

Державний торговельно-економічний університет  
Кафедра інженерії програмного забезпечення та кібербезпеки

# ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

**«Веб-орієнтований додаток торговельного підприємства з  
продажу музичного обладнання»**

Студента 4 курсу, 7 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
освітньої програми «Інженерія  
програмного забезпечення»

Касьяненко Антона  
Сергійовича

\_\_\_\_\_

підпис студента

Науковий керівник  
кандидат педагогічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

Котенко Наталія  
Олексіївна

\_\_\_\_\_

підпис керівника

Гарант освітньої програми  
кандидат технічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

Рзаєва Світлана  
Леонідівна

\_\_\_\_\_

підпис гаранта

КИЇВ – 2023

# Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

## Затверджую

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О. В.

«14» листопада 2022 р.

## Завдання

### на випускний кваліфікаційний проєкт студентів

Касьяненко Антона Сергійовича

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Веб-орієнтований додаток  
торговельного підприємства з продажу музичного обладнання»

Затверджена наказом ректора від «6» грудня 2022 р. № 3288

2. Строк здачі студентом закінченого проєкту 5 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту полягає в розробці веб-орієнтованого додатку, що забезпечить користувачам простий та зрозумілий інтерфейс для придбання музичного обладнання з будь-якого місця з доступом до мережі Інтернет.

Об'єкт дослідження це веб-орієнтовані додатки, що використовуються торговельними підприємствами з продажу музичного обладнання.

Предметом дослідження є проектування та розробка веб-орієнтованого додатку, що дозволяє користувачеві зручно та ефективно купувати музичне обладнання через мережу Інтернет.



4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Стан реалізації музичного обладнання

1.2 Розробка та впровадження веб-орієнтованого додатку для підприємства з прод

ажу музичного обладнання

1.3. Технічне завдання

1.4. Висновки до 1 розділу

## РОЗДІЛ 2 ПРОЄКТУВАННЯ ВЕБ-ОРІЄНТОВАНОГО ДОДАТКУ ДЛЯ ПІДПРИЄМСТВА З ПРОДАЖУ МУЗИЧНОГО ОБЛАДНАННЯ

2.1. Чинники, що визначають хід і результати роботи додатку

2.2. Аналіз функціональних вимог.

2.3. Висновки до 2 розділу

## РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ-ОРІЄНТОВАНОГО ДОДАТКУ ДЛЯ ПІДПРИЄМСТВА З ПРОДАЖУ МУЗИЧНОГО ОБЛАДНАННЯ

3.1. Бази даних та їх складові

3.2 Демонстрація розробленої системи

3.3 Висновки до 3 розділу

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

## 6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2022	21.09.2022
2.	<i>Розробка та затвердження завдання на проєкт</i>	14.11.2022	14.11.2022
3.	<i>Вступ та перелік літературних джерел</i>	23.12.2022	23.12.2022
4.	<i>Розділ 1 Аналіз предметної області</i>	27.01.2023	27.01.2023
5.	<i>Розділ 2. Проєктування веб-орієнтованого додатку для підприємства з продажу музичного обладнання</i>	03.03.2023	03.03.2023
6.	<i>Розділ 3. Реалізація веб-орієнтованого додатку для підприємства з продажу музичного обладнання</i>	14.04.2023	14.04.2023
7.	<i>Висновки</i>	28.04.2023	28.04.2023
8.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	17.05.2023	17.05.2023
9.	<i>Підготовка автореферату та презентації доповіді</i>	26.05.2023	26.05.2023
10.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	29.05.2023 – 02.06.2023	
11.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	05.06.2023	05.06.2023
12.	<i>Здача прошого випускного кваліфікаційного проєкту на кафедрі</i>	05.06.2023	05.06.2023
13.	<i>Публічний захист випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проєкту \_\_\_\_\_

Котенко Н.О.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми \_\_\_\_\_

Рзаєва С.Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент \_\_\_\_\_

Касьяненко А.С.

(прізвище, ініціали, підпис)





## АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці та реалізації веб орієнтованого додатку для підприємства з продажу музичного обладнання. Метою роботи було створення зручного та ефективного інструменту для музикантів або любителів музики для швидкого та зручного придбання музичного обладнання.

В результаті порівняльного аналізу аналогічних рішень визначено основні функціональні вимоги до додатку.

Додаток було реалізовано з використанням сучасних технологій та мов програмування, що дозволило досягти високої продуктивності та надійності системи. Проведено серію тестувань, які підтвердили функціональність та якість додатку.

Готовий програмний комплекс було успішно протестовано. Отриманий веб орієнтований додаток представляє собою ефективне рішення для музикантів або любителів музики для швидкого та зручного придбання музичного обладнання.

**Ключові слова:** бд, база даних, веб-орієнтований додаток, музичне обладнання, підприємство

## ABSTRACT

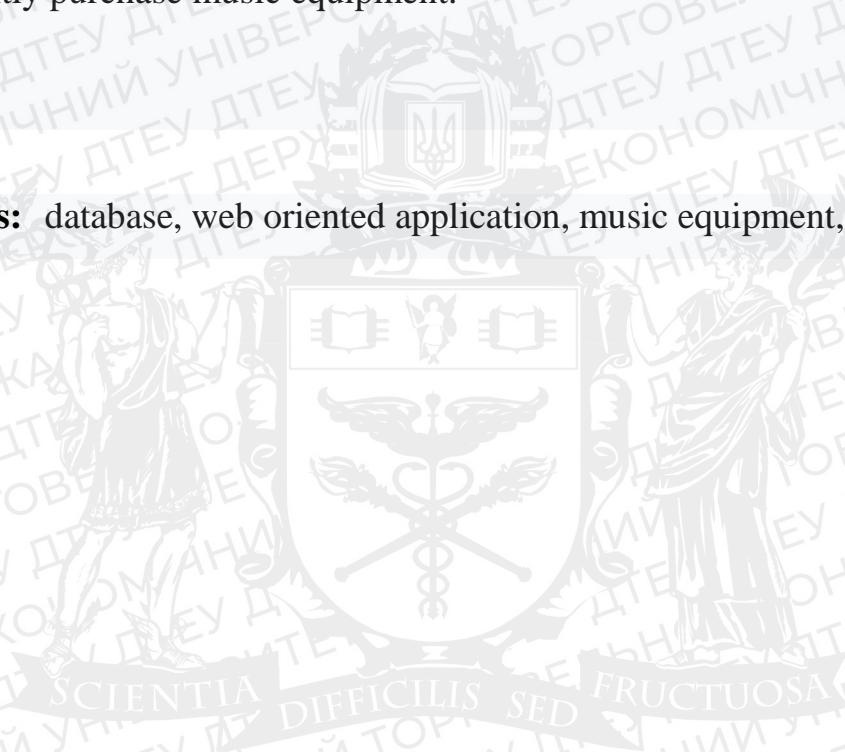
According to the purpose of the research, the work is devoted to the development and implementation of a web-oriented application for a company selling musical equipment. The goal of the work was to create a convenient and effective tool for musicians or music lovers to quickly and conveniently purchase musical equipment. As a result of a comparative analysis of similar solutions, the main functional requirements for the application were determined.



The application was implemented using modern technologies and programming languages, which made it possible to achieve high performance and reliability of the system. A series of tests were conducted that confirmed the functionality and quality of the application.

The finished software package was successfully tested. The resulting web-oriented application is an effective solution for musicians or music lovers to quickly and conveniently purchase music equipment.

**Keywords:** database, web oriented application, music equipment, enterprise



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

БД – база даних

РБД – реляційна база даних

НБД – нереляційна база даних

ВОД – веб–орієнтований додаток



<i>ДТЕУ 121 07-13.БР</i>					
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	
Зав. каф.		Криворучко О.В.		27.01.23	
Керівник		Котенко Н. О.		27.01.23	
Гарант		Рзаєва С.Л.		27.01.23	
Розробив		Касьяненко А.С		27.01.23	
Веб-орієнтований додаток торговельного підприємства з продажу музичного обладнання					
			<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
			<i>ПУП</i>	2	56
<i>Перелік умовних                  позначень, символів, од                  иниць скорочень і                  термінів</i>					
<i>Факультет інформаційних                  технологій                  4 курс, 7 група</i>					



## ЗМІСТ

<b>ВСТУП</b> .....	3
<b>РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	6
<b>1.1. Стан реалізації музичного обладнання</b> .....	6
<b>1.2 Розробка та впровадження веб–орієнтованого додатку для підприємства з продажу музичного обладнання</b> .....	13
<b>1.3. Технічне завдання</b> .....	16
<b>1.4. Висновки до 1 розділу</b> .....	22
<b>РОЗДІЛ 2 ПРОЄКТУВАННЯ ВЕБ–ОРІЄНТОВАНОГО ДОДАТКУ ДЛЯ ПІДПРИЄМСТВА З ПРОДАЖУ МУЗИЧНОГО ОБЛАДНАННЯ</b> .....	24
<b>2.1. Чинники, що визначають хід і результати роботи додатку</b> ..	24
<b>2.2. Аналіз функціональних вимог.</b> .....	33
<b>2.3 Висновки до 2 розділу</b> .....	38
<b>РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ–ОРІЄНТОВАНОГО ДОДАТКУ ДЛЯ ПІДПРИЄМСТВА З ПРОДАЖУ МУЗИЧНОГО ОБЛАДНАННЯ</b> .....	25
<b>3.1 Бази даних та їх складові</b> .....	25
<b>3.2. Впровадження розробленої системи</b> .....	45
<b>3.3 Висновки до 3 розділу</b> .....	49
<b>ВИСНОВКИ ТА ПРОПОЗИЦІЇ</b> .....	42
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	50
<b>ДОДАТКИ</b> .....	54

					<i>ДТЕУ 121 07-13.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Веб-орієнтований додаток торговельного підприємства з продажу музичного обладнання	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		3	3	56
Керівник		Котенко Н. О.		23.12.22		Факультет інформаційних технологій 4 курс, 7 група		
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Касьяненко А.С		23.12.22	<i>Зміст</i>			

## ВСТУП

*Актуальність.* Онлайн магазини зараз невід’ємна частина нашого життя, в нинішній час більша частина покупок здійснюється онлайн, онлайн магазини доступні двадцять чотири години на добу, в будь який момент де б ви не знаходились, за наявності підключення до інтернету ви можете зайти на веб сторінку та замовити той чи інший продукт

Існує багато видів розширення своїх клієнтів, такі як сторінки в соціальних мережах, сайти та додатки, на яких продають товари чи послуги. Це дуже зручно тому що завдяки цьому підприємству не потрібно відкривати фізичні магазини, платити за оренду та працівників, набагато зручніше мати онлайн магазин з доставкою по всій країні.

Веб-орієнтований додаток для онлайн продажу є одним з найважливіших місць розповсюдження товарів тому, що на власному сайті не буде конкурентів, які присутні в соціальних мережах і він не буде займати місце на гаджеті. Через ці причини для продажу музичного обладнання через мережу інтернет, було прийняте рішення створити веб-сайт.

Виходячи з вищесказаного було обрано тему: «Веб-орієнтований додаток торговельного підприємства з продажу музичного обладнання»

*Мета проекту* полягає в розробці веб-орієнтованого додатку, що забезпечить користувачам простий та зрозумілий інтерфейс для придбання музичного обладнання з будь-якого місця з доступом до мережі Інтернет.

*Об’єкт дослідження* це веб-орієнтовані додатки, що використовуються торговельними підприємствами з продажу музичного обладнання.

					<i>ДТЕУ 121 07-13.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Веб-орієнтований додаток торговельного підприємства з продажу музичного обладнання	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		В	4	57
Керівник		Котенко Н. О.		23.12.22		Факультет інформаційних технологій 4 курс, 7 група		
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Касьяненко А.С		23.12.22				
					<i>Вступ</i>			



*Предметом дослідження є проектування та розробка веб-орієнтованого додатку, що дозволяє користувачеві зручно та ефективно купувати музичне обладнання через мережу Інтернет.*

*Пошуки шляхів досягнення мети обумовили необхідність визначення таких завдань:*

- дослідити та проаналізувати підходи до створення веб-орієнтованих додатків*
- розробити веб-орієнтовані платформи для додатку*
- спроектувати користувацький інтерфейс веб-орієнтованого додатку для онлайн продажу, каталог товарів, кошик, базу даних;*
- додаток надає змогу користувачу та потенційному покупцю придбати музичне обладнання, вибрати його, дізнатися характеристики, ціну, додати до кошика та здійснити транзакцію*

*Інтернет-магазин буде корисним для розвитку підприємства, сприятиме залученню нових клієнтів і цим має збільшити прибутки.*

*У базі даних буде зберігатися опис та фотографії товару. Також в ній буде міститися інформація про клієнтів, які надали свою електронну пошту.*

						Аркуш
					ДТЕУ 121 07-13.БР	5
Зм.	Аркуш	№ докум	Підпис	Дата		

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 1.1. Стан реалізації музичного обладнання

Купівля музичного обладнання онлайн має широкий спектр переваг

Широкий вибір товарів, онлайн магазини музичного обладнання зазвичай мають широкий вибір товарів, включаючи різні бренди, моделі та типи обладнання. Можна легко порівняти характеристики та ціни на різних сайтах та вибрати той варіант, який найбільше відповідає потребам та бюджету.

Економія часу та грошей, замовляючи музичне обладнання онлайн, можна заощадити час та гроші на поїздках до магазинів, а також використовувати можливість знайти кращі пропозиції, знижки та акції в інтернет-магазинах.

Легкість замовлення, онлайн-магазини музичного обладнання зазвичай мають простий та зрозумілий процес замовлення, який дає можливість легко та швидко придбати потрібний товар

Також варто зазначити, що музичне обладнання стає все більш доступним для широкої аудиторії, що дозволяє користувачам з легкістю починати займатися новим хоббі, а може й сенсом свого життя

Еталонна модель сайту – це загальна структура та зовнішній вигляд сайту, яка є стандартом або прикладом для інших веб-сайтів. Це може бути зручно як для розробників, так і для користувачів, оскільки вона надає загальний опис того, як повинен виглядати та працювати сайт. (Рис 1.1)

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 07-13.БР			
Зав. каф.		Криворучко О.В.		27.01.23	Веб-орієнтований додаток торговельного підприємства з продажу музичного обладнання	Стадія	Аркуш	Аркушів
Керівник		Котенко Н. О.		27.01.23		РІ	6	57
Гарант		Рзаєва С.Л.		27.01.23		Факультет інформаційних технологій		
Розробив		Касьяненко А.С		27.01.23		4 курс, 7 група		



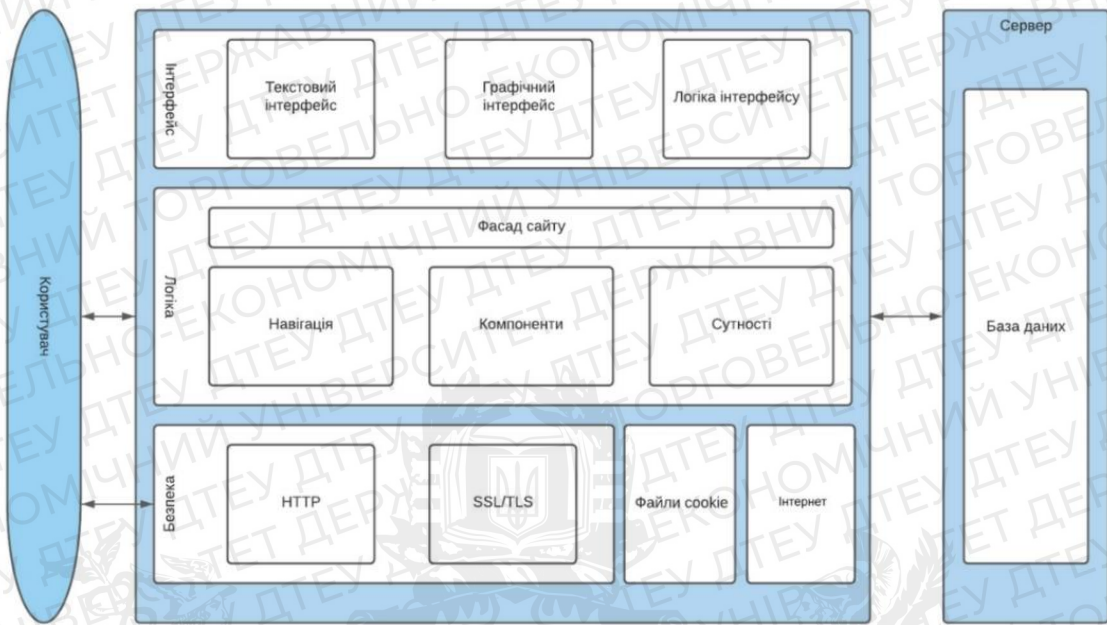


Рисунок 1.1 – Еталонна модель сайту

Джерело: побудовано автором

У еталонній моделі зображена взаємодія між користувачем та базою даних, що знаходиться на сервері.

Для внесення інформації в базу даних користувачем, необхідно розробити зрозумілий текстовий та графічний інтерфейс, що буде логічно пов'язаний.

Основні елементи еталонної моделі сайту можуть включати наступне:

**Шапка (Header):** включає логотип, головне меню, контактну інформацію та інші корисні посилання.

**Тіло (Body):** це основний вміст сайту, включаючи текст, зображення, відео, таблиці та інші елементи.

**Бічна панель (Sidebar):** зазвичай містить меню, яке допомагає користувачам знайти необхідну інформацію, пошук, календар, блоки з соціальними мережами та інші корисні функції.

Підвал (Footer): включає посилання на додаткову інформацію про сайт, контактну інформацію, авторські права та інші корисні посилання.

Діаграма варіантів використання – це найпростіша з поведінкових діаграм UML, та найкорисніша при висвітленні функціональних особливостей програми для людей, що не мають глибоких знань в ІТ галузі. Наприклад, замовників софта. Ця діаграма використовується для опису цілей, які переслідує користувач розробленої програми, будь це або людина, або інша програма. Говорячи строго, діаграма варіантів використання застосовується для опису функціональних вимог до програми, або її підсистем, або іншого об'єкта. Вона представляє загальну картину того, як програма буде використовуватись.

На діаграмі можна побачити, що існує 2 користувача сайту, і кожен має свої варіанти використання. Згідно їхніх ролей система буде надавати їм відповідний функціонал. Функціональні можливості продукту будуть відображені на діаграмі варіантів використання зображених на рисунку 2. Так клієнт може переглядати інформацію на сайті, додавати вибрані товари в кошик і оформляти замовлення. В обов'язки адміністратора входить: зв'язок з клієнтом, перевірка оплати і контроль виконання замовлення, оновлення цін і новин, перевірка актуальності інформації.

Діаграма кооперацій показує як саме відбувається оформлення замовлення на сайті. Діаграма кооперацій Поняття кооперації є одним з фундаментальних понять у мові UML. Воно служить для позначення безлічі взаємодіючих з певною метою об'єктів в загальному контексті модельованої системи.

Мета самої кооперації полягає в тому, щоб специфікувати особливості реалізації окремих найбільш значущих операцій в системі. Кооперація визначає структуру поведінки системи в термінах взаємодії учасників цієї кооперації.

						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			8



Діаграма кооперації насамперед відображає структуру взаємодії та містить такі елементи:

1. Екземпляри акторів і класів, що беруть участь в реалізації варіанту використання;
2. Асоціацію між екземплярами акторів і класів;
3. Повідомлення, що передаються між екземплярами акторів і класів.

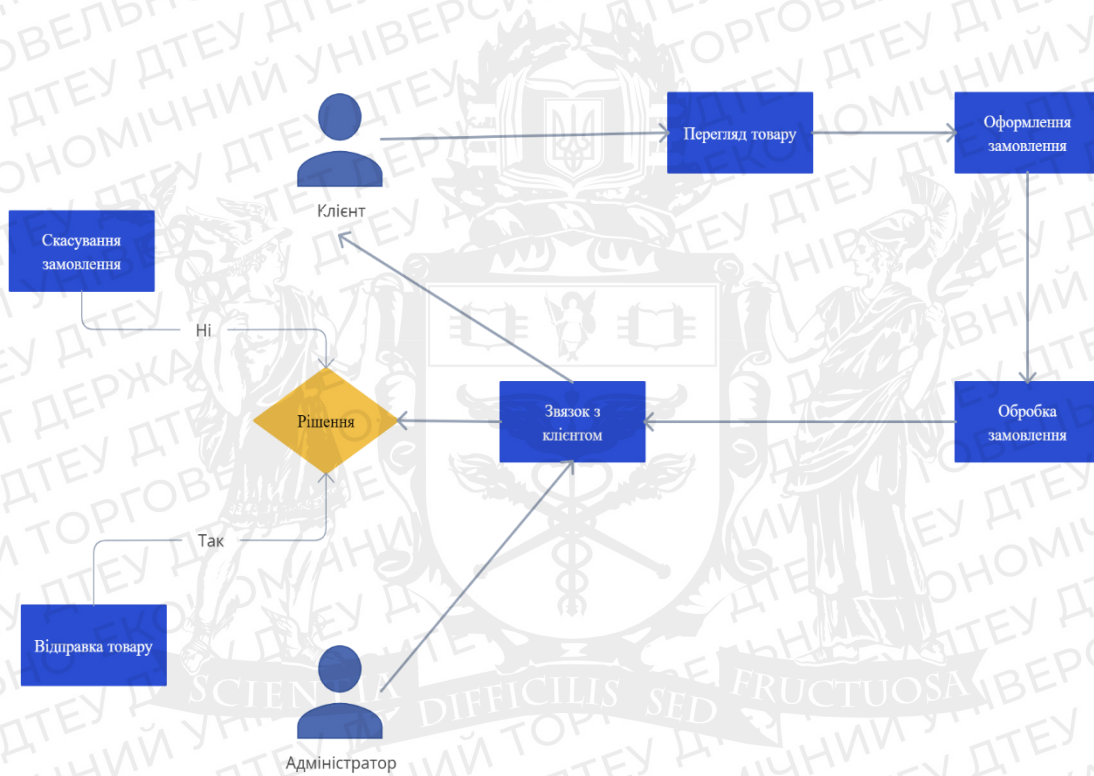


Рисунок 1.2 – Діаграма кооперації

Джерело: побудовано автором

Клієнт переглядає товари на сайті, потім оформляє замовлення яке зберігається до бази даних. Адміністратор компанії зв'язується з клієнтом і в залежності від отриманої відповіді від клієнта відправляє товар (Рис 1.3).

Клієнт – це особа або організація яка користується послугами або купує товари від певної компанії або постачальника. Клієнти можуть бути різні за своїми потребами, бюджетом та очікуваннями від продукту чи послуги, які

вони отримують.

Адміністратор – це людина або команда людей, яка відповідає за управління та підтримку різних аспектів діяльності організації, зокрема інформаційних систем, комп'ютерних мереж, баз даних, апаратного та програмного забезпечення, а також організацію робочих місць та документообігу.

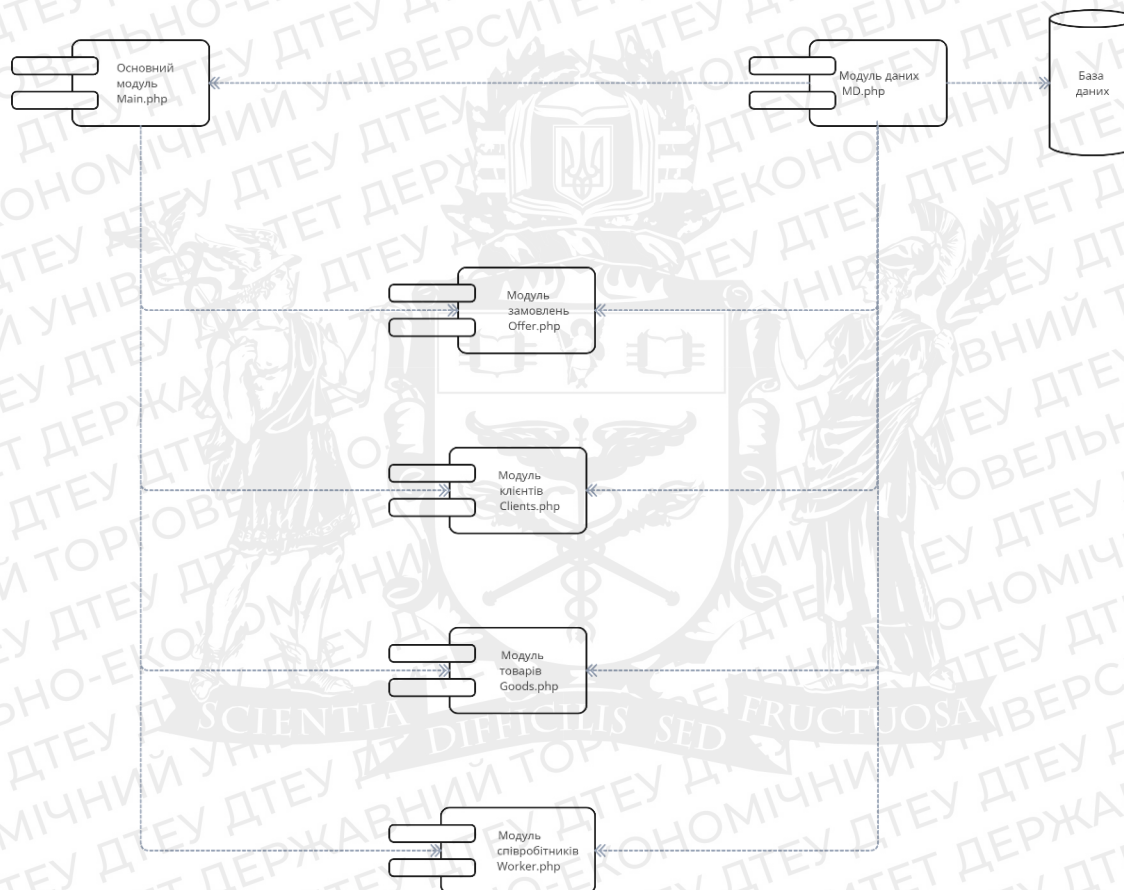


Рисунок 1.3 – Діаграма компонентів інтернет-магазину

Джерело: побудовано автором

Діаграма класів (Рис 1.4) – це діаграма, яка показує структуру класів та їх взаємодії в системі. Вона використовується в об'єктно-орієнтованому програмуванні для моделювання та проектування програмних систем.

Елементи діаграм класів пов'язані різними структурними зв'язками. Такі діаграми використовують для проектування словника системи чи

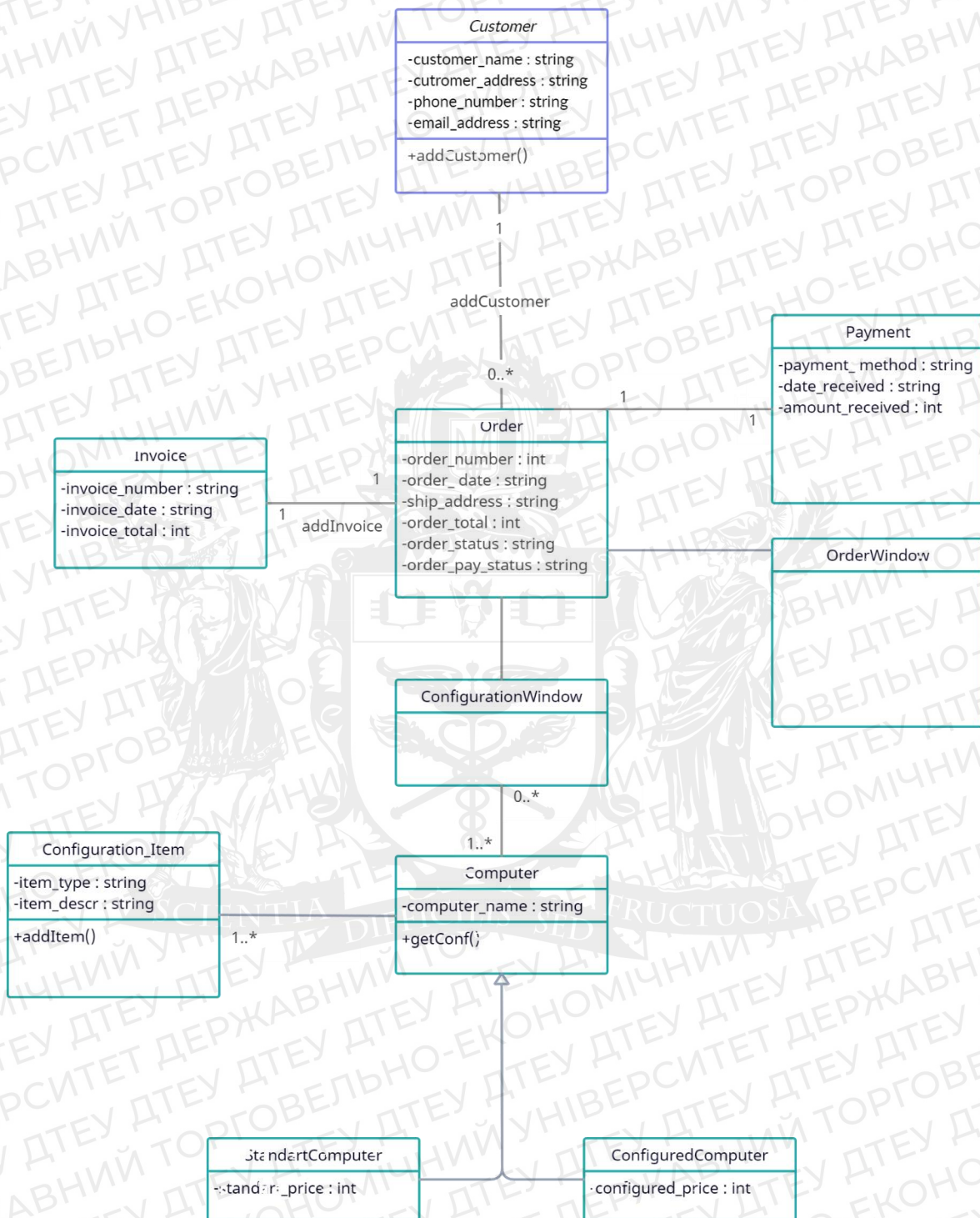


кооперацій та систем. Діаграми класів мовою UML використовують для того, щоб показати параметри блоків цих діаграм та їх зв'язків. На діаграмах класів зазвичай представлені такі елементи:

1. Класи;
2. Інтерфейси;
3. Залежності, узагальнення та асоціації.

Та інших елементів, що допомагають відображати зв'язки між класами. Кожен клас може мати атрибути та методи, які відображають його властивості та поведінку. Наприклад, діаграма класів може показати зв'язок між класами "Користувач", "Товар" та "Кошик", що дозволяє відобразити, які методи та атрибути повинні бути в кожному класі.

Також діаграми можуть включати в себе пакети або підсистеми; ті та інші групують елементи моделі у більш великі утворення. Іноді в діаграму класів потрібно додати примірники, особливо якщо є необхідність показати тип примірника. Створюючи діаграми класів UML потрібно пам'ятати, що кожна із них – це лише графічне зображення статичного дизайну системи. Жодна діаграма класів не зобов'язана включати все, що стосується дизайну системи. Але вона надає користувачеві повну інформацію, необхідну для статичного представлення системи; хоча кожна з них представляє лише один її аспект.



*Рисунок 1.4 – Діаграма класів  
Джерел: побудовано автором*



## 1.2 Розробка та впровадження веб-орієнтованого додатку для підприємства з продажу музичного обладнання

У сучасному обчислювальному світі веб-орієнтовані додатки стали популярною, оскільки вони використовуються практично щодня для різних додатків. При розробці веб-орієнтованого додатка, який протоколізує події, ви можете використовувати певні інструменти та підходи для збереження журналу подій.

Логування на рівні сервера: Для цього можна використовувати фреймворки або бібліотеки, що дозволяють логувати події на рівні сервера. Наприклад, для додатків на основі Django можна використовувати модуль logging для реєстрації подій у файлі журналу або іншому сховищі (наприклад, базі даних)

Системи централізованого журналювання: Можна використовувати системи централізованого журналювання, такі як ELK Stack (Elasticsearch, Logstash, Kibana), Splunk або Graylog. Ці системи дозволяють збирати, аналізувати та візуалізувати журнал подій з різних джерел

Сервіси аналізу журналів: Існують сервіси, які спеціалізуються на аналізі журналів та моніторингу подій, такі як Loggly, Papertrail, Logentries тощо. Вони надають інструменти для збору та аналізу журналів, дозволяють створювати звіти, виявляти проблеми та здійснювати моніторинг

Власний механізм запису подій: Також можливо розробити власний механізм запису подій, використовуючи бази даних, файлову систему або інші зберігаючі механізми. Це може включати створення таблиць або колекцій для зберігання подій, реалізацію логіки запису та читання подій, індексування для швидкого пошуку. Переваги та недоліки веб-додатків:

Переваги: Доступність: Веб-додатки можуть бути доступні з будь-якого місця, де є підключення до Інтернету. Користувачі можуть отримувати доступ до додатку з різних пристроїв, таких як комп'ютери, смартфони або планшети.

						Аркуш
					ДТЕУ 121 07-13.БР	13
Зм.	Аркуш	№ докум	Підпис	Дата		

**Платформна незалежність:** Веб–додатки працюють у веб–браузері, що робить їх незалежними від операційної системи. Можна використовувати веб–додатки на Windows, Mac, Linux або будь–якій іншій операційній системі з підтримкою веб–браузера.

**Легкість розгортання та оновлення:** Веб–додатки дозволяють швидко розгорнути та оновити нову версію без необхідності встановлення на кожен клієнтський пристрій. Це полегшує процес розповсюдження і підтримки додатку.

**Можливість обробки даних, незважаючи на місцезнаходження.** Користувачі можуть безпосередньо входити в систему, незважаючи на розташування або технологію процесорів.

**Синхронізація даних:** Завдяки веб–додаткам, дані можуть бути збережені на сервері і доступні для синхронізації на різних пристроях. Користувачі можуть отримувати оновлені дані на всіх своїх пристроях без необхідності вручну копіювати або передавати файли.

**Безпека.** Сервери мають кращий контроль доступу та ресурсів, щоб гарантувати, що лише авторизовані клієнти можуть отримувати доступ до даних та керувати ними.

**Недоліки:** Перевантажені сервери. Коли на сервер надходять часті одночасні клієнтські запити, сервер сильно перевантажується, утворюючи перевантаження трафіку.

**Залежність від Інтернету:** Веб–додатки вимагають постійного підключення до Інтернету для роботи. Якщо немає доступу до Інтернету або має місце низька швидкість зв'язку, користувачі можуть мати проблеми з використанням додатку

**Залежність від браузера:** Веб–додатки побудовані на веб–технологіях, і їх поведінка може відрізнитися в різних веб–браузерах. Це може призводити до проблем з сумісністю та потребує тестування на різних браузерах

						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			14



Вразливість до кібератак: Веб-додатки відкриті для зовнішнього доступу через Інтернет, що робить їх потенційно вразливими до кібератак та зламу. Забезпечення безпеки веб-додатків вимагає уваги до забезпечення валідності введених даних, захисту від SQL-ін'єкцій, перехоплення сесій тощо

Правильний вибір технологій дозволяє забезпечити ефективну та швидку розробку Для розробки клієнтської частини можна використовувати мови програмування, Java, C++, C#, Python, JavaScript тощо

Для розробки серверної частини можна використовувати бази даних, такі як MySQL, PostgreSQL, MongoDB, а також веб-сервери

Для забезпечення комунікації між частинами використовуватимережеві протоколи, HTTP, WebSocket, TCP/IP, UDP тощо.

Розробка серверної частини додатку передбачає створення програмного забезпечення, для розробки серверної частини можна використовувати різні технології, такі як фреймворки, бібліотеки, мови програмування, один з таких фреймворків для розробки серверної частини є Django, він дає зручний інтерфейс для роботи з базами даних, також є фреймворк Flask, він простіший за Django та з його використанням додатки робляться швидше.

Розробка та впровадження додатку для продажу музичного обладнання може бути доволі складна.

Визначення функціональності: Визначити які функції включити в додаток. Наприклад, це може включати перегляд каталогу обладнання, пошук товарів, додавання їх до кошика, оформлення замовлення, сплату, відстеження доставки

Розробка користувацького інтерфейсу: Створення зручного та привабливого інтерфейсу для взаємодії з користувачами. Використовуються відповідні технології, такі як мобільні фреймворки або веб-інтерфейси, в залежності від цільової платформи

Інтеграція платіжних систем: Розглядання використання платіжних

						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			15

систем, щоб дозволити користувачам здійснювати оплату за обладнання. Наприклад, інтегрувати платіжні шлюзи, такі як PayPal для безпечної обробки транзакцій.

Розробка системи управління замовленнями: Створення систему, яка дозволяє керувати замовленнями, включаючи обробку замовлень, відстеження доставки та сповіщення користувачів про стан їх замовлення.

Безпека та захист даних: Забезпечення належного рівня безпеки для додатку, включаючи захист особистої інформації користувачів та обробку платежів

Тестування та впровадження: Перевірення додатку на помилки та забезпечення його надійність перед впровадженням у роботу.

### 1.3. Технічне завдання

#### 1. Загальні відомості

##### 1.1 Веб-орієнтований додаток торговельного підприємства

###### 1.1.1 Веб-орієнтований додаток торговельного підприємства

###### 1.1.2 Веб-сайт «Musica»

##### 1.2 Планові терміни початку та закінчення робіт

Початок робіт – 01.01.2023

Закінчення робіт – 01.05.202

1.3 Роботи зі створення веб-додатку здаються розробником поетапно відповідно до календарного плану проекту та будуть оформлені у вигляді програмного коду, документації з поясненням алгоритмів, принципів роботи та інструкцій щодо встановлення, налаштування та користування додатком. Після закінчення кожного з етапів робіт розробник здає замовнику відповідні звітні документи етапу, склад яких визначено договором.

1.4 Головним бенефіціаром являється компанія «Musica». Потенційні користувачі системи якої являються музичні колективи, а також звичайні

						Аркуш
						16
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	



споживачі

## 2. Мета та призначення створення системи

### 2.1 Призначення системи

Веб орієнтований додаток призначений для продажу музичного обладнання

2.2 Метою створення системи є полегшення пошуку, забезпечення максимальної точності вибору та купівлю музичного обладнання, додаток повинен бути зручним та простим у користуванні

2.3 Розробити сайт який полегшить пошук музичного обладнання, його купівлю, оформлення замовлення.

### 3. Вимоги до системи

3.1 Інтуїтивний інтерфейс, адаптація під мобільні пристрої, висока швидкодія, інтеграція з Excel

3.1.1 Вимоги до структури та функціонування системи, перелік підсистем

Підсистема модель являє собою сукупність класів які реалізують архітектуру бази даних з усіма зв'язками в ній між усіма сутностями.

Підсистема контролер реалізує серверну частину додатка яка обробляє всі запити від користувачів до бази даних.

Підсистема вид відмальовує графічний інтерфейс сайту на основі отриманої з БД інформації.

3.1.1.1 Вимоги до способів і засобів інформаційного обміну між компонентами системи

Для обміну інформацією між користувачем і базою даних використовуються HTML запити до контролерів системи які, обробляючи їх, видають необхідну користувачу інформацію – ця технологія має назву Rest API. Для цього можуть використовуватися протоколи забезпечення безпеки та стандарти взаємодії, такі як HTTP, HTTPS, TCP/IP. Для передачі даних можуть

						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			17

використовуватися формати JSON або XML

### 3.1.1.2 Вимоги до режимів функціонування системи

Система працює в режимі запит–відповідь у режимі 24/7, без перебоїв та забезпечувати швидку відповідь на запити користувачів. Користувач, гортаючи сторінки сайту, весь час надсилає запити до серверної частини додатка.

### 3.1.1.3 Вимоги до діагностування системи

До вимог діагностування системи можна віднести діагностування на швидкодію системи, діагностування на предмет помилок в системі, діагностування безпеки системи що виникають у роботі системи

### 3.1.1.4 Вимоги до режимів управління системою

Система має зручний та простий інтерфейс управління. Режим роботи системи полягає в принципі запит відповідь. Користувач надсилає запит, а додаток надає відповідь у вигляді інформації яку хоче отримати користувач.

## 3.1.2. Показники призначення

3.1.2.1 Параметри, що характеризують ступінь відповідності системи призначенням

### 3.1.2.2 Вимоги до пристосовності системи до змін

Система може працювати на різних хостингах під операційною системою Windows Server. Адаптована під СКБД Microsoft SQL Server для збільшення швидкодії системи. Базу даних можна переносити на любий сервер, а також робити резервні копії. можна додавати нові функції та модулі до системи без необхідності її повного перепроєктування,

3.1.2.3 Вимоги до збереження працездатності системи в різних ймовірних умовах

Для забезпечення працездатності використовується Хостинг, через що працездатність системи гарантована для різних ймовірних умов та

						Аркуш
						18
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	



мінімальний час, необхідний для відновлення роботи системи після збою, можна працювати в різних операційних системах та на різних пристроях

### 3.1.3. Вимоги до надійності

#### 3.1.3.1. Склад показників надійності до системи в цілому

До показників надійності відносяться: стабільність роботи веб-додатка, швидкодія, відмовостійкість, відсутність помилок в роботі системи, частота відмов системи, час відновлення системи після відмови.

#### 3.1.3.2. Вимоги до надійності технічних засобів і програмного забезпечення

Для надійності веб-додатка було обрано технологію ASP.NET яка витримує велике навантаження і велику кількість запитів, надійність програмного забезпечення є високою і забезпечувати нормальну роботу системи без відмов. База даних на MS SQL SERVER яку легко адмініструвати і оновлювати у разі потреби та яка має велику швидкодію.

#### 3.1.3.3. Вимоги до методів оцінки і контролю показників надійності на різних стадіях створення системи

#### 3.1.4. Вимоги до ергономіки та технічної естетики

Додаток має зручний та логічно побудований інтерфейс для користувача, це дозволяє легко використовувати усі функції. Гамма кольорів зручна та приємна оку, не викликає дискомфорту у користувача.

#### 3.1.5. Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Додаток має простий та зрозумілий інтерфейс для розуміння користувачем процесів обслуговування та ремонту, для зберігання даних додаток має можливість резервного копіювання та відновлення даних, додаток працює на різних платформах та оперативних системах і має можливість оновлення програмного забезпечення з мінімальними зусиллями.

#### 3.1.6 Вимоги до захисту інформації від несанкціонованого доступу

						Аркуш
						19
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	

### 3.1.6.1. Вимоги до інформаційної безпеки

В додатку використовується протокол HTTPS та шифрування даних. Паролі перед зберіганням у базі даних шифруються на стороні сервера, система забезпечує конфіденційність, цілісність та доступність даних, Система має механізми аутентифікації та авторизації користувачів, система забезпечує моніторинг захисту інформації, виявлення та реагування на можливі інциденти.

### 3.1.6.2. Вимоги до антивірусного захисту

Система має антивірусне програмне забезпечення, воно забезпечує захист від шкідливих програм, антивірусне програмне забезпечення оновлюється регулярно.

### 3.1.6.3. Розмежування відповідальності ролей при доступі

На сайті використовується рольова модель доступу. Для кожної ролі, яка визначається адміністратором, визначаються конкретні права доступу до певних розділів сайту. Так наприклад користувач може лише переглядати інформацію на сайті і оформляти замовлення, менеджер обробляти замовлення користувачів, а адміністратор оновлювати прайси на сайті.

### 3.1.7. Вимоги до захисту від впливу зовнішніх факторів

Хостинг на якому знаходиться сайт представляє собою датацентр з великою кількістю серверів і резервних копій. Це дозволяє мінімізувати вплив зовнішніх факторів. Додаток захищений від випадкових або зловмисних дій користувачів, вірусів, троянських програм, шкідливого ПЗ, Додаток має можливість автоматично виявляти та ліквідувати загрози.

### 3.1.8. Вимоги безпеки

Додаток має можливість резервного копіювання даних та відновлення роботи системи в разі аварійної ситуації, забезпечує безпеку користувачів, безпеку підключення до системи, безпеку вводу та збереження конфіденційної інформації.

						ДТЕУ 121 07-13.БР	Аркуш
							20
Зм.	Аркуш	№ докум	Підпис	Дата			



### 3.3.2. Вимоги до інформаційного забезпечення

Для роботи системи від хостинга потребується система Windows. Система забезпечує безпечний обмін даними між клієнтом та сервером, забезпечує авторизацію користувачів, Система має можливість здійснювати архівування та резервне копіювання даних.

3.3.2.1. Вимоги до складу, структури і способів організації даних в системі

Структура організації даних є діаграма бази даних яка складається з таблиць які пов'язані між собою ключами. База даних нормалізована.

3.3.2.2. Вимоги до інформаційного обміну між компонентами системи

Для обміну інформацією використовується технологія HTTP запитів яка називається REST API. Коли користувач звертається до якоїсь сторінки веб-додатка надсилається запит до контролера який обробляє його і надає інформацію з БД.

3.3.2.5. Вимоги щодо застосування систем управління базами даних

В ролі системи керування баз даних вибрана база даних MS SQL SERVER 2019

4. Вимоги до програмного забезпечення

Додаток написаний на мові програмування, додаток має Для роботи системи потрібно абсолютно люба операційна система з встановленим браузером.

5. Вимоги до технічного забезпечення

Для роботи системи підійде абсолютно любий сучасний комп'ютер, або смартфон з доступом до інтернету.

6. Вимоги до методичного забезпечення

Документація до додатку повинна бути доступною для користувачів в електронному вигляді та містити докладний опис функціоналу та принципів роботи.

						Аркуш
						21
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	

#### 1.4. Висновки до 1 розділу

Онлайн магазини зараз невід’ємна частина нашого життя, в нинішній час більша частина покупок здійснюється онлайн, онлайн магазини доступні двадцять чотири години на добу, в будь який момент в будь якому місті, за наявності підключення до інтернету можна зайти на веб сторінку та замовити той чи інший продукт

Існує багато видів розширення клієнтів, такі як сторінки в соціальних мережах, сайти та додатки, на яких продають товари чи послуги. Це дуже зручно тому що завдяки цьому підприємству не потрібно відкривати фізичні магазини, платити за оренду та працівників, набагато зручніше мати онлайн магазин з доставкою по всій країні.

Веб-сторінка є одним з найважливіших місць розповсюдження товарів тому, що на власному сайті не буде конкурентів, які присутні в соціальних мережах і він не буде займати місце на гаджеті. Через ці причини для продажу музичного обладнання через мережу інтернет, було прийняте рішення створити веб-сайт

Основні етапи розробки та впровадження такого додатку можуть включати наступне: визначення вимог до додатку, вибір технологій, розробка серверної частини, розробка клієнтської частини, тестування та налагодження, впровадження, підтримка та доповнення.

Розробка та впровадження додатку для продажу музичного обладнання може бути складним та часовим процесом. Проте, якщо мати необхідний досвід та знання, а також використовувати ефективні інструменти, то можливо розробити потрібний додаток, який відповідатиме всім вимогам

						Аркуш
					ДТЕУ 121 07-13.БР	22
Зм.	Аркуш	№ докум	Підпис	Дата		



## РОЗДІЛ 2

### ПРОЄКТУВАННЯ ВЕБ-ОРІЄНТОВАНОГО ДОДАТКУ ДЛЯ ПІДПРИЄМСТВА З ПРОДАЖУ МУЗИЧНОГО ОБЛАДНАННЯ

#### 2.1. Чинники, що визначають хід і результати роботи додатку

Коли мова йде про веб орієнтований додаток з продажу музичного обладнання, існує декілька специфічних чинників, які можуть визначати хід і результати його роботи.

Забезпечення високої якості програмного забезпечення вимагає правильного планування, використання тестування, виявлення й виправлення помилок та постійного вдосконалення. Якість програмного забезпечення (ПЗ) визначається його здатністю виконувати свої функції, задовольняти вимоги користувачів та виявляти мінімум дефектів або помилок. Якість ПЗ впливає на його надійність, продуктивність, безпеку та задоволення користувачів.

**Функціональність:** ПЗ повинно відповідати вимогам та функціональним специфікаціям. Воно повинно виконувати необхідні операції та функції згідно з очікуваннями користувачів.

**Надійність:** Надійність вказує на стійкість ПЗ до відмов та його здатність функціонувати безперебійно протягом тривалого часу. ПЗ повинно бути стійким до помилок, витоків пам'яті або некоректної роботи, що може призвести до аварій або втрати даних.

**Ефективність:** Ефективність відноситься до продуктивності ПЗ та його здатності до ефективного використання ресурсів, таких як обсяг пам'яті, потужність процесора та мережевий трафік ПЗ повинно працювати швидко,

					<i>ДТЕУ 121 07-13.БР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		03.03.23	Веб-орієнтований додаток торговельного підприємства з продажу музичного обладнання	Стадія	Аркуш	Аркушів
Керівник		Котенко Н. О.		03.03.23		P2	24	57
Гарант		Рзаєва С.Л.		03.03.23		Факультет інформаційних технологій 4 курс, 7 група		
Розробив		Касьяненко А.С		03.03.23				

відповідати на запити користувачів вчасно та забезпечувати оптимальне використання ресурсів.

**Безпека:** Безпека відноситься до захисту ПЗ від несанкціонованого доступу, злому та витоку даних. Воно повинно мати відповідні механізми для шифрування даних, перевірки автентичності, контролю доступу та інших заходів для забезпечення конфіденційності та цілісності інформації.

**Легкість супроводу:** ПЗ повинно бути легким у супроводі та підтримці. Це означає, що його код повинен бути читабельним, документованим та добре організованим. Легкість супроводу полегшує виявлення та виправлення помилок, розширення функціональності та підтримку ПЗ протягом його життєвого циклу.

**Використовуваність:** ПЗ повинно бути легким у використанні для кінцевих користувачів. Інтерфейс користувача повинен бути зрозумілим, інтуїтивним та забезпечувати зручну навігацію. Використовуваність включає також наявність документації та підтримки для користувачів.

**Продуктивність програмного забезпечення** визначається його здатністю працювати ефективно та швидко відповідно до вимог та очікувань користувачів. Вона може бути виміряна різними метриками, такими як час відгуку, швидкість обробки даних, завантаження системи та інші. Загалом, продуктивність програмного забезпечення є важливим аспектом, особливо для додатків з великим обсягом обробки даних або високим рівнем взаємодії з користувачем. Оптимізація, використання ефективних алгоритмів та ресурсів, а також тестування допоможуть забезпечити високу продуктивність програмного забезпечення. Основні фактори, які впливають на продуктивність програмного забезпечення:

**Алгоритми та оптимізація:** Ефективний вибір алгоритмів та їх оптимізація можуть позитивно впливати на продуктивність програмного забезпечення. Це означає вибір швидких та ефективних алгоритмів обробки

						Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	



даних, оптимізацію запитів до бази даних, раціональне використання ресурсів та інші підходи, що сприяють прискоренню роботи програми.

**Архітектура програмного забезпечення:** Добре спроектована архітектура програмного забезпечення може позитивно впливати на продуктивність. Чітко визначена структура, розподілення функцій та відповідності між компонентами можуть забезпечити ефективну обробку даних та оптимальне використання ресурсів системи.

**Оптимізація бази даних:** Якщо програмне забезпечення використовує базу даних, оптимізація запитів, індексування даних та налагодження конфігурації бази даних можуть позитивно впливати на продуктивність. Це дозволяє зменшити час виконання запитів та покращити швидкість обробки даних.

**Ресурси обчислювальної системи:** Характеристики обчислювальної системи, такі як потужність процесора, обсяг оперативної пам'яті та пропускна здатність мережі, також впливають на продуктивність програмного забезпечення. Забезпечення достатніх ресурсів для програми дозволяє їй працювати швидко та ефективно.

**Тестування та оптимізація:** Проведення тестування продуктивності допомагає виявити слабкі місця та можливість оптимізації. Ідентифікація та усунення шляхів у процесі розробки допомагає покращити продуктивність програмного забезпечення.

**Безпека програмного забезпечення** є критичним аспектом у розробці та експлуатації будь-якого додатку чи системи. Забезпечення високого рівня безпеки дозволяє запобігти несанкціонованому доступу, зловживанню, втраті даних та іншим загрозам безпеці. Ці аспекти безпеки програмного забезпечення повинні бути враховані на кожному етапі розробки, від проектування до впровадження та підтримки. Забезпечення безпеки програмного забезпечення вимагає постійного моніторингу, оновлень і

						Аркуш
						26
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	

вдосконалення для забезпечення захисту від постійно змінюючихся загроз безпеці.

**Аутентифікація та авторизація:** Це процеси визначення і перевірки ідентифікації користувачів та надання їм прав доступу до різних функцій та ресурсів системи. Вони дозволяють забезпечити, що тільки правомірні користувачі отримують доступ до системи та її функцій.

**Захист від вразливостей:** Розробники повинні використовувати надійні практики програмування, щоб уникнути вразливостей програмного забезпечення. Це включає перевірку та очищення вхідних даних, управління пам'яттю, уникнення вразливостей зв'язків, використання безпечних бібліотек і фреймворків, а також проведення тестування на безпеку.

**Моніторинг та аудит:** Системи повинні мати можливості моніторингу та аудиту, щоб виявляти незвичайну активність, вторгнення або порушення безпеки. Це включає ведення журналів подій, аналіз журналів, системи сповіщень про вторгнення та інші механізми для виявлення потенційних загроз.

**Оновлення та патчі:** Регулярні оновлення та застосування патчів до програмного забезпечення дозволяють виправляти виявлені вразливості та підвищувати безпеку системи. Важливо відслідковувати випуск оновлень виробниками програмного забезпечення та ретельно виконувати процес оновлення.

Кількість одночасних з'єднань визначає максимальну кількість одночасних з'єднань, які можуть бути встановлені між клієнтами та сервером. Цей параметр може бути обмежений різними факторами, такими як апаратне обладнання, мережева пропускна здатність, оброблювальна потужність сервера та конфігурація програмного забезпечення. Для оптимальної продуктивності та надійності сервера важливо правильно налаштувати кількість одночасних з'єднань. Недостатня кількість одночасних з'єднань може

						Аркуш
					ДТЕУ 121 07-13.БР	27
Зм.	Аркуш	№ докум	Підпис	Дата		



призвести до того, що додаток не зможе виконувати всі завдання, які потрібно виконати, що може вплинути на його продуктивність та ефективність. Якщо кількість з'єднань перевищує максимальну допустиму, це може призвести до перевантаження сервера, падіння продуктивності та зниження доступності.

Тому важливо правильно налаштувати кількість одночасних з'єднань для додатку забезпечивши оптимальний баланс між продуктивністю та ефективністю роботи. Системна обмеження операційної системи може включати максимальну кількість одночасних TCP/IP з'єднань, максимальну кількість відкритих файлів або дескрипторів, обмеження ресурсів пам'яті та оброблювальної потужності.

Оптимальне налаштування кількості одночасних з'єднань залежить від специфіки додатку, його навантаження, очікуваної кількості користувачів та характеристик сервера. Це може вимагати виконання тестувань продуктивності та моніторингу системи для знаходження оптимального балансу між ресурсами та потребами додатку.

Пропускна здатність мережі визначає максимальний обсяг даних, який може бути переданий через мережу за певний проміжок часу. Вона вимірюється у бітах в секунду (bps) або його кратних одиницях, таких як кілобіти в секунду (Kbps), мегабіти в секунду (Mbps) або гігабіти в секунду (Gbps).

Швидкість передачі даних (Data transmission rate): Це максимальна швидкість, з якою дані можуть бути передані через фізичну мережу або її сегменти. Вона може бути обмежена характеристиками мережевих пристроїв, таких як маршрутизатори, комутатори або мережеві карти. Для забезпечення високої пропускної здатності мережі важливо правильно проектувати, налаштовувати та управляти її компонентами, враховуючи нижчезазначені фактори. Наприклад використання високошвидкісного мережевого професійного обладнання, оптимізація протоколів передачі даних та розподіл

						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			28

пропускної здатності можуть допомогти підвищити пропускну здатність мережі

**Пропагаційна затримка (Propagation delay):** Це час, який потрібен сигналу, щоб пройти від відправника до отримувача. Пропагаційна затримка залежить від фізичних властивостей мережевого середовища, таких як довжина кабелю або бездротовий зв'язок.

**Затори (Congestion):** Затори в мережі виникають, коли обсяг даних, що передаються через мережу, перевищує її здатність обробки. Це може призвести до зниження пропускної здатності мережі, затримок та втрати пакетів даних.

**Тип мережевого з'єднання:** Пропускна здатність може варіюватися в залежності від типу мережевого з'єднання. Наприклад, провідне з'єднання Ethernet може мати вищу пропускну здатність порівняно з бездротовим з'єднанням Wi-Fi.

**Інші фактори:** Інші фактори, такі як рівень шуму в мережі, розподіл пропускної здатності між різними користувачами або додатками, алгоритми керування потоками та рівень обробки даних на кінцевих пристроях, також можуть впливати на пропускну здатність.

Розміщення сервера є важливим фактором для забезпечення ефективності, доступності та безпеки клієнт-серверного додатку. Оптимальне розміщення сервера може залежати від кількох чинників. Врахування цих факторів та проведення аналізу ризиків допоможуть визначити оптимальне розміщення сервера для клієнт-серверного додатку.

**Фізичне розміщення:** Вирішення, чи буде сервер розміщений у власних приміщеннях (он-приміщення) або в арендованому дата-центрі. Власне розміщення може надати більший контроль над сервером, але вимагає наявності відповідних приміщень, енергозабезпечення, охорони та інфраструктури.

						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			29



Аренда дата-центру може забезпечити високий рівень доступності, фізичної безпеки та інфраструктури, але може бути вартісним.

**Географічне розміщення:** Розташування сервера визначається врахуванням географічної відстані до користувачів. Якщо додаток призначений для глобального використання, може бути доцільним розташування серверів у різних регіонах для забезпечення низького пінгу та швидкого відгуку для користувачів з різних частин світу.

**Безпека:** Розміщення сервера повинно забезпечувати високий рівень безпеки. Це включає фізичну безпеку (наприклад, захищені приміщення, контроль доступу), мережеву безпеку (наприклад, файрволи, VPN) та заходи безпеки даних (наприклад, резервне копіювання, шифрування).

**Мережева доступність:** Розміщення сервера повинно забезпечувати належну мережеву доступність для користувачів. Це може включати вибір провайдера Інтернету з високою швидкістю та низькими рівнями переповнення, а також забезпечення надійного мережевого обладнання та конфігурації.

**Масштабованість:** Розміщення сервера повинно бути спроектовано з урахуванням масштабованості. Якщо передбачається зростання обсягу користувачів або навантаження, важливо мати можливість легко розширити серверне середовище, додавши нові сервери або ресурси.

**Законодавство та регуляторні вимоги:** Розміщення сервера може бути обмежене законодавством та регуляторними вимогами, такими як вимоги щодо зберігання даних або обмеження щодо міжнародного переміщення даних.

**Рівень безпеки є дуже важливим аспектом веб-орієнтованого додатку, особливо якщо додаток містить конфіденційну або особисту інформацію. Рівень безпеки визначає, наскільки добре захищені дані від несанкціонованого доступу, а також наскільки добре захищений додаток від зловмисників.**

						ДТЕУ 121 07-13.БР	Аркуш
							30
Зм.	Аркуш	№ докум	Підпис	Дата			

Високий рівень безпеки може забезпечити захист від різних видів атак, таких як атаки на вразливості програмного забезпечення, відмова в обслуговуванні (DoS) та інші атаки

Захист від вразливостей, одаток повинен бути захищений від вразливостей, таких як переповнення буфера, введення SQL-запитів, міжсайтовий скриптинг тощо. Застосування технік безпеки програмного забезпечення, таких як валідація введених даних, параметризовані запити до бази даних та обмеження доступу до системних ресурсів, може допомогти запобігти вразливостям

Фізична безпека, якщо додаток розміщений на власних серверах, важливо забезпечити фізичну безпеку серверних приміщень, обмеження доступу до них та контроль над фізичним обладнанням Загальний рівень безпеки додатку залежить від того, наскільки добре реалізовані ці заходи та політики безпеки. Розробка безпечного програмного забезпечення є процесом, що вимагає систематичного підходу та постійного вдосконалення.

Вибір правильних технологій може забезпечити більш ефективну роботу додатку та отримання кращих результатів.

Вибір правильних технологій може бути важливим для успіху веб-орієнтованого додатку. Технології можуть включати мови програмування, бази даних, фреймворки та інші інструменти

Потужність обладнання впливає на ефективність та продуктивність клієнт-серверного додатку. Оптимальна потужність обладнання залежить від різних факторів, включаючи розмір і складність додатку, очікуване навантаження, кількість одночасних користувачів і вимоги до продуктивності.

Центральний процесор (CPU) потужність CPU визначає швидкість обробки даних і виконання операцій. Для додатків з великою обробкою даних або високою навантаженням, потрібен потужний і швидкий CPU.

						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			31



Обсяг оперативної пам'яті RAM впливає на здатність обробляти великі обсяги даних та запускати багато процесів одночасно. Для додатків, які працюють з великими обсягами даних або мають багато активних користувачів, потрібна достатня кількість RAM.

Сховище даних (дисковий простір) великий обсяг дискового простору необхідний для зберігання даних, файлів та інших ресурсів, використовуваних додатком. Важливо мати достатньо простору для забезпечення надійного зберігання даних та можливість масштабування, якщо обсяг даних зростає

Графічний процесор (GPU) у деяких випадках, наприклад, якщо додаток включає в себе обробку графіки або відео, потрібний потужний графічний процесор для забезпечення плавної роботи і відтворення високоякісного візуального вмісту. Крім цього, варто враховувати можливість масштабування обладнання, тобто здатність додати додаткові ресурси або оновити існуюче обладнання у разі необхідності

Каталог продуктів, якість та актуальність каталогу музичного обладнання мають велике значення для задоволення потреб клієнтів. Важливо мати повну та точну інформацію про продукти, включаючи опис, технічні характеристики, фотографії тощо.

Управління замовленнями, ефективне управління замовленнями є ключовим аспектом додатку. Користувачі повинні мати можливість легко розміщувати замовлення, відстежувати статус доставки, отримувати сповіщення про підтвердження замовлення та оновлення.

Інтеграція платіжних систем, платіжна інфраструктура грає важливу роль у забезпеченні успішних продажів. Інтеграція безпечних і зручних платіжних систем, таких як PayPal, або інші, дозволяє користувачам здійснювати покупки без зайвих перешкод.

Користувацький досвід, зручний та привабливий користувацький інтерфейс (UI/UX) є ключем до задоволення користувачів і підвищення

						Аркуш
					ДТЕУ 121 07-13.БР	32
Зм.	Аркуш	№ докум	Підпис	Дата		

продажів. Додаток повинен мати інтуїтивно зрозумілу навігацію, зручну систему фільтрації, можливість порівняння продуктів та інші функції, які полегшують процес пошуку і вибору товару.

Швидкодія та масштабованість, додаток повинен бути швидким у відповідях та здатним обробляти багато запитів одночасно, особливо в періоди збільшеного трафіку, наприклад, під час акцій або святкових періодів.

Маркетинг і просування, ефективна стратегія маркетингу та просування додатку допомагає залучати нових користувачів, збільшувати обсяг продажів і підвищувати свідомість про бренд.

## 2.2. Аналіз функціональних вимог.

Розробка веб орієнтованого додатку для підприємства з продажу музичного обладнання може включати в себе використання різних інструментів залежно від конкретних вимог проекту.

Мови програмування: Вибір мови програмування залежить від ваших вподобань та технічних вимог. Деякі популярні мови для розробки клієнтської та серверної частин додатку включають Java, Python, C#, JavaScript (зокрема Node.js), C++ Ruby

Java — це мова програмування загального призначення, створена компанією Sun Microsystems 1995 року. Її використовують для розробки програмного забезпечення, вебсервісів, ігор і застосунків.

Java є однією з найпопулярніших мов програмування у світі завдяки нескладному синтаксису, гнучкості, безпеці, портативності та масштабованості.

Згідно з рейтингом мов програмування, Java — третя за популярністю мова серед українських розробників. 14,3% айтівців писали нею станом на початок 2022 року.

Переваги Java:

						Аркуш
						33
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	



- Популярна на ринку праці;
- Має велике ком'юніті;
- Розвинена екосистема;
- Об'єктноорієнтована і суворо типізована мова, а отже, не складна у вивченні;

- перевірена часом.

Водночас Java має і низку недоліків:

- Є популярною лише у бекенд-розробці;
- Синтаксис мови є багатослівним і несучасним;
- Java потужна завдяки своїм фреймворкам і бібліотекам, які також потрібно вчити;
- Велика кількість проєктів використовує стару версію мови.

Python проста у використанні, та водночас повноцінна мова програмування, що надає набагато більше засобів для структурування і підтримки великих програм.

Python – інтерпретована мова, що дозволяє заощадити значну кількість часу, що зазвичай витрачається на компіляцію. Інтерпретатор можна використовувати інтерактивно, що дозволяє експериментувати з можливостями мови, писати шаблони програм або тестувати функції при розробці “знизу–вверх”. Він також зручний як настільний калькулятор. Python дозволяє писати дуже компактні й зручні для читання програми. Програми, написані мовою Python, звичайно значно коротші еквівалента на C або C++ з декількох причин.

- типи даних високого рівня дозволять Вам виразити складні операції однією інструкцією;
- групування інструкцій виконується за допомогою відступів замість фігурних дужок;
- немає необхідності в оголошенні змінних.

						Аркуш
						34
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	

C#, вимовляється як "C-sharp", є об'єктно-орієнтованою мовою програмування від Microsoft, яка дозволяє розробникам створювати програми, які працюють на платформі .NET. C# бере свій початок у сімействі мов програмування C і має багато тих самих характеристик, що й у C і C++, а також у Java та JavaScript.

Мова C# була розроблена в Microsoft переважно Андерсом Хейлсбергом, Скоттом Вільтамутом і Пітером Голдом. Microsoft випустила першу широко розповсюджену реалізацію C# у липні 2000 року як частину своєї ініціативи .NET Framework. C# мав бути простою, сучасною мовою програмування загального призначення, яку можна було б використовувати для розробки компонентів програмного забезпечення для розподіленого середовища.

C++ — це об'єктно-орієнтована мова програмування загального призначення. Його створив Б'ярн Страуструп у Bell Labs приблизно в 1980 році. C++ дуже схожий на C (винайдений Деннісом Річі на початку 1970-х років). C++ настільки сумісний із C, що він, ймовірно, скомпілює понад 99% програм на C, не змінюючи жодного рядка вихідного коду. Хоча C++ є значною мірою добре структурованою та безпечнішою мовою, ніж C, оскільки вона базується на ООП.

У загальному, вибір мови програмування залежить від вимог проекту, досвіду розробника та інших факторів, таких як доступність бібліотек та фреймворків.

Без баз даних додатку не розробити, База даних (БД) – це структуроване збереження даних, яке дозволяє організовувати, зберігати та управляти інформацією. База даних використовується для зберігання даних, які можуть бути доступні для обробки і використання програмами та системами. У контексті розробки клієнт-серверного додатку для підприємства з продажу музичного обладнання, база даних може використовуватись для зберігання

						Аркуш
					ДТЕУ 121 07-13.БР	35
Зм.	Аркуш	№ докум	Підпис	Дата		



інформації про продукти, клієнтів, замовлення, складський облік тощо. База даних дозволяє ефективно організувати та управляти цими даними, забезпечувати швидкий доступ до них та виконувати різні операції збереження, оновлення та вибірки даних Існують різні типи баз даних, такі як реляційні, об'єктно–реляційні, NoSQL тощо. Кожен тип має свої особливості та використовується залежно від потреб проекту. Для розробки додатку можна використовувати популярні системи управління базами даних (СУБД), такі як MySQL, PostgreSQL, Oracle, MongoDB, або інші, що відповідають вимогам проекту.

MySQL, PostgreSQL, Oracle та MongoDB – це популярні системи управління базами даних (СУБД), які використовуються для зберігання та управління даними.

MySQL: MySQL є однією з найпоширеніших відкритих реляційних СУБД. Він відомий своєю швидкодією, простотою в використанні та надійністю. MySQL часто використовується для веб–додатків, особливо тих, що побудовані на платформі LAMP (Linux, Apache, MySQL, PHP/Python/Perl). Він підтримує мови SQL для роботи з даними.

PostgreSQL: PostgreSQL є ще однією популярною відкритою реляційною СУБД. Він має багатий набір функцій та можливостей, включаючи підтримку розширень та географічних даних. PostgreSQL часто використовується для вимогливих проектів, де потрібна велика гнучкість та розширюваність.

Oracle: Oracle – це комерційна реляційна СУБД, яка широко використовується у підприємственому середовищі. Вона пропонує високу надійність, масштабованість та широкий набір функцій для управління даними. Oracle зазвичай використовується для великих корпоративних систем, де потрібна висока продуктивність та безпека.

MongoDB: MongoDB – це документоорієнтована NoSQL база даних. Вона використовує JSON–подібні документи для зберігання даних, що

дозволяє гнучко організувати структуру даних. MongoDB часто використовується для проектів, де важливо швидко зберігати та отримувати документи, таких як веб-аналітика, системи керування контентом та розробка масштабованих додатків.

Вибір СУБД залежить від ваших вимог до проекту, таких як масштабованість, продуктивність, надійність та особливості даних, які ви плануєте зберігати та обробляти. Кожна з цих СУБД має свої переваги та може бути використана для розробки клієнт-серверного додатку для підприємства з продажу музичного обладнання, але важливо врахувати вимоги проекту та можливості вашої команди розробників.

При розробці додатку для підприємства з продажу музичного обладнання можна скористатись різноманітними фреймворками, які спрощують розробку, забезпечують певний функціонал та допомагають підтримувати структурованість та ефективність проекту. Деякі з популярних фреймворків для розробки клієнт-серверних додатків.

**Django:** Django є потужним фреймворком розробки веб-додатків на мові Python. Він надає готові компоненти для роботи з базами даних, аутентифікації, шаблонами, маршрутизації та іншими функціями. Django спрощує процес розробки, дозволяючи швидко створювати масштабовані та безпечні додатки.

**Ruby on Rails:** Ruby on Rails є фреймворком розробки веб-додатків на мові Ruby. Він пропонує конвенцію над конфігурацією, що дозволяє розробникам швидко створювати функціональні додатки. Ruby on Rails включає багато вбудованих бібліотек та модулів для роботи з базами даних, маршрутизації, аутентифікації тощо.

**Node.js (з фреймворками Express.js або Koa.js):** Node.js є платформою для розробки серверних додатків з використанням JavaScript. З фреймворками Express.js або Koa.js ви можете легко створити серверну частину вашого

						Аркуш
					ДТЕУ 121 07-13.БР	37
Зм.	Аркуш	№ докум	Підпис	Дата		



додатку. Node.js дозволяє побудувати швидкі та ефективні сервери, які можуть обробляти багато одночасних з'єднань.

**ASP.NET:** ASP.NET є фреймворком розробки веб-додатків на мові C#. Він надає потужні інструменти для розробки клієнт-серверних додатків, включаючи підтримку баз даних, аутентифікацію, маршрутизацію та інші функції. ASP.NET дозволяє розробляти масштабовані та безпечні додатки для платформи .NET.

**Laravel:** Laravel є фреймворком розробки веб-додатків на мові PHP. Він пропонує елегантну синтаксичну структуру та велику кількість готових компонентів для роботи з базами даних, аутентифікації, маршрутизації та іншими аспектами розробки. Laravel дозволяє швидко розробляти додатки з хорошою структурованістю та розширюваністю.

Ці фреймворки є лише декількома прикладами інструментів, які можна використовувати для розробки веб-орієнтованих додатків. Вибір конкретного фреймворку залежить від ваших вимог, навичок розробників та особливостей проекту.

### 2.3 Висновки до 2 розділу

Для розробки веб-орієнтованого додатку можна використовувати різні мови програмування, такі як Java, Python, C++, C# або JavaScript. Для зберігання даних про обладнання та користувачів можна використовувати різні бази даних, такі як MySQL, PostgreSQL або MongoDB. Розробка за допомогою фреймворків може спростити процес розробки та зменшити кількість написаного коду. При розробці додатку для підприємства з продажу музичного обладнання можна скористатись різноманітними фреймворками, які спрощують розробку, забезпечують певний функціонал та допомагають підтримувати структурованість та ефективність проекту фреймворки, такі як Spring для Java, Flask або Django для Python, або Express.js для JavaScript. REST

						Аркуш
					ДТЕУ 121 07-13.БР	38
Зм.	Аркуш	№ докум	Підпис	Дата		

API може використовуватися для забезпечення комунікації між клієнтом та сервером, дозволяючи взаємодіяти з сервером через HTTP запити. Git може бути корисним інструментом для контролю версій коду та спільної роботи над проектом.



						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			39



## РОЗДІЛ 3

# РЕАЛІЗАЦІЯ ВЕБ-ОРІЄНТОВАНОГО ДОДАТКУ ДЛЯ ПІДПРИЄМТВА З ПРОДАЖУ МУЗИЧНОГО ОБЛАДНАННЯ

### 3.1 Бази даних та їх складові

Бази даних відіграють важливу роль у сучасному світі, де великі обсяги інформації потрібно зберігати та обробляти.

База даних (БД) – це організована колекція даних, яка зберігається та управляється в електронному форматі. Вона використовується для організації, зберігання, управління та доступу до великого обсягу даних. База даних дозволяє ефективно зберігати, оновлювати та отримувати інформацію, а також виконувати різноманітні операції над даними.

Бази даних використовуються в різних сферах, включаючи бізнес, науку, освіту, організаційні структури, інформаційні системи та багато інших. Вони є основою для зберігання та обробки великого обсягу даних, що дозволяє забезпечити доступ до інформації за потреби та забезпечити її цілісність та безпеку.

Бази даних можуть бути реляційними або нереляційними, залежно від типу структури даних та способу організації і зв'язку між ними.

Реляційні бази даних (РБД) є одним з найпоширеніших типів баз даних. Вони базуються на реляційній моделі даних, яка використовує таблиці для зберігання і структурування даних. РБД використовують спеціальні мови запитів, такі як SQL (Structured Query Language), для взаємодії з даними. Основні характеристики:

					<i>ДТЕУ 121 07-13.БР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		14.04.23	Веб-орієнтований додаток торговельного підприємства з продажу музичного обладнання	Стадія	Аркуш	Аркушів
Керівник		Котенко Н. О.		14.04.23		РЗ	40	57
Гарант		Рзаєва С.Л.		14.04.23		Факультет інформаційних технологій 4 курс, 7 група		
Розробив		Касьяненко А.С		14.04.23				

Таблиці, дані в реляційних базах даних зберігаються в таблицях, де кожна таблиця складається з рядків (записів) і стовпців (полів). Кожен запис представляє окремий елемент даних, а кожен стовпець містить конкретний тип даних.

Схема, РБД використовують схему бази даних, яка визначає структуру таблиць, типи даних, зв'язки між таблицями та обмеження цілісності даних. Схема визначає, як дані організовані і як їх можна запитувати.

Ключі, РБД використовують ключі для ідентифікації та зв'язку даних між таблицями. Первинний ключ унікально ідентифікує кожен запис в таблиці, а зовнішній ключ встановлює зв'язок між двома таблицями.

Запити, реляційні бази даних використовують SQL для створення запитів до даних. SQL дозволяє виконувати операції, такі як вибірка, вставка, оновлення та видалення даних з таблиць.

Переваги реляційних баз даних: Інтеграція та структурованість даних, РБД забезпечують організовану структуру даних, що спрощує зберігання, пошук та зв'язок між даними.

Цілісність даних, РБД надають механізми для встановлення обмежень цілісності даних, таких як унікальність значень, обмеження на значення полів тощо, що забезпечує цілісність та надійність даних.

Можливості запитів, РБД дозволяють виконувати складні запити до даних, використовуючи SQL. Це дозволяє витягувати потрібну інформацію з бази даних за допомогою різних умов та фільтрів.

Масштабованість, РБД можуть бути масштабовані для виконання зростаючих потреб даних. Додавання нових таблиць та індексів, оптимізація запитів і розподіл даних дозволяють підтримувати продуктивність навіть при зростанні обсягу даних.

Хоча реляційні бази даних мають багато переваг, вони не підходять для всіх сценаріїв та вимог. У деяких випадках, коли необхідно зберігати

						Аркуш
					ДТЕУ 121 07-13.БР	41
Зм.	Аркуш	№ докум	Підпис	Дата		



нерегулярні або великі обсяги даних, можуть використовуватись інші типи баз даних, такі як нереляційні бази даних (NoSQL) або графові бази даних.

Нереляційні бази даних, часто відомі як NoSQL (Not Only SQL), є альтернативою реляційним базам даних. Вони використовують різні моделі даних та підходи до зберігання та обробки даних, що відрізняються від традиційної реляційної моделі. Основні характеристики:

**Гнучкість моделі даних:** Нереляційні бази даних можуть використовувати різні моделі даних, такі як ключ–значення, документи, стовпці, графи тощо. Кожна модель має свої особливості і може бути використана для конкретних типів даних та сценаріїв.

**Масштабованість,** нереляційні бази даних часто мають дуже добру масштабованість горизонтальним способом. Вони можуть легко розширюватись на кластери серверів, дозволяючи обробляти великі обсяги даних та високе навантаження.

**Висока продуктивність,** завдяки своїй структурі та способу зберігання даних, нереляційні бази даних можуть забезпечувати високу продуктивність при операціях запису та читання. Вони зазвичай мають швидкий доступ до даних, особливо при великому обсязі даних.

**Переваги нереляційних баз даних:** Гнучкість, нереляційні бази даних дозволяють легко зберігати та обробляти нерегулярні, неконсистентні або невстановлені дані. Вони можуть зберігати структуровані та неструктуровані дані без потреби в жорсткій схемі.

**Швидкість,** нереляційні бази даних зазвичай забезпечують високу продуктивність, особливо при операціях запису та читання великого обсягу даних. Вони можуть швидко масштабуватись, щоб задовольнити зростаючі потреби в продуктивності.

						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			42

Масштабованість, нереляційні бази даних добре пристосовані для розподіленого середовища. Вони можуть бути легко розширені на кластери серверів, дозволяючи обробляти великі обсяги даних та високу навантаження.

Недоліки нереляційних баз даних: Відсутність стандартизації оскільки нереляційні бази даних використовують різні моделі та підходи, відсутня єдина стандартизація. Це може призвести до складнощів у взаємодії з різними системами та інструментами.

Обмежена підтримка мови запитів у порівнянні з SQL, який є потужною мовою запитів для реляційних баз даних, нереляційні бази даних можуть мати обмежені можливості у виконанні складних запитів та аналізу даних.

Складність моделювання зв'язків в деяких випадках, коли дані мають складні зв'язки та взаємозв'язки, моделювання структури даних у нереляційних базах даних може бути складним та вимагати спеціалізованих знань.

У кожному конкретному випадку вибір між реляційною та нереляційною базою даних залежить від вимог проекту, типу даних, очікуваного навантаження та інших факторів. Обидва підходи мають свої переваги та обмеження, і важливо обґрунтувати свій вибір на основі конкретних вимог проекту.

Для проекту була вибрана реляційна база даних MS SQL.

Microsoft SQL Server (MS SQL) є реляційною базою даних, розробленою компанією Microsoft. Вона є однією з найпопулярніших комерційних баз даних на ринку і забезпечує широкий набір функціональних можливостей для зберігання, керування та обробки даних. До переваг відносяться:

Надійність: MS SQL Server має високий рівень надійності та стійкості. Він пропонує механізми резервного копіювання, відновлення та реплікації, що забезпечують захист даних від втрати і можливість відновлення в разі виникнення неполадок.

						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			43



Масштабованість MS SQL Server може бути легко масштабований як вертикальним (за допомогою потужнішого обладнання), так і горизонтальним (за допомогою кластерів серверів). Це дозволяє вам розширювати його можливості для обробки зростаючого обсягу даних та навантаження.

Безпека: MS SQL Server пропонує різні механізми безпеки для захисту даних, включаючи автентифікацію, авторизацію, шифрування та аудит. Ви можете керувати правами доступу користувачів до бази даних та забезпечити захищений доступ до важливих даних.

Зручність у використанні: MS SQL Server має потужне графічне інтерфейсне середовище управління (SQL Server Management Studio), яке надає зручні інструменти для адміністрування бази даних, написання запитів та моніторингу продуктивності.

Незважаючи на переваги, MS SQL Server має деякі недоліки, зокрема:

Вартість MS SQL Server є комерційним продуктом, тому використання його може бути пов'язано з витратами на ліцензії та підтримку. Це може бути фактором, який впливає на бюджет проекту.

Обмежена переносимість MS SQL Server працює під керуванням операційної системи Windows і є відносно менш переносимою порівняно зі звичайними відкритими реляційними базами даних, які підтримуються на різних платформах.

Обмежена безпека на рівні рядків: MS SQL Server не надає рівень безпеки на рівні рядків, що означає, що контроль доступу до окремих рядків даних може бути обмеженим.

Однак, з урахуванням потужного функціоналу та надійності MS SQL Server є популярним вибором для багатьох підприємств, особливо тих, що працюють на платформі Windows і потребують потужної та масштабованої реляційної бази даних.

						ДТЕУ 121 07-13.БР	Аркуш
							44
Зм.	Аркуш	№ докум	Підпис	Дата			

### 3.2. Впровадження розробленої системи

Головна сторінка сайту з продажу музичного обладнання є важливою, оскільки вона першою сторінкою, яку бачать відвідувачі як показано на рисунку 6, на сторінці можуть бути такі елементи як логотип та назва компанії: Розміщення логотипу та назви компанії на видному місці сприяє брендунню і підкреслює ідентичність вашої компанії. Слайдер або банери: Ротуючі слайди або банери можуть бути використані для презентації нових товарів, спеціальних пропозицій або рекламних матеріалів, що стимулює інтерес та залучає увагу відвідувачів. Навігаційне меню: Меню, яке містить основні категорії товарів або послуг, допомагає відвідувачам швидко зорієнтуватися на сайті та перейти до потрібних розділів.



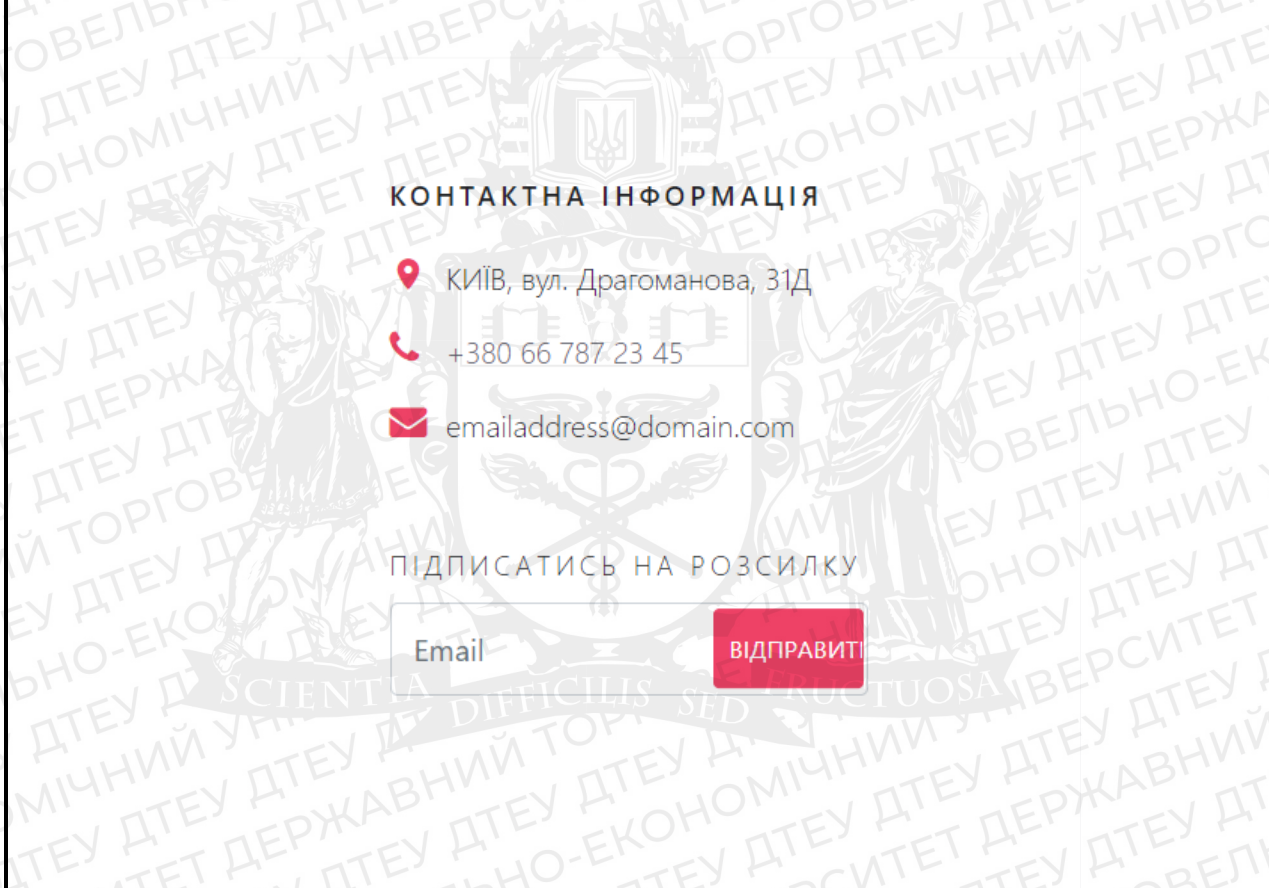
*Рисунок 3.1– Головна сторінка  
Джерело: побудовано автором*

Надання контактної інформації на сайті є важливим для спілкування з відвідувачами та клієнтами, цей розділ повинен знаходитися на видному та зручному місці для користувача як показано на рисунку 7 для того щоб він зміг

					ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		45



швидко та інформативно знайти інформацію яку він шукає без зайвих труднощів, контактна інформація може містити в собі номер телефону, адресу, пошту імейл та різні соціальні мережі такі як телеграм, інстаграм, вайбер, ютуб та тощо. Ці елементи контактної інформації можна розмістити на окремій сторінці "Контакти" або в підвалі (footer) сайту, щоб вони були доступні на кожній сторінці сайту.



*Рисунок 3.2 – Контактна інформація  
Джерело: побудовано автором*

Створення зручного та організованого каталогу на сайті з продажу музичного обладнання є важливим для забезпечення зручного пошуку та перегляду товарів він повинен містити в собі такі елементи як категорії товарів для того щоб користувач зміг швидко знайти саме ту категорію яка йому потрібна (Рис 8,9), наприклад ‘Духові інструменти’, фільтри, щоб дозволити

						Аркуш
						46
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	

користувачам швидко обмежити вибір за такими параметрами, як ціновий діапазон, бренд, характеристики тощо. Також потрібний зручний пошук це можливість пошуку за ключовими словами або фразами, щоб користувачі могли швидко знайти потрібний товар, якщо вони знають точну назву або опис.



Рисунок 3.3 – Каталог

Джерело: побудовано автором

Вкладка товару є важливою частиною сторінки товару на сайті з продажу музичного обладнання. Вона надає детальну інформацію про конкретний товар, що допомагає користувачам зробити інформований вибір. Вона може містити в собі такі елементи як детальний опис товару, з його повними характеристиками, фото та відео, ціною та відгуками користувачів, це все допомагає клієнту обрати та придбати саме той товар який йому потрібен, також для більш точного вибору може бути додана функція порівняння товарів, їх характеристик та цін. (Рис 10)

						Аркуш
						47
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	



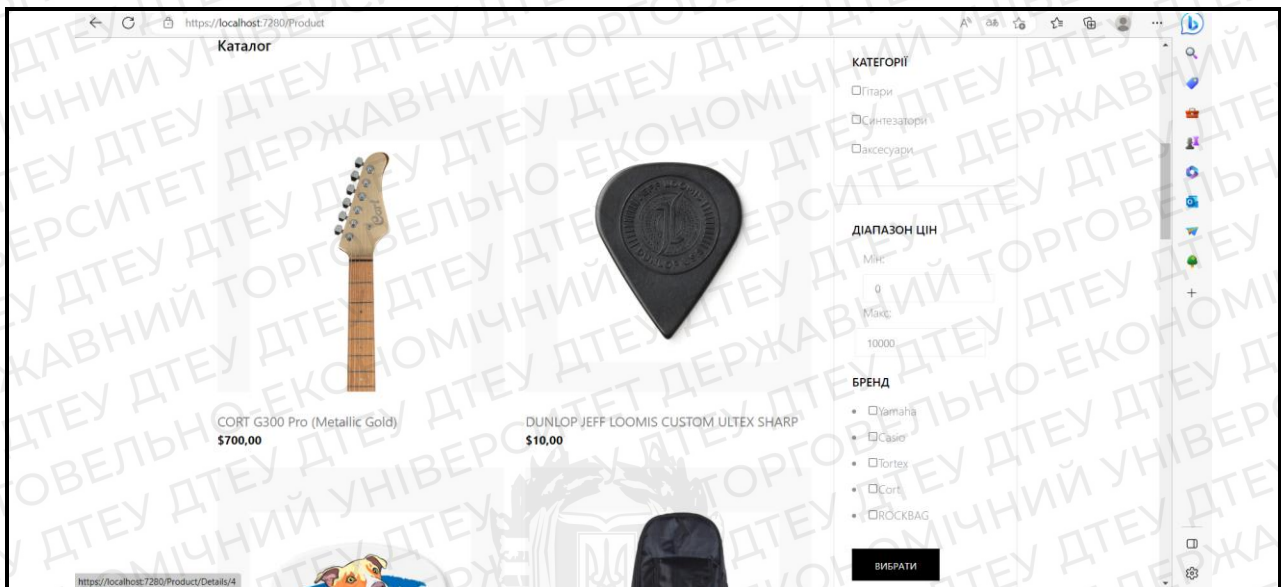


Рисунок 3.4 – Каталог  
Джерело: побудовано автором

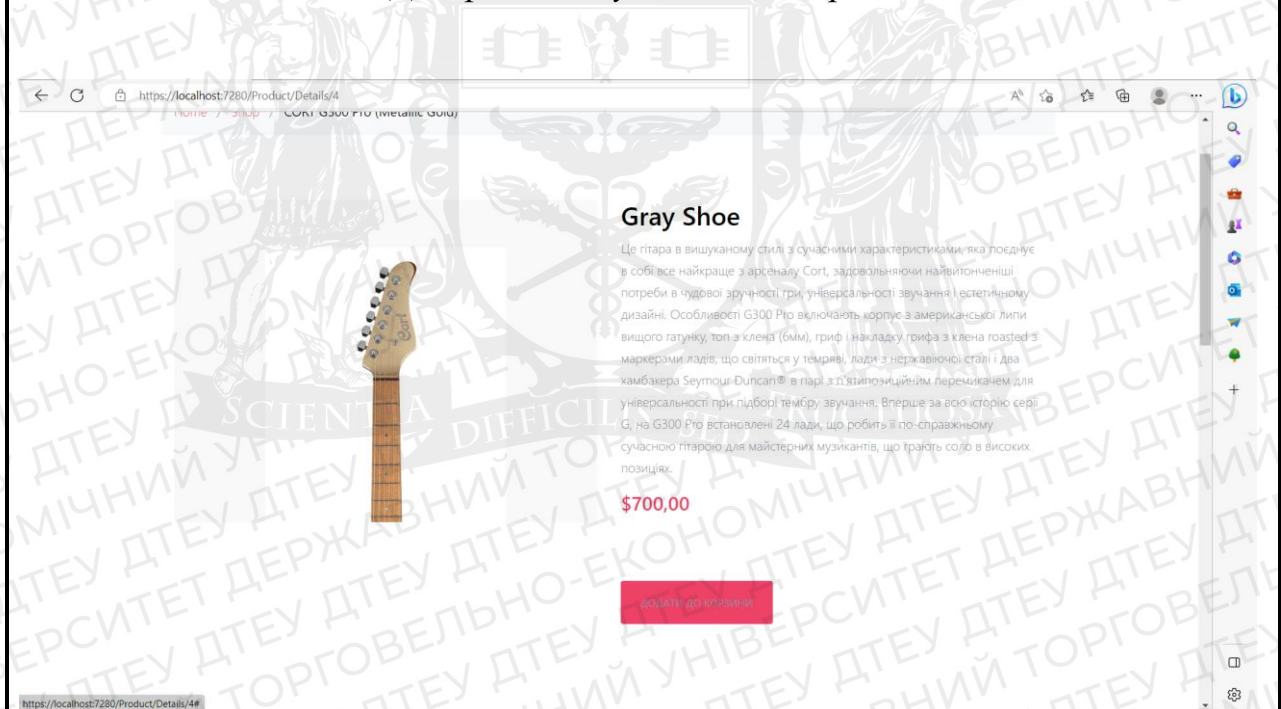


Рисунок 3.5 – Опис товару  
Джерело: побудовано автором

Не менш важливою складовою є кошик, Він надає можливість користувачам збирати обрані товари перед оформленням замовлення. Включає в себе додавання товарів, видалення, відображення змісту кошика,

						Аркуш
						48
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	

показування користувачам список товарів, які вони додали до кошика, відображає важливу інформацію, таку як назва товару, фотографія, кількість одиниць, ціна та загальна вартість замовлення, також повинна бути функція групування товарів та їх загальна ціна.



Рисунок 3.6 – Кошик

Джерело: побудовано автором

### 3.3 Висновки до 3 розділу

Бази даних відіграють важливу роль у сучасному світі, де великі обсяги інформації потрібно зберігати та обробляти.

База даних (БД) – це організована колекція даних, яка зберігається та управляється в електронному форматі. Вона використовується для організації, зберігання, управління та доступу до великого обсягу даних. База даних дозволяє ефективно зберігати, оновлювати та отримувати інформацію, а також виконувати різноманітні операції над даними.

А інтерфейс веб-орієнтованого додатку є критично важливим, оскільки він є першим контактом користувача з веб-додатком, якість інтерфейсу може



має значний вплив на враження користувача, його зручність використання та загальне задоволення від взаємодії з сайтом



						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			50

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Попит на музичне обладнання є суттєвим елементом музичної індустрії. Цей попит визначається широким спектром факторів, які включають музичні тенденції, потреби музикантів, розвиток технологій та фінансові можливості споживачів

Одним з ключових факторів, що впливають на попит на музичне обладнання, є музичні тенденції. Зміни у смаках і вподобаннях аудиторії можуть створювати попит на конкретні типи обладнання.

Наприклад, якщо популярним стає певний жанр музики, збільшується попит на обладнання, яке відповідає потребам виконавців цього жанру, наприклад, електронні музичні інструменти або звукові системи.

Другим фактором, що впливає на попит на музичне обладнання, є потреби музикантів. Професійні музиканти, студії звукозапису та виставкові зали часто шукають високоякісне та професійне обладнання для своїх потреб.

Вони оцінюють якість звуку, надійність та функціональні можливості обладнання. У цьому контексті попит на професійне аудіо– та студійне обладнання може бути високим.

Розвиток технологій також має значний вплив на попит на музичне обладнання. Інновації у галузі звукових технологій, синтезу звуку, цифрової обробки сигналів та інших областей сприяють появі нових типів обладнання. Нові можливості та функціональність привертають увагу музикантів і можуть створювати попит на ці продукти. Отже, попит на музичне обладнання є результатом взаємодії різних факторів, таких як музичні тенденції, потреби музикантів, технологічний прогрес та фінансові можливості споживачів.

					<i>ДТЕУ 121 07-13.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Веб-орієнтований додаток торговельного підприємства з продажу музичного обладнання  Висновки та пропозиції	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		28.04.23		<i>ВП</i>	51	57
Керівник		Котенко Н. О.		28.04.23		<i>Факультет інформаційних технологій 4 курс, 7 група</i>		
Гарант		Рзаєва С.Л.		28.04.23				
Розробив		Касьяненко А.С		28.04.23				



Розуміння цих факторів допомагає виробникам та продавцям адаптуватися до змін в попиті та задовольняти потреби ринку музичного обладнання

Для розробки клієнт–серверного додатку можна використовувати різні мови програмування, такі як Java, Python, C++, C# або JavaScript. Для зберігання даних про обладнання та користувачів можна використовувати різні бази даних, такі як MySQL, PostgreSQL або MongoDB. Розробка за допомогою фреймворків може спростити процес розробки та зменшити кількість написаного коду. При розробці додатку для підприємства з продажу музичного обладнання можна скористатись різноманітними фреймворками, які спрощують розробку, забезпечують певний функціонал та допомагають підтримувати структурованість та ефективність проекту фреймворки, такі як Spring для Java, Flask або Django для Python, або Express.js для JavaScript.

						ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			52

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. UA5.ORG [Електронний ресурс] URL: <https://ua5.org/oop/392-diagrami-klasiv.html>
2. StudFiles [Електронний ресурс] URL: <https://studfile.net/preview/5010027/page:4/>
3. UKMAIR [Електронний ресурс] URL: <https://ekmair.ukma.edu.ua/server/api/core/bitstreams/148bf8ee-33fd-49fb-a24d-a45e94b8603b/content>
4. DOU [Електронний ресурс] URL: <https://dou.ua/lenta/articles/how-to-learn-java/>
5. Plone [Електронний ресурс] URL: <http://www.plug.org.ua/documentation/about-python>
6. TechTarget [Електронний ресурс] URL: <https://www.techtarget.com/whatis/definition/C-Sharp>

<i>ДТЕУ 121 07-13.БР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
Зав. каф.		Криворучко О.В.		23.12.22
Керівник		Котенко Н. О.		23.12.22
Гарант		Рзаєва С.Л.		23.12.22
Розробив		Касьяненко А.С		23.12.22
Веб-орієнтований додаток торговельного підприємства з продажу музичного обладнання Список використаних джерел				
		<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
		СВД	53	57
Факультет інформаційних технологій 4 курс, 7 група				



## ДОДАТКИ

Додаток А

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using MusicShop.Data;
using MusicShop.Models;
using MusicShop.ViewModels;

namespace MusicShop.Controllers
{
    public class ProductController : Controller
    {
        private readonly DataContext _context;

        public ProductController(DataContext context)
        {
            _context = context;
        }

        public IActionResult Index(string brand, string searchString, int[]
brandIds, int[] categoryIds, int page = 1, string sort = "name", decimal minPrice =
0, decimal maxPrice = 100000)
        {
            int pageSize = 10;
            var categories = _context.Categories.ToList();
            var products = _context.Products.Include(p =>
p.Category).Include(p => p.ProductImages).AsQueryable();
            //IQueryable<Product> products = _context.Products;
            if (!string.IsNullOrEmpty(searchString))
            {
                products = products.Where(p =>
p.Name.Contains(searchString) || p.Description.Contains(searchString));
            }
            if (categoryIds != null && categoryIds.Length > 0)
            {
                products = products.Where(p =>
categoryIds.Contains(p.CategoryId));
            }
            if (brandIds != null && brandIds.Length > 0)
            {
                products = products.Where(p => brandIds.Contains(p.BrandId));
            }
            switch (sort)
            {
                case "name":
                    products = products.OrderBy(p => p.Name);
                    break;
                case "price":
                    products = products.OrderBy(p => p.Price);
                    break;
                // інші варіанти сортування
                default:
                    products = products.OrderBy(p => p.Id);
                    break;
            }
        }
    }
}
```

						Аркуш
					ДТЕУ 121 07-13.БР	54
Зм.	Аркуш	№ докум	Підпис	Дата		

```

if(minPrice > 0 && maxPrice > 0)
{
    products = products.Where(p => p.Price >= minPrice && p.Price
<=maxPrice);
}

var viewModel = new ProductsViewModel
{
    Products = products.Skip((page - 1) * pageSize).Take(pageSize),
    PagingInfo = new PagingInfo
    {
        CurrentPage = page,
        ItemsPerPage = pageSize,
        TotalItems = products.Count()
    },
    CurrentSortOrder = sort,
    CategoryListViewModel = new CategoryListViewModel
    {
        Categories = categories,
        SelectedCategories = categoryIds
    },
    Brands = _context.Brands.ToList()
};
//var t = _context.ProductImages.ToList();
return View(viewModel);
}

public async Task<IActionResult> Details(int id)
{
    var product = await _context.Products
.Include(p => p.Category)
.Include(p => p.ProductImages)
.Include(p => p.ProductSpecifications)
.FirstOrDefaultAsync(p => p.Id == id);

    if (product == null)
    {
        return NotFound();
    }

    ViewBag.Reviews = _context.ProductReviews.ToList();
    return View(product);
}
[HttpPost]
public IActionResult AddReview(ProductReview review)
{
    var product = _context.Products.SingleOrDefault(p => p.Id ==
review.ProductId);
    if (product == null)
    {
        return NotFound();
    }

    review.Date= DateTime.Now;
    review.IsApproved = true;

    _context.ProductReviews.Add(review);
    _context.SaveChanges();

    return RedirectToAction("Details", "Product", new { id =
review.ProductId });
}

```

						Аркуш
						55
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-13.БР	



## Додаток Б

```
    }  
    }  
}  
  
using Microsoft.AspNetCore.Mvc;  
using MusicShop.Data;  
using MusicShop.Models;  
using MusicShop.ViewModels;  
using Newtonsoft.Json;  
  
namespace MusicShop.Controllers  
{  
    public class OrderController : Controller  
    {  
        private readonly DataContext _context;  
  
        public OrderController(DataContext context)  
        {  
            _context = context;  
        }  
  
        public IActionResult Index()  
        {  
            return View();  
        }  
  
        public IActionResult Checkout()  
        {  
            return View();  
        }  
  
        [HttpPost]  
        public IActionResult Checkout(OrderViewModel order)  
        {  
            var cartString = HttpContext.Session.GetString("Cart");  
            var cart = new Cart();  
            if (cartString != null)  
            {  
                cart = JsonConvert.DeserializeObject<Cart>(cartString);  
            }  
            else  
            {  
                ModelState.AddModelError("", "Sorry, your cart is empty!");  
            }  
  
            if (ModelState.IsValid)  
            {  
                var newOrder = new Order()  
                {  
                    Name = order.Name,  
                    Address = order.Address,  
                    Email = order.Email,  
                    Phone = order.Phone,  
                    Comment = order.Comment,  
                    OrderDate = DateTime.Now,  
                    Total = cart.GetTotal(),  
                };  
            }  
        }  
    }  
}
```

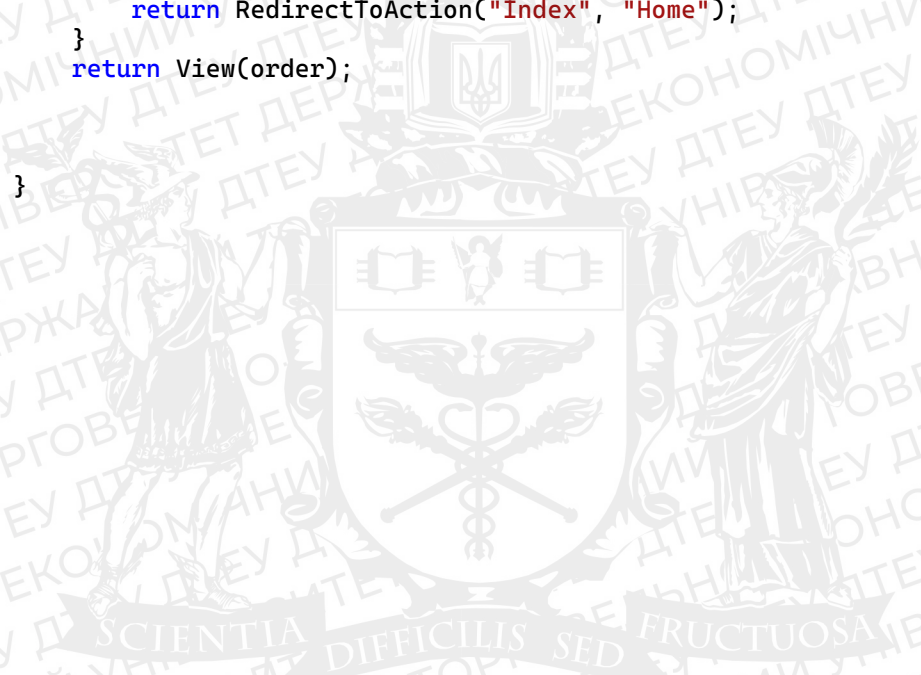
					ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		56

```

_context.Orders.Add(newOrder);
_context.SaveChanges();

foreach (var item in cart.Items)
{
    var orderItem = new OrderItem()
    {
        OrderId = newOrder.OrderId,
        ProductId = item.ProductId,
        Quantity = item.Quantity
    };
    _context.OrderItems.Add(orderItem);
}
_context.SaveChanges();
HttpContext.Session.Remove("Cart");
return RedirectToAction("Index", "Home");
}
return View(order);
}
}
}

```



					ДТЕУ 121 07-13.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		57