using (Db db = new Db())
{
    string userName = User.Identity.Name;

    if (userName != model.Username)
    {
        userName = model.Username;
        userNameIsChanged = true;
    }

    if (db.Users.Where(x => x.Id != model.Id).Any(x => x.Username == userName))
    {
        ModelState.AddModelError("", $"Username {model.Username} не підходить, вгадайте щось інше.");
        model.Username = "";
        return View("UserProfile", model);
    }

    UserDTO dto = db.Users.Find(model.Id);

    dto.FirstName = model.FirstName;
    dto.LastName = model.LastName;
    dto.EmailAddress = model.EmailAddress;
    dto.Username = model.Username;

    if (!string.IsNullOrEmpty(model.Password))
    {
        dto.Password = model.Password;
    }

    db.SaveChanges();

    TempData["SM"] = "Ви змінили ваші данні!";

    if (!userNameIsChanged)
        return View("UserProfile", model);
    else
        return RedirectToAction("Logout");
}

```

Рисунок 3.2.8 – реалізація модуля редагування профілю користувача.

### 3.3 Керівництво користувача

У цьому розділі здійснений покроковий опис користування веб-додатком для різних категорій користувачів: гість (не є зареєстрованим користувачем), клієнт (є зареєстрованим користувачем), адміністратор.

Користувач який має роль «гість», має можливість переглядати товари та зареєструватись. Сторінка реєстрації для таких користувачів зображена на рисунку 3.3.1.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		47

© 2023 – KasperShop

Рисунок 3.3.1 – сторінка реєстрації

Для користувача який вже є зареєстрованим, веб-додаток пропонує широкий спектр функціоналу, цей користувач має можливість додавати товари до кошику, та сплачувати товари які він обрав на сторінці кошика. Також зареєстрований користувач може змінювати свої дані в його профілі облікового запису(рис. 3.3.2).

© 2023 – KasperShop

Рисунок 3.3.2 – сторінка редагування облікового запису

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		48

Користувач, незалежно чи зареєстрований він, чи ні, має можливість переглядати товари з потрібної йому категорії, і детальну інформацію про кожний із-товарів, це можна зробити за допомогою кнопки «Детальніше» поряд з будь-яким товаром.

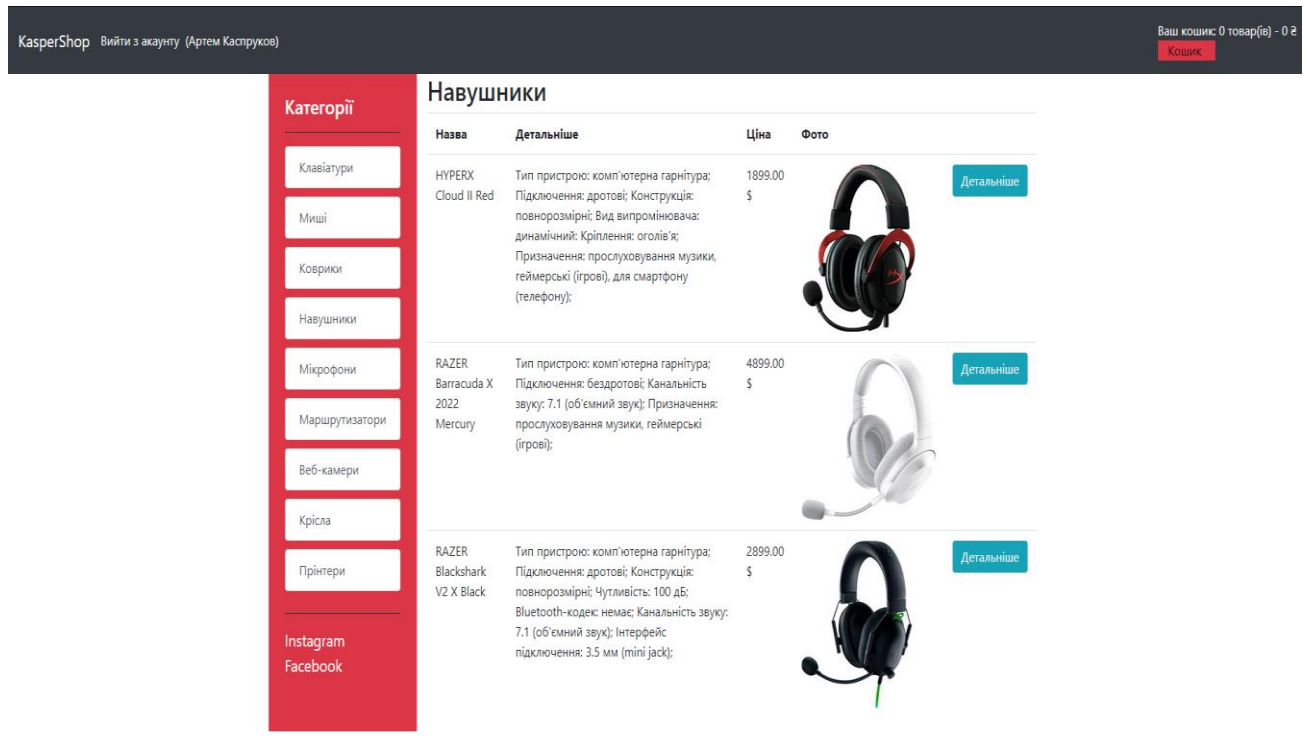


Рисунок 3.3.3 – сторінка категорії «Наушники»

У веб-застосунку, надається зручна функція для користувачів – перегляд та управління їхнім кошиком з обраними товарами. Якщо користувач вже визначився з товаром або просто хоче переглянути товари, які він додав до свого кошика, він може легко здійснити цю дію.

Для зручності користувача, на кожній сторінці веб-застосунку у правому верхньому куті розміщена кнопка з написом «Кошик». Просто натиснувши на цю кнопку, користувач відкриває сторінку зі своїм кошиком, де він може бачити список обраних товарів для придбання.

На сторінці кошика користувачу надається можливість ознайомитися з усім списком товарів, які він додав до свого кошика. Він може бачити назву,

зображення, ціну та іншу важливу інформацію про кожен товар. Це дозволяє користувачу зробити перевірку та впевнитися, що вибрані товари відповідають його потребам.

Крім того, користувач може здійснювати дії з товарами безпосередньо на сторінці кошика. Наприклад, якщо користувач бажає видалити товар зі свого кошика, він може легко знайти відповідну опцію та натиснути на кнопку видалення товару. Також, якщо користувач хоче змінити кількість одного з товарів, він може використати спеціальні кнопки для збільшення або зменшення кількості товару. Вигляд сторінки-кошика можна побачити на рисунку 3.3.4.

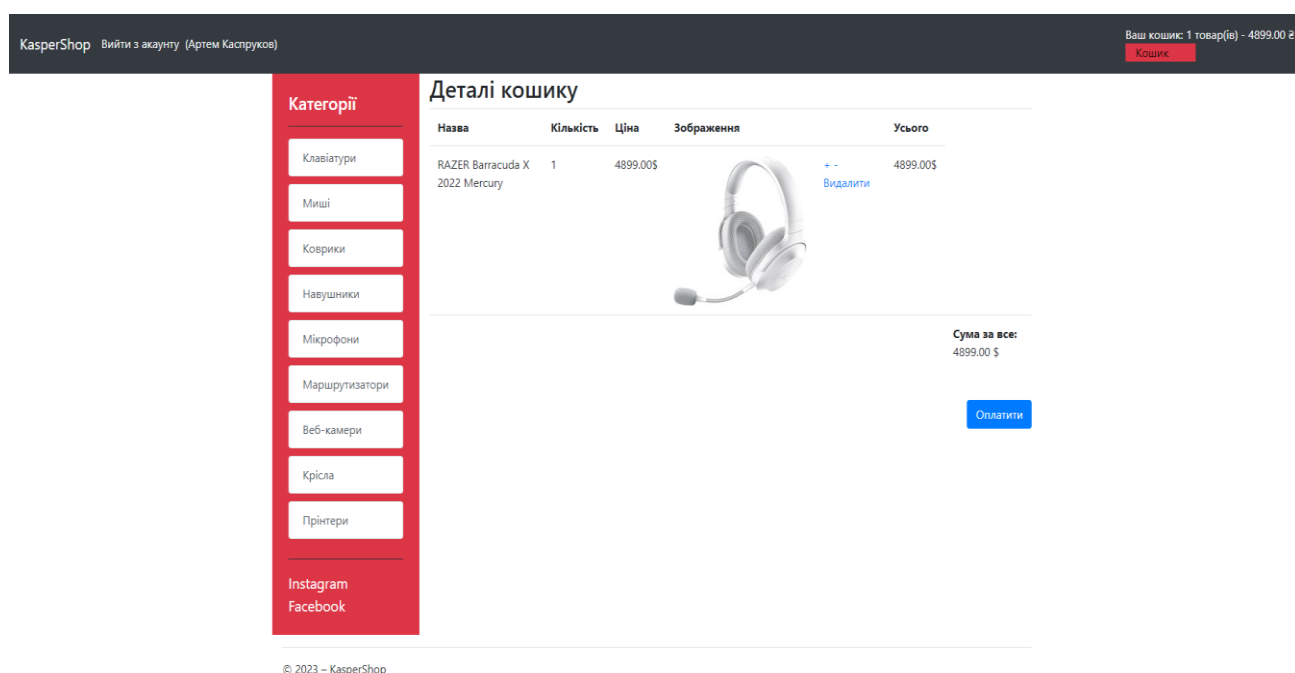


Рисунок 3.3.4 – сторінка-кошик користувача

Адміністратор веб-додатку має можливість додавати, видаляти або змінювати категорії товарів та самі товари, змінювати інформацію про товар, також має можливість додати новий товар у потрібну йому категорію, та видаляти їх. На рисунку 3.3.5 зображена сторінка редагування категорій веб-додатку, а на рисунку 3.3.6 зображена сторінка додавання нового товару.

### Категорії

Назва категорії

[Додати нову категорію](#)

Назва	
Клавіатури	<a href="#">Видалити</a>
Миші	<a href="#">Видалити</a>
Коврики	<a href="#">Видалити</a>
Наушники	<a href="#">Видалити</a>
Мікрофони	<a href="#">Видалити</a>
Маршрутизатори	<a href="#">Видалити</a>
Веб-камери	<a href="#">Видалити</a>
Крісла	<a href="#">Видалити</a>
Прінтери	<a href="#">Видалити</a>

© 2023 – EitLift

Рисунок 3.3.5 – сторінка редагування категорій

### Додати товар

Назва

Детальна інформація

Ціна

Категорія

Фото продукту

[Вибрати файл](#) HYPERX C...ackRed.jpg



[Додати товар](#)

[Повернутись](#)

Рисунок 3.3.6 – сторінка додавання нового товару

Зм.	Арк	№ докум.	Підпис	Дата

КВРІПЗ. 190134.01.10.ПЗ

Арк.

51

На рисунку 3.3.7 можна побачити список товарів які додав адміністратор в каталог магазину, він може відсортувати їх за категоріями, змінювати їх, видаляти, або додати новий товар.

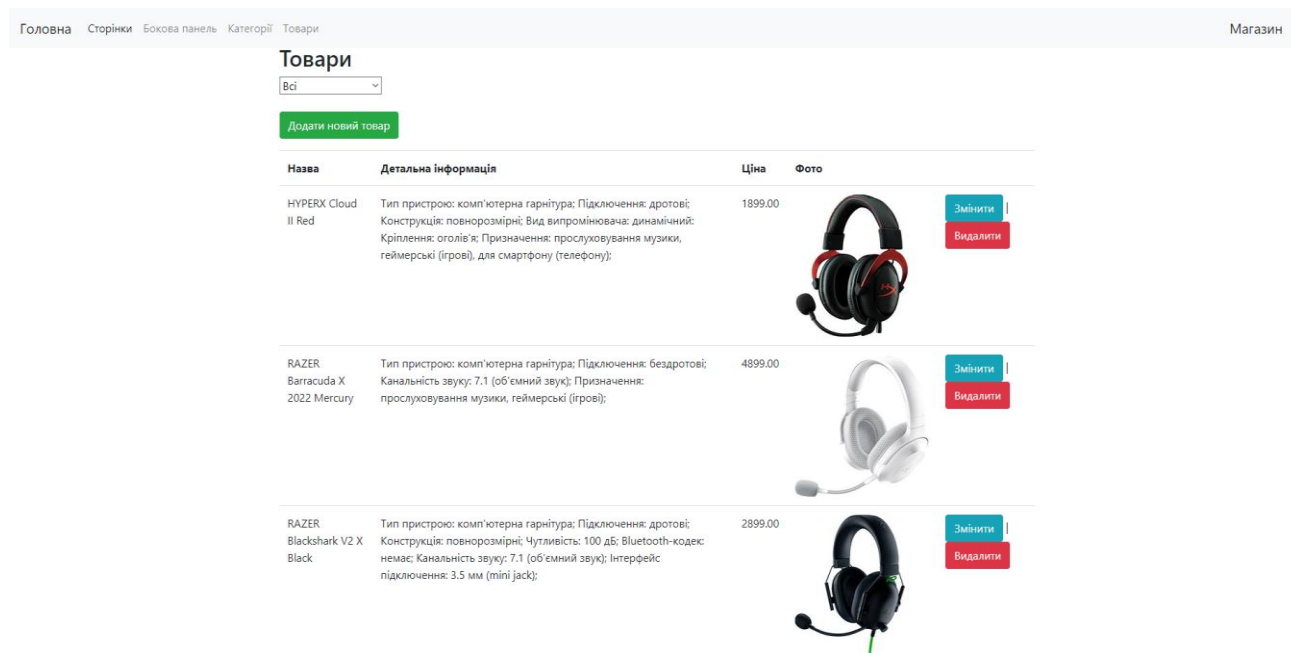


Рисунок 3.3.7 – список товарів які додав адміністратор

### 3.4 Технічні характеристики веб-додатка

Основні технічні характеристики веб-додатка включають:

1. Мова програмування: Для розробки веб-додатка ми використовуємо мову програмування C#.
2. Фреймворк: Веб-додаток розробляється на основі фреймворку ASP.NET MVC.
3. Система управління базами даних: Для зберігання даних використовується база даних SQL Server.
4. Веб-сервер: Веб-додаток буде розгортатися на сервері IIS (Internet Information Services).

5. Клієнтська частина: Використовується HTML, CSS і JavaScript для розробки користувацького інтерфейсу.

Цей веб-додаток не має особливих вимог до технічного середовища. Доступ до сайту забезпечено з будь-якого веб-браузера та пристрою. Рекомендується враховувати мінімальні вимоги до характеристик пристрою та браузера для оптимальної роботи веб-додатку:

Мінімальні системні вимоги для портативного комп'ютера:

- Наявність інтернет з'єднання.
- Вимоги до програмної частини:
- Версія операційної системи Windows 7 та вище, або Mac OS X 10.7

Lion та вище.

- Оперативна пам'ять 4гб або вище.
- Процесор: Pentium Dual-Core E5200 або вище.
- Мінімальні системні вимоги для мобільних пристроїв:
- Наявність інтернет з'єднання.
- Вимоги до програмної частини:
- Версія операційної системи: Android 5.1 та вище, або IOS 10 та вище.
- Процесор: Intel Atom Processor Z2520 1.2 ГГц або більш швидкий процесор, або для Apple пристроїв процесор – Apple A10.
- ОЗП: 512Мб.

### 3.5 Аналіз методів тестування веб-додатка

Тестування програмного забезпечення є невід'ємною частиною розробки програмних продуктів. Воно спрямоване на перевірку функціональності, якості та надійності програмного забезпечення перед його випуском. Тестування має

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		53

на меті виявлення помилок, аномалій та потенційних проблем, що можуть виникнути під час використання програмного забезпечення.

Тестування програмного забезпечення включає різні рівні тестування, методи та техніки. Ось деякі з них:

1. Тестування функціональності: Основним пріоритетом для успішного веб-сайту є безперебійна робота його функціональності на різних веб-сторінках, формах та підключеннях до бази даних. Важливо перевірити всі поля, значення за замовчуванням та файли cookie, щоб забезпечити зручний користувацький досвід. Також необхідно оптимізувати веб-сайт для пошукових систем та перевіряти HTML/CSS.

2. Тестування веб-юзабіліті: Сучасні користувачі очікують плавного переходу по веб-сайту та простої навігації. Веб-сайт повинен бути легким у використанні, зрозумілими спливаючими вікнами та повідомленнями, а також послідовним, щоб залучати та утримувати відвідувача. Зміст на сторінках веб-сайту має бути чітким і змістовним.

3. Тестування інтерфейсу: Важливо перевірити, що всі взаємодії між сервером додатків та веб-сервером працюють належним чином. Це включає перевірку сумісності програмного забезпечення, апаратного забезпечення, мережі та бази даних.

4. Тестування на сумісність веб-сайту: Варто перевірити сумісність веб-сайту з різними браузерами, операційними системами та мобільними пристроями. Необхідно виконати тестування на сумісність з різними браузерами, такими як Google Chrome, Mozilla Firefox, Microsoft Edge, Safari тощо. Також потрібно враховувати сумісність з різними версіями операційних систем, таких як Windows, macOS, Linux, а також з різними мобільними пристроями, такими як iOS та Android.

5. Тестування безпеки: Важливо забезпечити безпеку веб-додатку шляхом перевірки на наявність потенційних вразливостей, таких як кросс-сайтовий скриптинг (XSS), SQL-ін'єкції, вразливості автентифікації та

					КВРІПЗ.190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		54

авторизації, а також захисту від зламу та витоку даних. Також необхідно перевірити налаштування сертифікатів безпеки та захист від атак типу "force browsing".

6. Тестування продуктивності: Для веб-додатку важливо перевірити, як він працює під навантаженням. Це включає тестування швидкодії та витривалості системи. Можна використовувати інструменти для вимірювання часу відгуку сервера, завантаження сторінок, оптимізації запитів до бази даних та роботи з кешем.

7. Тестування на витривалість: Веб-додаток повинен бути стійким до помилок та непередбачуваних ситуацій. Тому важливо провести тестування на витривалість, перевіряючи, як додаток поводить себе при неправильному введенні даних, недоступності бази даних, помилках мережі та інших непередбачуваних ситуаціях.

### 3.6 Розробка тестів

Під час фази розроблення тестів був створений тестовий випадок, спеціально призначений для функціонального тестування. Цей тестовий випадок включає детальні кроки, тестові дані та очікувані результати, які дозволяють перевірити, чи відповідає програма своїм специфікаціям.

Основне завдання функціонального тестування полягає у перевірці, чи виконує програма очікувані функції та операції, описані у її специфікаціях. Це означає, що під час виконання тестового випадку перевіряється коректність роботи програми на кожному етапі повного циклу проходження замовлення.

Тестовий випадок може містити такі елементи, як введення тестових даних, взаємодія з інтерфейсом користувача або API, виконання певних дій і перевірку отриманих результатів. Шляхом виконання цього тестового випадку

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		55

можна виявити потенційні проблеми або некоректну роботу програми, якщо результати не відповідають очікуванням.

Таблиця 3.6.1 – Тестовий випадок.

Дії	Очікувані результати	Реальні результати
1. Запуск сайту	1. Має відкритись браузер, та повинна з'явитись головна сторінка веб-застосунку	1. Відкривається головна сторінка веб-застосунку
2. Перейти на сторінку реєстрації	2. При натисканні на кнопку «Логін», повинна відкритись сторінка авторизації, і далі, після натискання на кнопку «Створити акаунт», має відкритись сторінка реєстрації.	2. Після натискання на кнопку «Логін», відбувається перехід на сторінку авторизації, далі, натиснувши на кнопку «Створити акаунт» з'являється сторінка реєстрації.
3. Зареєструватись	3. Після введення даних у відповідні комірочки на сторінці реєстрації, повинна з'явитись сторінка авторизації.	3. Ввівши дані в комірочки реєстрації, відбувся автоматичний перехід до сторінки авторизації.
4. Авторизуватись	4. Після введення даних у комірочки авторизації та натискання кнопки «Авторизуватись», має відбутись перехід на головну сторінку веб-застосунку	4. Ввівши дані та натиснувши на кнопку «Авторизуватись», відбувся перехід на головну сторінку сайту.

Продовження таблиці 3.6.1

<p>5. Перейти до деталей товару з категорії «Клавіатури».</p>	<p>5. На головній сторінці натиснувши на кнопку «Клавіатури», має відкритись сторінка з товарами з категорії «Клавіатури», після цього, натиснувши, навпроти будь-якого товару, на кнопку «Детальніше» повинна відкритись сторінка з детальною інформацією про товар.</p>	<p>5. Після натискання на кнопку «Клавіатури» з списку категорій, з'явилась сторінка з товарами, після натискання на кнопку «Детальніше», навпроти з товаром, відкрилась сторінка з інформацією про товар.</p>
<p>6. Перейти до особистого кабінету користувача</p>	<p>6. На будь-якій сторінці веб-застосунка натиснувши на кнопку з ім'ям та прізвищем користувача, має здійснитись перехід на сторінку профіля користувача з його даними.</p>	<p>6. Натиснувши на кнопку з ім'ям та прізвищем користувача, відбувся перехід з усією інформацією користувача.</p>

На основі результатів функціонального тестування можна зробити висновок, що програмний продукт успішно пройшов всі тестові сценарії і відповідає очікуванням. Це свідчить про те, що розроблений проект можна вважати успішним і готовим до подальшого використання.

### 3.7 Висновок

У третьому розділі кваліфікаційної роботи була проведена розробка бази даних, програмних модулів та керівництво користувача для веб-застосунку з

продажу комп'ютерної периферії. Також були визначені технічні характеристики веб-додатка, проведений аналіз методів тестування та розробка тестів.

У підрозділі "Розробка бази даних" була створена структура бази даних для зберігання інформації про товари, замовлення, користувачів та інші важливі дані. Використовуючи Microsoft SQL Server, були створені таблиці, встановлені зв'язки між ними та реалізовані необхідні запити для доступу до даних.

Під час розробки програмних модулів була використана мова програмування та технології, що найкраще відповідають вимогам проекту. Були розроблені модулі для додавання товарів у кошик, оформлення замовлення та оплати, а також для взаємодії з базою даних.

В керівництві користувача була надана детальна інформація про функціональність веб-застосунку та пояснення щодо його використання. Користувачі отримали необхідні вказівки та кроки для здійснення покупок комп'ютерної периферії через веб-застосунок.

Технічні характеристики веб-додатка були визначені з урахуванням вимог до продуктивності, безпеки та масштабованості системи. Було встановлено необхідне середовище для роботи веб-додатка, а також виконано налаштування, щоб забезпечити його оптимальну продуктивність.

Аналіз методів тестування веб-додатка дозволив визначити ефективні підходи до перевірки якості розробленого веб-застосунку. Було розроблено тести для перевірки функціональності, безпеки та стабільності системи.

У результаті реалізації третього розділу були успішно розроблені база даних, програмні модулі та керівництво користувача для веб-застосунку з продажу комп'ютерної периферії. Веб-додаток відповідає вимогам щодо технічних характеристик та функціональності. Проведені тести підтверджують його надійність та придатність для використання у реальних умовах.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		58

## ВИСНОВКИ

У результаті виконання поставлених завдань у кваліфікаційній роботі було успішно розроблено веб-застосунок, що забезпечує зручний та ефективний процес покупки комп'ютерної периферії для користувачів. Вивчення потреб та вимог потенційних користувачів дало змогу збагнути їхні очікування та врахувати їх у розробці застосунку.

Аналіз існуючих ринкових рішень та веб-технологій допоміг вибрати оптимальні рішення для розробки. Завдяки цьому вдалося створити веб-застосунок, що відповідає сучасним технологічним стандартам та вимогам ринку.

Проектування архітектури та інтерфейсу веб-застосунку було здійснено з урахуванням потреб користувачів та найкращих практик. Чіткий та зручний інтерфейс дозволяє користувачам з легкістю вибрати товари, додавати їх у кошик, оформляти замовлення та здійснювати оплату.

Після розробки функціональності для додавання товарів у кошик, оформлення замовлення та оплати, веб-застосунок пройшов етап тестування з метою виявлення та усунення помилок та недоліків. Це дозволило забезпечити надійну та стабільну роботу застосунку перед його впровадженням.

Отримані результати підтверджують успішне виконання поставлених завдань та досягнення поставленої мети. Розроблений веб-застосунок відповідає вимогам та потребам користувачів, а його ефективність та придатність для використання у реальних умовах підтверджують його цінність. Використання цього веб-застосунку сприятиме зручній та успішній покупці комп'ютерної периферії для користувачів.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		59

## СПИСОК ДЖЕРЕЛ

1. Бондарев, В.М. "Об'єктно-орієнтовне програмування на C#": навч. посіб. / В.М. Бондарев. - Х.: Компанія СМІТ, 2009. - 224 с.
2. Павловський, В.В., Журавльова, І.С. "Веб-технології: HTML, CSS, JavaScript, PHP, AJAX, MySQL" / В.В. Павловський, І.С. Журавльова. - К.: КНЕУ, 2016. - 375 с.
3. Литвин, А. "Розробка веб-додатків на PHP. Об'єктно-орієнтоване програмування" / А. Литвин. - К.: ДМК Прес, 2017. - 384 с.
4. Запорожець, О.В. "Системи управління базами даних та знань в інтернет-технологіях" / О.В. Запорожець. - К.: НУХТ, 2013. - 192 с.
5. Батинцев, В.А. "Основи мережевого адміністрування" / В.А. Батинцев. - К.: ТОВ "Техніка", 2015. - 224 с.
6. Іванов, С.В. "Веб-дизайн та веб-програмування" / С.В. Іванов. - К.: Дія, 2018. - 352 с.
7. Дмитрієнко, С.В. "Основи баз даних" / С.В. Дмитрієнко. - К.: Айріс-Прес, 2015. - 320 с.
8. Савельєва, Н.И. "Разработка web-сайтов" / Н.И. Савельєва. - М.: Питер, 2017. - 416 с.
9. Фаулер, М. "Архітектура корпоративних програмних додатків" / М. Фаулер. - К.: Видавничий дім "Київський університет", 2016. - 576 с.
10. Макконнелл, С. "Джентельменський набір методів для об'єктно-орієнтованого дизайну" / С. Макконнелл. - К.: Видавничий дім "Київський університет", 2018. - 416 с.
11. Максимчук, О.Є. "Web-програмування та Інтернет-технології" / О.Є. Максимчук. - К.: Центр учбової літератури, 2019. - 432 с.
12. Рейнгольд, Д. "Об'єктно-орієнтоване програмування. Підхід через класи" / Д. Рейнгольд. - К.: Видавництво "Бібліотека", 2016. - 528 с.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		60

13. Мартін, Р. С. "Чиста архітектура" / Р. С. Мартін. - К.: Пітер, 2018. - 432 с.
14. Крістенсен, М. "Сервісно-орієнтована архітектура в дії" / М. Крістенсен. - К.: ДМК Прес, 2016. - 464 с.
15. Фаулер, М. "Рефакторинг. Покроковий підхід до вдосконалення програмного коду" / М. Фаулер. - К.: Видавничий дім "Київський університет", 2019. - 464 с.
16. Копка, Д., Уильямс, Д. "Розробка вимог до програмного забезпечення" / Д. Копка, Д. Уильямс. - К.: ДМК Прес, 2017. - 608 с.
17. Хант, Е., Томас, Д. "Програмування на платформі .NET" / Е. Хант, Д. Томас. - К.: ДМК Прес, 2016. - 704 с.
18. Макконнелл, С. "Совершенный код" / С. Макконнелл. - К.: Видавничий дім "Київський університет", 2017. - 896 с.
19. Фаулер, М. "Мова програмування С#" / М. Фаулер. - К.: ДМК Прес, 2016. - 464 с.
20. Гамма, Е., Хелм, Р., Джонсон, Р., Вліссідес, Дж. "Паттерни проектування. Підхід до об'єктно-орієнтованого проектування" / Е. Гамма, Р. Хелм, Р. Джонсон, Дж. Вліссідес. - К.: Видавничий дім "Київський університет", 2019. - 368 с.
21. Книжник, М.В. "Веб-дизайн. Технології HTML, CSS, JavaScript, PHP, MySQL" / М.В. Книжник. - К.: Ленгвіст, 2018. - 464 с.
22. Лоулер, М. "Система контролю версій Git" / М. Лоулер. - К.: Видавництво "Бібліотека", 2017. - 216 с.
23. Флаунері, Д. "RESTful API Design" / Д. Флаунері. - К.: ДМК Прес, 2016. - 296 с.
24. Павловський, В.В., Луців, І.І. "Проектування баз даних: методичні рекомендації" / В.В. Павловський, І.І. Луців. - К.: КНЕУ, 2017. - 192 с.
25. Черненко, М.В. "Веб-дизайн та веб-програмування: підручник" / М.В. Черненко. - К.: Центр учбової літератури, 2019. - 336 с.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		61

26. Зеланд, М. "Розумний фронтенд" / М. Зеланд. - К.: Пітер, 2018. - 320 с.
27. Крусіб, А., Вітовський, В., Загревський, А. "Програмування веб-додатків на Java" / А. Крусіб, В. Вітовський, А. Загревський. - К.: ДМК Прес, 2016. - 480 с.
28. Богданович, В.Ю., Вовк, В.П. "PHP: практика розробки веб-додатків" / В.Ю. Богданович, В.П. Вовк. - К.: ДМК Прес, 2017. - 464 с.
29. Макконнелл, С. "Ефективна робота з програмним кодом" / С. Макконнелл. - К.: Видавничий дім "Київський університет", 2018. - 928 с.
30. Рубін, Е., Бойс, Д. "Веб-програмування на платформі .NET" / Е. Рубін, Д. Бойс. - К.: ДМК Прес, 2016. - 416 с.
31. Ільїн, А. "Python. Підручник" / А. Ільїн. - К.: Видавництво "Бібліотека", 2017. - 416 с.
32. Фаулер, М. "Рефакторинг баз даних" / М. Фаулер. - К.: Видавничий дім "Київський університет", 2019. - 320 с.
33. Кейлі, Д., Гарланд, Б. "Об'єктно-орієнтоване проектування на Java" / Д. Кейлі, Б. Гарланд. - К.: Видавничий дім "Київський університет", 2017. - 480 с.
34. Бейтс, Б., Семендюк, В. "Системи керування базами даних" / Б. Бейтс, В. Семендюк. - К.: Центр учбової літератури, 2019. - 464 с.
35. Скляр, В. "PHP 7 в подлиннике" / В. Скляр. - К.: Вільямс, 2016. - 800 с.
36. Дуглас, К. "JavaScript. Підручник для початківців" / К. Дуглас. - К.: Видавництво "Бібліотека", 2017. - 720 с.
37. Макконнелл, С. "Управление разработкой программных проектов" / С. Макконнелл. - К.: Видавничий дім "Київський університет", 2019. - 704 с.
38. Роуч, Дж. "Веб-разработка с применением Node и Express" / Дж. Роуч. - К.: Видавництво "Бібліотека", 2018. - 320 с.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		62

39. Гамма, Е., Хелм, Р., Джонсон, Р., Вліссідес, Дж. "Аналіз і проектування об'єктно-орієнтованих систем" / Е. Гамма, Р. Хелм, Р. Джонсон, Дж. Вліссідес. - К.: Видавничий дім "Київський університет", 2017. - 384 с.

40. Крицька, О.В., Матвійчук, В.О. "Інформаційні системи і технології управління проектами" / О.В. Крицька, В.О. Матвійчук. - К.: КНЕУ, 2018. - 256 с.

41. Гуржій, А.Є. "Архітектура комп'ютера. Сучасні мікропроцесори" / А.Є. Гуржій. - К.: НУБіП України, 2016. - 312 с.

42. Буткевич, О.В. "Концептуальне моделювання інформаційних систем: засоби, методи, практика" / О.В. Буткевич. - К.: НУБіП України, 2018. - 332 с.

43. Денисюк, Ю.В. "Теорія прийняття рішень в інформаційних системах" / Ю.В. Денисюк. - К.: НУБіП України, 2017. - 252 с.

44. Захарчук, В.В. "Методи та засоби розробки програмного забезпечення" / В.В. Захарчук. - К.: НУБіП України, 2019. - 208 с.

45. Лук'янчук, І.В. "Основи баз даних. Моделювання і проектування" / І.В. Лук'янчук. - К.: НУБіП України, 2015. - 240 с.

46. Поліщук, О.М. "Web-технології. Основи проектування та програмування" / О.М. Поліщук. - К.: НУБіП України, 2016. - 328 с.

					КВРІПЗ. 190134.01.10.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		63

ДОДАТОК А

**ТЕХНІЧНЕ ЗАВДАННЯ**

## **Введення**

Робота виконується в рамках проекту розробки веб-застосунку для продажу комп'ютерних периферійних пристроїв, у вигляді сайту, що дозволяє здійснювати онлайн покупки.

## **1 Підстава для розробки**

Підставою для розробки даного веб-застосунку є «Завдання на кваліфікаційну роботу», затверджене завідувачем кафедри програмного забезпечення. Найменування розробки: «Веб-застосунок для продажу комп'ютерних периферійних пристроїв».

## **2 Призначення розробки**

### **2.1 Функціональне призначення**

Функціональне призначення веб-застосунку - надати клієнтам можливість перегляду та придбання комп'ютерної периферії через персональний комп'ютер чи смартфон. Додаток також забезпечує можливість оплати товарів в електронному вигляді та виступає посередником між клієнтами та працівниками магазину.

### **2.2. Експлуатаційні призначення**

Веб-застосунок має використовуватись звичайними користувачами без професійної підготовки за допомогою браузера на ПК, ноутбуках, планшетах та смартфонах, підтримуючи операційні системи Windows, Mac, Android та iOS.

## **3 Вимоги до програми**

### **3.1 Вимоги до функціональних характеристик**

Веб-застосунок для продажу комп'ютерних периферійних пристроїв має забезпечувати наступні функції:

- додавання товарів до каталогу;
- реєстрація та авторизація у системі;

- налаштування особистого профілю(редагування даних, зміна пароллю);
- відображення товарів та інформації про них;
- фільтрація для пошуку товарів;
- додавання товарів до кошику;
- оформлення замовлення.

### **3.2 Вимоги до надійності**

Веб-застосунок повинен забезпечувати захист даних користувачів та передавати особисті дані клієнтів і інформацію про їх спосіб оплати тільки в зашифрованому вигляді. Система повинна відображати повідомлення про помилку та перенаправляти користувача на належно функціонуючу сторінку при виникненні помилок.

### **3.3 Умови експлуатації**

Умови експлуатації мають відповідати санітарним і технічним нормам експлуатації персонального комп'ютера, при температурі та відносній вологості навколишнього середовища, визначених для персональної обчислювальної техніки згідно з ГОСТ 15150-69 [7].

Для обслуговування веб-застосунку допускаються тільки спеціально навчені адміністратори або розробники. До користування веб-застосунком допускаються користувачі мережі Інтернет.

### **3.4 Вимоги до складу та параметрів технічних засобів**

Мінімальні системні вимоги для портативного комп'ютера:

- Наявність інтернет з'єднання.
- 
- Вимоги до програмної частини:

- Версія операційної системи Windows 7 та вище, або Mac OS X 10.7 Lion та вище.

- Оперативна пам'ять 4Гб або вище.
- Процесор: Pentium Dual-Core E5200 або вище.

Мінімальні системні вимоги для мобільних пристроїв:

- Наявність інтернет з'єднання.
- 
- Вимоги до програмної частини:
- Версія операційної системи: Android 5.1 та вище, або IOS 10 та вище.
- Процесор: Intel Atom Processor Z2520 1.2 ГГц або більш швидкий процесор, або для Apple пристроїв процесор – Apple A10.
- ОЗП: 512Мб.

### **3.5 Вимоги до інформаційної та програмної сумісності**

Для створення клієнтської частини веб-застосунку використовувалася мова розмітки HTML, каскадні таблиці стилів CSS та мова програмування JS

Для створення серверної частини веб-застосунку використовувалися технології платформи ASP.NET, мова програмування C# та фреймворки ASP.NET MVC і Entity Framework

### **3.6 Спеціальні вимоги**

Для коректної функціональності ПЗ спеціальні вимоги відсутні.

## **4 Вимоги до програмної документації**

Набір документації, що надається у момент здачі проекту:

- Опис модулів програми з демонстрацією їх роботи
- Текст програми з коментарями

- Керівництво користувача для звичайних користувачів та адміністратора
- Довідкова інформація
- Керівництво для розробника

## 5 Стадії та етапи розробки

Стадії та етапи розробки веб-застосунку для продажу комп'ютерних периферійних пристроїв продемонстровано у таблиці 1

Таблиця 1 - стадії та етапи розробки

Стадія розробки	Етапи робіт	Зміст робіт
Технічне завдання 02.01 – 31.01.2023	Обґрунтування необхідності розробки програми	Коротка характеристика програмного забезпечення; підстава і призначення розробки; вимоги до програмної системи і документація; стадії і етапи розробки програми; порядок контролю і приймання
Ескізний проект 01.02 – 14.02 2023	Розробка ескізного проекту	Попередня розробка структури вхідних і вихідних даних; уточнення середовища програмування; розробка і опис загальної алгоритмічної структури системи, що буде розроблюватися
Технічний проект 15.02 – 28.02 2023	Розробка технічного проекту	Уточнення структури вхідних і вихідних даних; розробка докладного алгоритму; розробка структури програми; остаточне визначення конфігурації технічних засобів
Робочий проект 01.03 – 10.04.2023	Розробка програмного забезпечення	Реалізація програмного забезпечення; відладка; проведення попереднього тестування
Розробка програмної документації 11.04 – 20.04.2023	Розробка документації до програмного забезпечення	Розробка необхідної документації, передбаченої технічним завданням
Тестування системи 21.04 – 30.04.2023	Проведення тестування програмного забезпечення	Розробка методики тестування; проведення основних тестів; коректування програмного забезпечення
Впровадження	Підготовка і передача програми	Підготовка і передача програмного забезпечення; навчання персоналу використуванню програмного забезпечення; внесення коректувань в програмне забезпечення і документацію

## **6 Порядок контролю та приймання**

Контроль здійснюється користувачами застосунку підключеними на етапі тестування. Прийом програми здійснюється після повного її розгортання, тестування і налаштування для справного функціонування

## ДОДАТОК Б

### КОД (ЛІСТИНГ) ПРОГРАМИ

Код ShopController.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using MVC_Store.Models.Data;
using MVC_Store.Models.ViewModels.Shop;

namespace MVC_Store.Controllers
{
    public class ShopController : Controller
    {
        private readonly Db _db;

        public ShopController()
        {
            _db = new Db();
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                _db.Dispose();
            }
            base.Dispose(disposing);
        }

        public ActionResult Index()
        {
            return RedirectToAction("Index", "Pages");
        }

        public ActionResult CategoryMenuPartial()
        {
            List<CategoryVM> categoryVMList = _db.Categories
                .OrderBy(x => x.Sorting)
                .Select(x => new CategoryVM(x))
                .ToList();

            return PartialView("_CategoryMenuPartial", categoryVMList);
        }

        public ActionResult Category(string name)
        {
            CategoryDTO = _db.Categories.FirstOrDefault(x => x.Slug == name);

            if (categoryDTO == null)
            {
                return RedirectToAction("Index", "Shop");
            }
        }
    }
}
```

```

    }

    List<ProductVM> productVMList = _db.Products
        .Where(x => x.CategoryId == categoryDTO.Id)
        .Select(x => new ProductVM(x))
        .ToList();

    ViewBag.CategoryName = categoryDTO.Name;

    return View(productVMList);
}

[ActionName("product-details")]
public ActionResult ProductDetails(string name)
{
    ProductVM model = _db.Products
        .Where(x => x.Slug == name)
        .Select(x => new ProductVM(x))
        .FirstOrDefault();

    if (model == null)
    {
        return RedirectToAction("Index", "Shop");
    }

    string galleryPath = Server.MapPath("~/Images/Uploads/Products/" + model.Id
+ "/Gallery/Thumbs");
    model.GalleryImages =
Directory.EnumerateFiles(galleryPath).Select(Path.GetFileName);

    return View("ProductDetails", model);
}
}
}

```

### Код CartController.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Mail;
using System.Web;
using System.Web.Mvc;
using MVC_Store.Models.Data;
using MVC_Store.Models.ViewModels.Cart;

namespace MVC_Store.Controllers
{
    public class CartController : Controller
    {
        public ActionResult Index()
        {
            var cart = GetCartItems();

            if (cart.Count == 0 || cart == null)
            {
                ViewBag.Message = "Ваш кошик пустий.";
                return View();
            }

            decimal total = CalculateTotal(cart);

            ViewBag.GrandTotal = total;

```

```

        return View(cart);
    }

    public ActionResult CartPartial()
    {
        CartVM model = new CartVM();
        var cart = GetCartItems();

        if (cart != null)
        {
            int qty = 0;
            decimal price = 0m;

            foreach (var item in cart)
            {
                qty += item.Quantity;
                price += item.Quantity * item.Price;
            }

            model.Quantity = qty;
            model.Price = price;
        }
        else
        {
            model.Quantity = 0;
            model.Price = 0m;
        }

        return PartialView("_CartPartial", model);
    }

    public ActionResult AddToCartPartial(int id)
    {
        var cart = GetCartItems();
        CartVM model = new CartVM();

        using (Db = new Db())
        {
            ProductDTO product = db.Products.Find(id);

            var productInCart = cart.FirstOrDefault(x => x.ProductId == id);

            if (productInCart == null)
            {
                cart.Add(new CartVM()
                {
                    ProductId = product.Id,
                    ProductName = product.Name,
                    Quantity = 1,
                    Price = product.Price,
                    Image = product.ImageName
                });
            }
            else
            {
                productInCart.Quantity++;
            }
        }

        int qty = 0;
        decimal price = 0m;

        foreach (var item in cart)
        {

```

```

        qty += item.Quantity;
        price += item.Quantity * item.Price;
    }

    model.Quantity = qty;
    model.Price = price;

    SaveCartItems(cart);

    return PartialView("_AddToCartPartial", model);
}

public JsonResult IncrementProduct(int productId)
{
    var cart = GetCartItems();

    using (Db = new Db())
    {
        CartVM model = cart.FirstOrDefault(x => x.ProductId == productId);

        if (model != null)
        {
            model.Quantity++;

            var result = new { qty = model.Quantity, price = model.Price };

            SaveCartItems(cart);

            return Json(result, JsonRequestBehavior.AllowGet);
        }
    }

    return Json(null);
}

public ActionResult DecrementProduct(int productId)
{
    var cart = GetCartItems();

    using (Db = new Db())
    {
        CartVM model = cart.FirstOrDefault(x => x.ProductId == productId);

        if (model != null)
        {
            if (model.Quantity > 1)
                model.Quantity--;
            else
            {
                model.Quantity = 0;
                cart.Remove(model);
            }

            var result = new { qty = model.Quantity, price = model.Price };

            SaveCartItems(cart);

            return Json(result, JsonRequestBehavior.AllowGet);
        }
    }

    return Json(null);
}

```

```

public void RemoveProduct(int productId)
{
    var cart = GetCartItems();

    using (Db = new Db())
    {
        CartVM model = cart.FirstOrDefault(x => x.ProductId == productId);

        if (model != null)
            cart.Remove(model);
    }

    SaveCartItems(cart);
}

public ActionResult PaypalPartial()
{
    var cart = GetCartItems();

    return PartialView(cart);
}

[HttpPost]
public void PlaceOrder()
{
    var cart = GetCartItems();

    string userName = User.Identity.Name;

    int orderId = 0;

    using (Db = new Db())
    {
        OrderDTO = new OrderDTO();

        var user = db.Users.FirstOrDefault(x => x.Username == userName);

        if (user != null)
        {
            orderDto.UserId = user.Id;
            orderDto.CreatedAt = DateTime.Now;

            db.Orders.Add(orderDto);
            db.SaveChanges();

            orderId = orderDto.OrderId;

            foreach (var item in cart)
            {
                OrderDetailsDTO = new OrderDetailsDTO()
                {
                    OrderId = orderId,
                    UserId = user.Id,
                    ProductId = item.ProductId,
                    Quantity = item.Quantity
                };

                db.OrderDetails.Add(orderDetailsDto);
            }

            db.SaveChanges();
        }
    }
}

```

```

        SendOrderConfirmationEmail(orderId);

        ClearCart();

        Response.Redirect("~/Cart/OrderConfirmation");
    }

    private List<CartVM> GetCartItems()
    {
        return Session["cart"] as List<CartVM> ?? new List<CartVM>();
    }

    private void SaveCartItems(List<CartVM> cart)
    {
        Session["cart"] = cart;
    }

    private decimal CalculateTotal(List<CartVM> cart)
    {
        decimal total = 0m;

        foreach (var item in cart)
        {
            total += item.Total;
        }

        return total;
    }

    private void ClearCart()
    {
        Session["cart"] = null;
    }

    private void SendOrderConfirmationEmail(int orderId)
    {
        var client = new SmtpClient("smtp.mailtrap.io", 2525)
        {
            Credentials = new NetworkCredential("ee053ae36467a9",
"8bf25933605f02"),
            EnableSsl = true
        };
        client.Send("shop@example.com", "admin@example.com", "Новое замовлення",
$"Ви отримали замовлення. Номер замовлення: {orderId}");
    }
}
}

```

## Код AccountController.cs

```

using System;
using System.Linq;
using System.Web.Mvc;
using System.Web.Security;
using MVC_Store.Models.Data;
using MVC_Store.Models.ViewModels.Account;
using Microsoft.AspNet.Identity;

namespace MVC_Store.Controllers
{
    public class AccountController : Controller
    {

```

```

public ActionResult Index()
{
    return RedirectToAction("Login");
}

[HttpGet]
public ActionResult Register()
{
    return View("Register");
}

[HttpPost]
public ActionResult Register(RegisterViewModel model)
{
    if (!ModelState.IsValid)
        return View("Register", model);

    if (!model.Password.Equals(model.ConfirmPassword))
    {
        ModelState.AddModelError("", "Паролі не співпадають!");
        return View("Register", model);
    }

    using (Db db = new Db())
    {
        if (db.Users.Any(x => x.Username.Equals(model.Username)))
        {
            ModelState.AddModelError("", $"Ім'я користувача {model.Username}
вже використовується.");
            model.Username = "";
            return View("Register", model);
        }

        UserDTO userDTO = new UserDTO()
        {
            FirstName = model.FirstName,
            LastName = model.LastName,
            Email = model.Email,
            Username = model.Username,
            Password = model.Password
        };

        db.Users.Add(userDTO);
        db.SaveChanges();

        int id = userDTO.Id;

        UserRoleDTO userRoleDTO = new UserRoleDTO()
        {
            UserId = id,
            RoleId = 2
        };

        db.UserRoles.Add(userRoleDTO);
        db.SaveChanges();

        TempData["SuccessMessage"] = "Ви успішно зареєстровані. Увійдіть, щоб
продовжити.";

        return RedirectToAction("Login");
    }

    [HttpGet]

```

```

public ActionResult Login()
{
    string userName = User.Identity.Name;

    if (!string.IsNullOrEmpty(userName))
        return RedirectToAction("UserProfile");

    return View();
}

[HttpPost]
public ActionResult Login(LoginViewModel model)
{
    if (!ModelState.IsValid)
        return View(model);

    bool isValid = false;

    using (Db db = new Db())
    {
        if (db.Users.Any(x => x.Username.Equals(model.Username) &&
x.Password.Equals(model.Password)))
            isValid = true;

        if (!isValid)
        {
            ModelState.AddModelError("", "Неправильне ім'я користувача або
пароль.");
            return View(model);
        }
        else
        {
            FormsAuthentication.SetAuthCookie(model.Username,
model.RememberMe);
            return Redirect(FormsAuthentication.GetRedirectUrl(model.Username,
model.RememberMe));
        }
    }
}

public ActionResult Logout()
{
    FormsAuthentication.SignOut();
    return RedirectToAction("Login");
}

public ActionResult UserNavPartial()
{
    string userName = User.Identity.Name;

    UserNavPartialViewModel model;

    using (Db db = new Db())
    {
        UserDTO dto = db.Users.FirstOrDefault(x => x.Username == userName);

        model = new UserNavPartialViewModel()
        {
            FirstName = dto.FirstName,
            LastName = dto.LastName
        };
    }

    return PartialView(model);
}

```

```

}

[HttpGet]
public ActionResult UserProfile()
{
    string userName = User.Identity.Name;

    UserProfileViewModel model;

    using (Db db = new Db())
    {
        UserDTO dto = db.Users.FirstOrDefault(x => x.Username == userName);

        model = new UserProfileViewModel(dto);
    }

    return View("UserProfile", model);
}

[HttpPost]
public ActionResult UserProfile(UserProfileViewModel model)
{
    bool isUserNameChanged = false;

    if (!ModelState.IsValid)
    {
        return View("UserProfile", model);
    }

    if (!string.IsNullOrEmpty(model.Password))
    {
        if (!model.Password.Equals(model.ConfirmPassword))
        {
            ModelState.AddModelError("", "Паролі не співпадають.");
            return View("UserProfile", model);
        }
    }

    using (Db db = new Db())
    {
        string userName = User.Identity.Name;

        if (userName != model.Username)
        {
            userName = model.Username;
            isUserNameChanged = true;
        }

        if (db.Users.Where(x => x.Id != model.Id).Any(x => x.Username ==
userName))
        {
            ModelState.AddModelError("", $"Ім'я користувача {model.Username}
вже використовується. Виберіть інше ім'я.");
            model.Username = "";
            return View("UserProfile", model);
        }

        UserDTO dto = db.Users.Find(model.Id);

        dto.FirstName = model.FirstName;
        dto.LastName = model.LastName;
        dto.Email = model.Email;
        dto.Username = model.Username;
    }
}

```

```

        if (!string.IsNullOrEmpty(model.Password))
        {
            dto.Password = model.Password;
        }

        db.SaveChanges();
    }

    TempData["SuccessMessage"] = "Ваші дані було успішно оновлено!";

    if (!isUserNameChanged)
        return View("UserProfile", model);
    else
        return RedirectToAction("Logout");
    }
}
}

```

## Код Login.cshtml

```

@model MVC_Store.Models.ViewModels.Account.LoginUserVM

@{
    ViewBag.Title = "Авторизація";
}

<head>
    <link type="text/css" rel="stylesheet" href="@Url.Content("~/Content/Style.css")" />
</head>

<h2>Авторизація</h2>

@if (TempData["SM"] != null)
{
    <div class="alert alert-success">
        @TempData["SM"]
    </div>
}

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            <p class="control-label col-md-2">Логін</p>
            <div class="col-md-10">
                @Html.EditorFor(model => model.Username, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Username, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            <p class="control-label col-md-2">Пароль</p>
            <div class="col-md-10">
                @Html.PasswordFor(model => model.Password, new { @class = "form-
control" })
            </div>
        </div>
    </div>
}

```



```

        <p class="control-label col-md-2">Електронна пошта</p>
        <div class="col-md-10">
            @Html.EditorFor(model => model.EmailAddress, new { htmlAttributes = new {
@class = "form-control", placeholder = "Введіть електронну пошту" } })
            @Html.ValidationMessageFor(model => model.EmailAddress, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        <p class="control-label col-md-2">Логін</p>
        <div class="col-md-10">
            @Html.EditorFor(model => model.Username, new { htmlAttributes = new {
@class = "form-control", placeholder = "Введіть логін" } })
            @Html.ValidationMessageFor(model => model.Username, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        <p class="control-label col-md-2">Пароль</p>
        <div class="col-md-10">
            @Html.PasswordFor(model => model.Password, new { @class = "form-control",
placeholder = "Введіть пароль" })
            @Html.ValidationMessageFor(model => model.Password, "", new { @class =
"text-danger" })
        </div>
    </div>

    <div class="form-group">
        <p class="control-label col-md-2">Підтвердити пароль</p>
        <div class="col-md-10">
            @Html.PasswordFor(model => model.ConfirmPassword, new { @class = "form-
control", placeholder = "Підтвердіть пароль" })
            @Html.ValidationMessageFor(model => model.ConfirmPassword, "", new { @class
= "text-danger" })
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Створити" class="btn btn-default" />
        </div>
    </div>
    }

```

## ДОДАТОК В

### ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

# Кваліфікаційна робота

На тему: «Веб-застосунок для продажу комп'ютерних периферійних пристроїв»

Виконав:  
студент групи ПЗ-19-1  
Каспруков А.В.

Керівник роботи:  
канд. техн. наук, доцент  
Бедратюк Л. П.

## Актуальність курсового проекту

У сучасному світі комп'ютерна техніка та Інтернет займають важливе місце в житті людини, що призвело до зростання попиту на інтернет-магазини. Ці магазини дозволяють зручно та швидко здійснювати покупки без виходу з дому. За останні роки кількість інтернет-магазинів в Україні значно зросла, а обсяги продажу через Інтернет постійно збільшується.

Комп'ютерна периферія є важливим компонентом сучасних комп'ютерних систем, і попит на неї постійно зростає. Зміни в способі життя, такі як робота з дому та дистанційна освіта, вимагають більше комп'ютерних пристроїв та аксесуарів, що сприяє популярності інтернет-магазинів комп'ютерної периферії.

Отже, розробка функціонуючого інтернет-магазину комп'ютерної периферії є актуальним завданням, що задовольняє потреби ринку та споживачів.

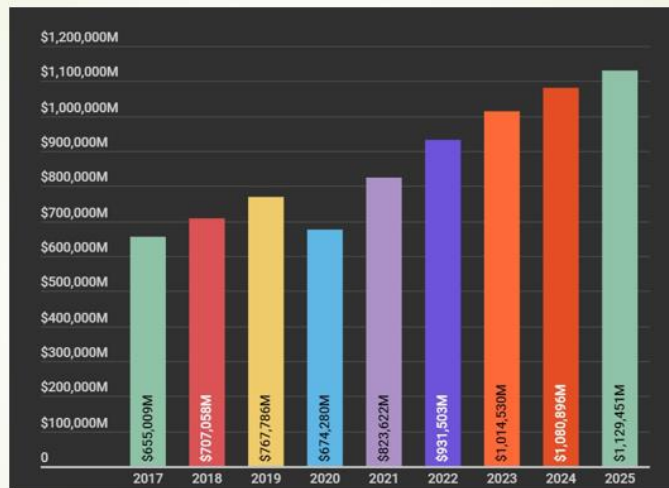


Рис. 1 - Динаміка світових обсягів електронного роздрібного продажу

## Мета кваліфікаційної роботи

Мета кваліфікаційної роботи полягає в розробці концепції та створенні інтернет-магазину комп'ютерної периферії. Робота включає в себе вивчення потреб та побажань споживачів, аналіз ринку та конкурентів, вибір технологічної платформи та створення дизайну інтерфейсу магазину. Також в роботі буде розглянуто питання забезпечення безпеки операцій та даних споживачів при здійсненні покупок в інтернет-магазині.

Очікуваним результатом кваліфікаційної роботи буде розробка та запуск функціонуючого інтернет-магазину комп'ютерної периферії, який зможе забезпечити споживачів широким вибором якісних товарів за доступними цінами. Також, магазин буде мати зручний та простий інтерфейс для замовлення та оплати, що забезпечить комфортний та швидкий процес покупок.

## Визначення функціональних вимог до ПЗ

Для вдалого функціонування веб-застосунку комп'ютерної периферії необхідне належне програмне забезпечення, що у подальшому забезпечить належну надійність, ефективність роботи магазину та задоволення потреб клієнтів.

Функції користувача, які повинні бути виконанні ПЗ включають:

- Можливість перегляду товарів у магазині, їх характеристики, фотографій та цін.
- Можливість оформлення замовлення.
- Можливість сплати замовлення.
- Можливість виконувати вхід користувачу, або реєстрацію, якщо користувач вперше на сайт.
- Можливість додавання товару до кошика.

## Визначення функціональних вимог до ПЗ

Не менш важливим є реалізація функціоналу для адміністрації ПЗ. Адміністратор повинен мати можливість взаємодіяти з товарами на сайті, тобто, додавати, видаляти, редагувати товари. Відслідковувати замовлення користувачів.

Проект в кінцевому результаті повинен складатись з головної сторінки, з інформацією про магазин, та каталогу, з якого користувачі можуть переходити на сторінку з товарами. Також сторінка авторизації/реєстрації, сторінки редагування облікового запису та сторінки-корзини.

## Основні актори ПЗ



Гість



Клієнт



Адміністратор

Актор	Функції
Гість (користувач без реєстрації на сайті)	Перегляд товарів, їх характеристик, фотографій та ціни; використання категорій для пошуку потрібних товарів; виконання реєстрації для повноцінного використання сайту.
Клієнт	Перегляд товарів, їх характеристик, фотографій та ціни; використання категорій для пошуку потрібних товарів; авторизація; редагування профілю, яка включає зміну паролю, пошти, ім'я та прізвища;
Адміністратор сайту	Керування контентом сайту (додавання/редагування текстів, фото, тощо.); підтримка та оновлення ПЗ магазину; налаштування безпеки сайту та захист від <u>взломів</u> , DDOS-атак та зловмисного програмного забезпечення.

## Архітектура

У рамках проекту буде використана клієнт-серверна архітектура, де клієнтська частина представлена веб-інтерфейсом для користувачів, а серверна частина відповідає за обробку запитів, доступ до бази даних та бізнес-логіку. Буде використано паттерн MVC (Model-View-Controller), що дозволяє розділити логіку програми на модель даних, представлення та контролери.

Модель (Model) відповідає за обробку даних, включаючи роботу з базою даних, об'єктами доменної моделі та бізнес-логікою. Вона забезпечує доступ до даних та виконання операцій з ними.

Вид (View) відображає інформацію користувачам та обробляє їх взаємодію з системою. Він відповідає за створення та відображення веб-сторінок, інтерфейсів користувача, форм, таблиць тощо. Вид взаємодіє з моделлю та контролером для отримання та передачі даних.

Контролер (Controller) приймає запити від користувача та керує потоком даних між моделлю та видом. Він відповідає за обробку запитів, виконання відповідних дій та відправку відповідей користувачеві.

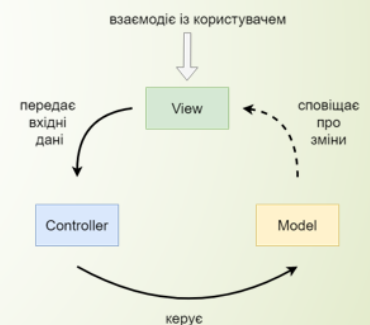


Рисунок 2 – діаграма взаємодії між компонентами шаблону MVC

На етапі проектування було проаналізовано основні типи баз даних:

Характеристика	Реляційні бази даних	Бази даних "NoSQL"
Структура даних	Таблиці зі схемою	Різноманітні структури даних
Зв'язки	Реляційні зв'язки	Зв'язки зазвичай відсутні або слабкі
Гнучкість схеми	Суворі схеми даних	Гнучка схема даних
Масштабованість	Зазвичай вертикальна масштабованість	Горизонтальна масштабованість
Швидкодія	Зазвичай менш ефективна	Зазвичай краща швидкодія
Запити	SQL-запити	Неструктуровані або спеціалізовані запити
Транзакції	Підтримка транзакцій	Залежить від конкретної реалізації
Розширюваність	Обмежена розширюваність	Краща розширюваність
Гнучкість моделі	Обмежена гнучкість моделі	Велика гнучкість моделі
Складність утримання	Зазвичай більш складно утримувати	Зазвичай менш складно утримувати
Використання	Широко використовується у традиційних системах	Застосовується у веб-розробці та областях з високою потоком даних

## Результати роботи. Головна сторінка

KasperShop [Логін](#)
Ваш кошик: 0 товар(ів) - 0 zł [Купити](#)

**Категорії**

- [Клавіатури](#)
- [Миші](#)
- [Коврики](#)
- [Навушники](#)
- [Мікрофони](#)
- [Маршрутизатори](#)
- [Веб-камери](#)
- [Крісла](#)
- [Принтери](#)

Instagram  
Facebook

### KasperShop


Ефективність роботи на комп'ютері залежить від якості приєднаних периферійних пристроїв. Вони забезпечують введення та виведення даних, а також відповідають за інші процеси.


Під периферійними пристроями для ПК нають на увазі пристрої та аксесуари, призначені для розширення функціональних можливостей.

**Різновиди комп'ютерної периферії**

Периферія може знаходитися всередині ПК або зовні.

- Внутрішня апаратура часто входить до базової комплектації портативних та стаціонарних комп'ютерів.
- Зовнішня продається окремо і підключається за допомогою дротових та бездротових інтерфейсів.





## Результати роботи. Сторінка «Навушники» з каталогу




KasperShop [Логін](#) Ваш кошик: 0 товар(ів) - 0 €  
[Кошик](#)

### Категорії

- Клавіатури
- Миші
- Колірки
- Навушники
- Мікрофони
- Маршрутизатори
- Веб-камери
- Крісла
- Принтери

[Instagram](#)  
[Facebook](#)

### Навушники

Назва	Детальніше	Ціна	Фото	
HYPERX Cloud II Red	Тип пристрою: комп'ютерна гарнітура; Підключення: дротові; Конструкція: повнорозмірні; Вид випромінювача: динамічний; Кріплення: оголіть; Призначення: прослуховування музики, геймерські (ігрові), для смартфона (телефону);	1899 €		<a href="#">Детальніше</a>
RAZER Barracuda X 2022 Mercury	Тип пристрою: комп'ютерна гарнітура; Підключення: бездротові; Канальність звуку: 7.1 (об'ємний звук); Призначення: прослуховування музики, геймерські (ігрові);	4899 €		<a href="#">Детальніше</a>
RAZER Blackshark V2 X Black	Тип пристрою: комп'ютерна гарнітура; Підключення: дротові; Конструкція: повнорозмірні; Чутливість: 100 дБ; Bluetooth-кодес: немає; Канальність звуку: 7.1 (об'ємний звук); Інтерфейс підключення: 3.5 мм (mini jack);	2899 €		<a href="#">Детальніше</a>




© 2023 - KasperShop

## Результати роботи. Сторінка з товарами в панелі адміністратора.

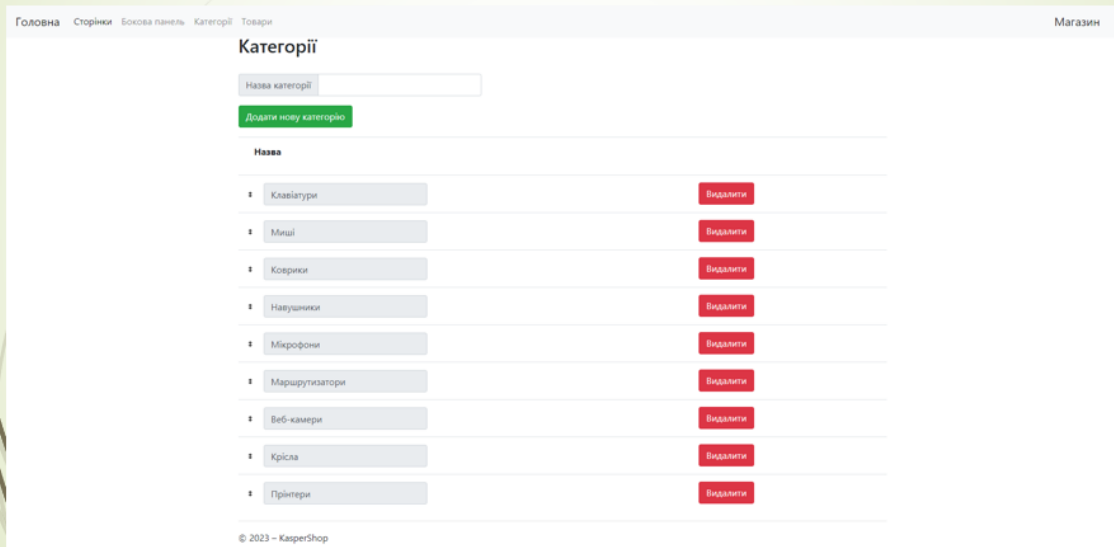
[Головна](#) [Сторінки](#) [Бокова панель](#) [Категорії](#) [Товари](#) Магазин

### Товари

Всі [Додати новий товар](#)

Назва	Детальна інформація	Ціна	Фото	
HYPERX Cloud II Red	Тип пристрою: комп'ютерна гарнітура; Підключення: дротові; Конструкція: повнорозмірні; Вид випромінювача: динамічний; Кріплення: оголіть; Призначення: прослуховування музики, геймерські (ігрові), для смартфона (телефону);	1899,00		<a href="#">Зняти</a> <a href="#">Видати</a>
RAZER Barracuda X 2022 Mercury	Тип пристрою: комп'ютерна гарнітура; Підключення: бездротові; Канальність звуку: 7.1 (об'ємний звук); Призначення: прослуховування музики, геймерські (ігрові);	4899,00		<a href="#">Зняти</a> <a href="#">Видати</a>
RAZER Blackshark V2 X Black	Тип пристрою: комп'ютерна гарнітура; Підключення: дротові; Конструкція: повнорозмірні; Чутливість: 100 дБ; Bluetooth-кодес: немає; Канальність звуку: 7.1 (об'ємний звук); Інтерфейс підключення: 3.5 мм (mini jack);	2899,00		<a href="#">Зняти</a> <a href="#">Видати</a>

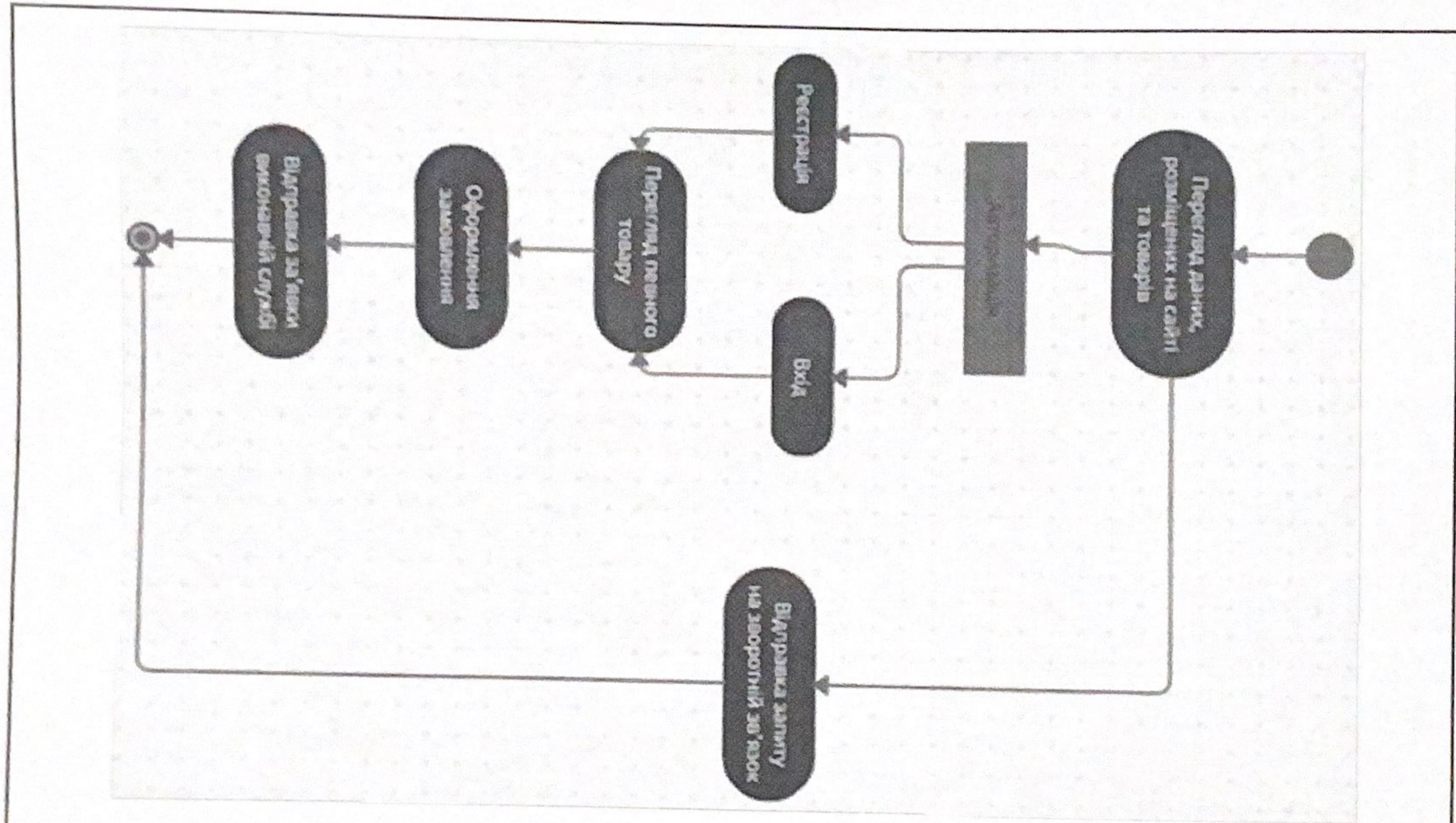
## Результати роботи. Сторінка «Категорії» в панелі адміністратора.



## Висновок

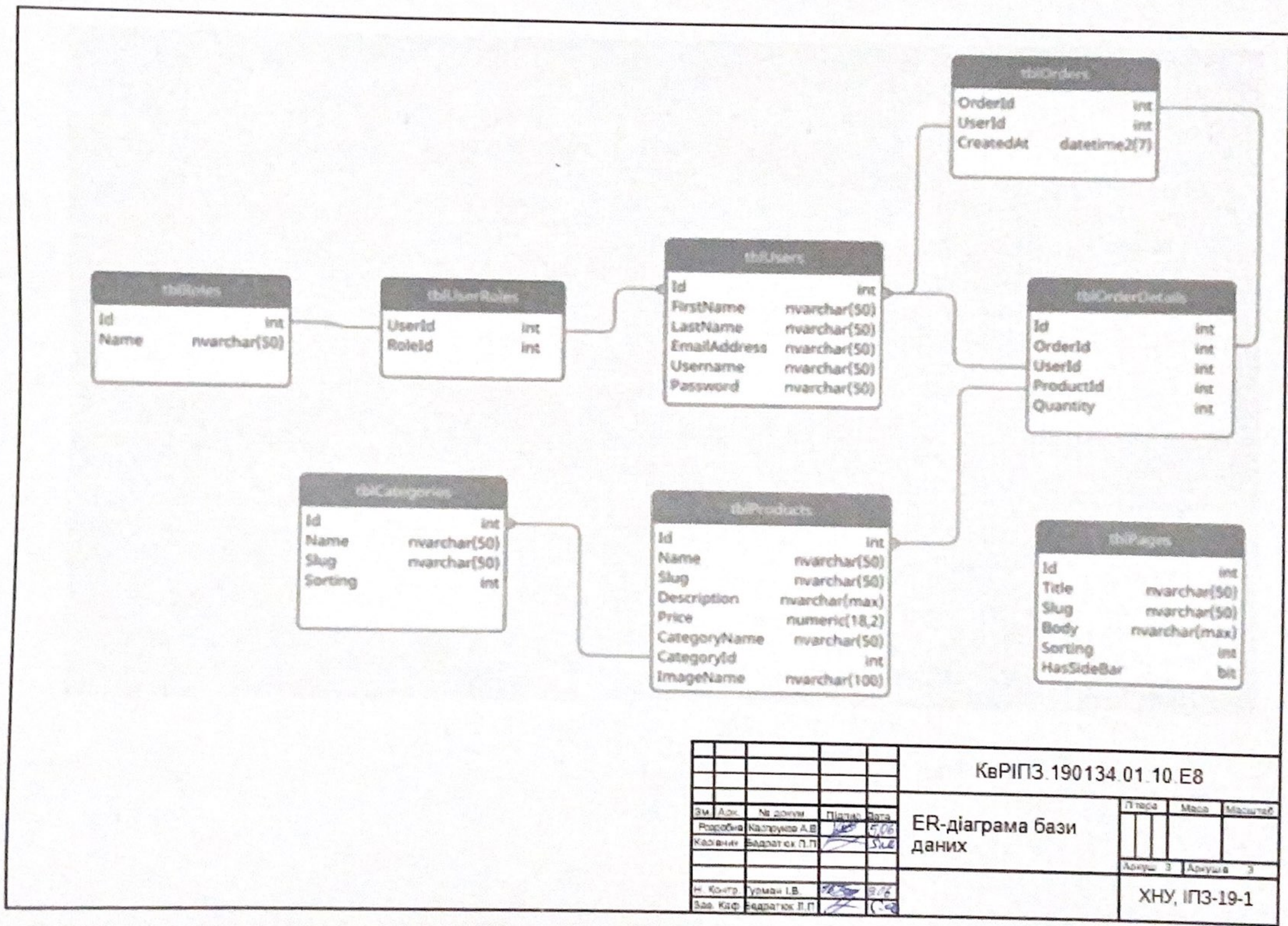
В результаті виконання кваліфікаційної роботи було успішно розроблено веб-застосунок для зручної та ефективної покупки комп'ютерної периферії. Дослідження потреб користувачів дало змогу врахувати їх очікування у розробці застосунку. Аналіз ринкових рішень та веб-технологій допоміг вибрати оптимальні рішення для розробки, відповідні сучасним стандартам. Проектування архітектури та інтерфейсу з урахуванням потреб користувачів забезпечило зручну навігацію та здійснення покупок. Тестування гарантує надійну та стабільну роботу застосунку. Отримані результати свідчать про вдале виконання завдань та придатність розробленого веб-застосунку для успішної покупки комп'ютерної периферії.





					КвРІПЗ.190134.01.10.Е8		
Вид	Док.	№ докум.	Підпис	Дата	Листів	Маса	Масштаб
Розробка	Каструнов А.В.		<i>[Signature]</i>	5.06			
Корекція	Бадрагек Л.П.		<i>[Signature]</i>	5.06			
Н. Контр.	Турман І.В.		<i>[Signature]</i>	5.06			
Зав. Каф.	Бадрагек Л.П.		<i>[Signature]</i>	5.06			
					Листів 2		Листів 3
					ХНУ, ІПЗ-19-1		

ER-діаграма бази даних



						КвРІПЗ.190134.01.10.Е8		
Зм.	Док.	№ докум.	Підпис	Дата	ER-діаграма бази даних			
Розробив	Калитченко А.В.		<i>[Signature]</i>	07.06	П'ятра	Маса	Масичка	
Коректував	Бадрачук П.П.		<i>[Signature]</i>	07.06				
Н. Контр.	Турман І.В.		<i>[Signature]</i>	07.06	Аркуш	З	Аркушів	
Зав. Каф.	Бадрачук П.П.		<i>[Signature]</i>	07.06	ХНУ, ІПЗ-19-1			

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ  
освітнього ступеня «Бакалавр»

Дипломник Каспруков Артем Віталійович

Тема Веб-застосунок для продажу комп'ютерних периферійних пристроїв

Спеціальність 121 – Інженерія програмного забезпечення

**Обсяг кваліфікаційної роботи:**

Кількість листів креслень 3 ; кількість сторінок записки 64

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі було досліджено і проаналізовано предметну область, визначено усі функціональні та нефункціональні вимоги. Був проведений аналіз існуючих програм на ринку, розглянуто їх переваги і недоліки, та доведено актуальність розробки нового програмного забезпечення. Розглянуто інструменти для реалізації спроектованих рішень, в результаті чого створено програмне забезпечення. Також було проведено тестування програми, за результатами якого доведено, що розроблене програмне забезпечення працює коректно та готове до експлуатації.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі доведено актуальність теми, визначено мету та завдання дипломного проектування. У першому розділі проведено аналіз предметної області, розглянуто існуючі рішення та визначені функціональні і нефункціональні вимоги до розроблюваного програмного забезпечення. У другому розділі проведено аналіз сучасних архітектур, розглянуто їх переваги і недоліки та визначено, що система буде відповідати монолітній архітектурі та моделі клієнт-сервер. У третьому розділі підготовлено всі залежності для написання коду та виконано практичну розробку програмних модулів і описано їх особливості, в результаті чого створено програмний продукт.

4. Позитивні сторони роботи Тематика кваліфікаційної роботи є актуальною, оскільки в сучасному цифровому світі попит на продаж таких пристроїв, відповідає потребам сучасних користувачів і бізнесів. Також було застосовано новітні технології для побудови програмного продукту та актуальні архітектурні рішення.

5. Негативні сторони роботи У роботі пошук товарів був реалізований лише за категоріями – було б доцільно додати розширений пошук.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконано відповідно до теми кваліфікаційної роботи та подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Матеріал пояснювальної записки структурований, послідовний, чіткий та простий, що дозволяє чітко зрозуміти викладений матеріал у рамках тематики проектування. Графічний матеріал дає можливість наочно побачити деталі проектування системи.

8. Інші зауваження \_\_\_\_\_

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ Говорушенко Тетяна Олександрівна, доктор технічних наук, професор, зав.кафедри комп'ютерної інженерії та інформаційних систем (КІС) ХНУ

“ 5 ” червня \_\_\_\_\_ 2023 р.

(підпис)

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатами звіту/звітів подібності щодо роботи, продукуваними програмно-технічним засобом (ами) перевірки текстів на плагіат:

Назва: «Веб-застосунок для продажу комп'ютерних периферійних пристроїв»

Автор: Каспруков Артем Віталійович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Бедратюк Леонід Петрович, д. фіз.-мат. наук, проф

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<b>відповідає</b>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) усі запозичення є фрагментарними або мають належним чином оформлені посилання;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 3,58% і адресується до 158 джерел з Інтернет і 84 джерела з Бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 2.06.2023

Завідувач кафедри



Леонід БЕДРАТЮК

Гарант освітньої програми



Леонід БЕДРАТЮК

Керівник кваліфікаційної роботи



Леонід БЕДРАТЮК

Завідувачу кафедри інженерії програмного  
забезпечення проф. Бедратюку Л. П.

здобувача вищої освіти

Каспрукова А.В.

Прізвище, ініціали

факультет ІТ, 4 курс, група ІПЗ-19-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності в Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та/або Anti-Plagiarism) і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

01.06.2023  
дата

  
підпис

## Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 9%

ID: 114528 Назва: БКР Веб-застосунок для продажу комп'ютерних периферійних пристроїв Додано в БД: 2023-06-01 Автора: Каспруков А. В. Керівники: Бедратюк Л. П. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	70699	555	1375 (2%)	24 (4%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми



Ім'я користувача:  
Кафедра ІПЗ

Дата перевірки:  
01.06.2023 22:50:39 EEST

Дата звіту:  
01.06.2023 22:51:56 EEST

ID перевірки:  
1015380211

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
100005589

Назва документа: Каспруков Кваліфікаційна робота на перевірку

Кількість сторінок: 63 Кількість слів: 10873 Кількість символів: 86572 Розмір файлу: 2.95 MB ID файлу: 1015045634

### 3.58% Схожість

Найбільша схожість: 2.74% з джерелом з Бібліотеки (ID файлу: 1015045648)

1.33% Джерела з Інтернету 158 ..... Сторінка 65

3.2% Джерела з Бібліотеки 84 ..... Сторінка 66

### 0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

### 0% Вилучень

Немає вилучених джерел