

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

КУРСОВИЙ РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Бази даних»

на тему:

«База даних інтернет-магазину вінілових платівок»

студента II курсу групи ПЗ-20-2
спеціальності 121 «Інженерія програмного
забезпечення»

Палія Івана Володимировича

(прізвище, ім'я та по-батькові)

Керівник Кравченко С.М.

Дата захисту: " 27 " липня 2022 р.

Національна шкала _____

Кількість балів: _____

Оцінка: ECTS _____

Члени комісії

_____	<u>І.І.Сугоняк</u>
(підпис)	(прізвище та ініціали)
_____	<u>О.В. Коротун</u>
(підпис)	(прізвище та ініціали)
_____	<u>О.В. Чижмотря</u>
(підпис)	(прізвище та ініціали)
_____	<u>С.М. Кравченко</u>
(підпис)	(прізвище та ініціали)

Житомир – 2022

Спеціальність 121 «Інженерія програмного забезпечення»

“ ” 2022p.

[illegible]

8. Дата видачі завдання “20” Лютого 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсового проекту	Строк виконання етапів проекту	Примітки
1	Постановка задачі	01.03 -07.03	виконано
2	Пошук, огляд та аналіз аналогічних розробок	08.03 -12.03	виконано
3	Формулювання технічного завдання	13.03- 18.03	виконано
4	Опрацювання літературних джерел	19.03- 23.03	виконано
5	Проектування структури	24.03- 10.04	виконано
6	Написання програмного коду	11.04- 10.06	виконано
7	Відлагодження	11.06- 30.06	виконано
8	Написання пояснювальної записки	01.07- 03.07	виконано
9	Захист	28.08	виконано

Студент _____
(підпис)

Керівник проекту _____
(підпис)

Палій І.В.
(прізвище та ініціали)

Кравченко С.М.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до курсового проекту на тему «База даних інтернет-магазину вінілових платівок», складається зі вступу, чотирьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 34 сторінках друкованого тексту.

Пояснювальна записка має 14 сторінок додатків. Список використаних джерел містить 6 найменувань і займає 1 сторінку. В роботі наведено 15 рисунків. Загальний обсяг роботи 48 сторінок.

У першому розділі було обґрунтовано створення бази даних на тему інтернет магазину вінілових платівок.

У другому розділі проведено проектування і розробка сайту.

У третьому розділі проведено тестування програмного продукту.

Висновок містить в собі результати виконаної роботи при створенні сайту на тему «База даних інтернет-магазину вінілових платівок».

У додатку представлений лістинг розробленого програмного продукту.

КЛЮЧОВІ СЛОВА: ВІНІЛОВІ ПЛАТІВКИ, МАГАЗИН, ВЕБ, ДИЗАЙН, ІНТЕРНЕТ, ПОКУПКИ, JAVASCRIPT, WEB-САЙТ, MONGODB, СЕРВЕР. АВТОРИЗАЦІЯ, МАГАЗИН, ІНТЕРНЕТ, ДАНІ, NODE JS, REACT,.

					ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка інтернет магазину вінілових платівок	Літ.	Арк.	Аркушів
Розроб.		Палій І.В.						
Перевір.		Кравченко С.М.					4	48
Керівник						ФІКТ Гр. ІПЗ-20-2[2]		
Н. контр.								
Зав. каф.		Морозов А.В.						

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ.....	8
1.1 Аналіз інформаційних потреб та визначення предметної області дослідження.....	8
1.2 Обґрунтування вибору засобів реалізації.	12
Висновок до 1 розділу:	12
РОЗДІЛ 2. ПРОЕКТУВАННЯ БАЗИ ДАНИХ ІНТЕРНЕТ МАГАЗИНУ ВІНІЛОВИХ ПЛАТІВОК	13
2.1 Аналіз інформаційних процесів.....	13
2.2 Проектування структури бази даних сайту	16
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ ІНТЕРНЕТ МАГАЗИНУ	21
3.1 Проектування інтерфейсу обробки даних	21
3.2 Реалізація операцій обробки даних з БД.....	25
3.3 Організація звітності системи.....	28
Висновки до 3 розділу	29
РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ.....	30
4.1 Розробка заходів захисту інформації в БД	30
4.2 Налаштування параметрів роботи в MongoDB	31
Висновки до 4 розділу	33
ВИСНОВКИ.....	34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	35
ДОДАТКИ.....	36
Додаток А.....	37
Додаток В	43

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – База даних

СУБД – система управління базами даних

IDE – Інтегроване середовище розробки

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Актуальність теми. Створення інтернет магазину стає все більш популярною послугою. За останніми даними, аудиторія в інтернеті стрімко зростає, а продажі через інтернет у великих містах досягають до 25%, при цьому фахівці підкреслюють тенденцію до зростання продажів саме через інтернет.

Сайт інтернет-магазин – є сучасним торговим каналом. За допомогою інтернет магазину Ви маєте можливість продавати Ваші товари або послуги величезної аудиторії, яка використовує доступ до Інтернету. Інтернет магазин для покупця це: економія часу, грошей та сил. Саме тому, за статистикою, все більше людей у світі робить свої покупки через інтернет магазин.

Тема курсового проекту – «Інтернет магазин». Даний додаток буде працювати напряду з базою даних та відображати всю необхідну інформацію для користувача.

Метою створення даного курсового проекту є розробка базу даних магазину вінілових платівок який містить:

Завданням дослідження було:

- 1) каталог товарів;
- 2) інформаційні сторінки товарів;
- 3) кошик
- 4) Оформлення замовлень, та можливість перегляду замовлень адміністратора, видалення замовлення тим же адміністратором.
- 5) Авторизацію та вхід на сайт;
- 6) панель адміністратора;
- 7) форма для додавання.

Об'єктом дослідження – є розробка бази даних для в середовищі MongoDB та використання мови NodeJS для роботи з нею

Предметом дослідження вивчення можливостей стеку MERN (MongoDB, Express, React, NodeJS), для розробки WEB – додатків.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1 Аналіз інформаційних потреб та визначення предметної області дослідження.

На початку роботи ознайомимося із поставленою задачею курсового проекту(роботи). Це дозволить більш ефективно та якісно працювати з БД.

Задачею курсового проекту є написання завершеного програмного продукту. Він повинен бути зрозумілим для усіх, простим у використанні, мати простий та зручний інтерфейс, без зайвих деталей. WEB-додаток має коректно працювати при усіх можливих варіантах поведінки користувача, захист від не вірно введених даних.

1. База даних має містити інформацію про товари, користувачів, їхні корзини та замовлення, рейтинг товару, та допоміжну інформацію для визначення основного функціоналу проекту.

2. Аналіз вимог

2.1.Користувач може переглянути всі товари.

2.2.Користувач може переглянути інформацію про конкретний товар.

2.3. Користувач може додати цей товар до кошику, якщо він авторизований.

2.4. Користувач може маніпулювати кількість товару у кошику.

2.5. Користувач може замовити товари у кошику.

2.6. Користувач може залишити рейтинг товару.

2.7. Адміністратор може додавати або видаляти товар з сайту.

2.8.Адміністратор може продивлятися та видаляти заклази користувача.

2.9.Адміністратор може дивитися статистику магазина.

2.10.

3. Проаналізувати способи для реалізації поставлених цілей.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

4. Спроектувати загальний алгоритм взаємодії клієнтської частини з серверною та базою даних, а також алгоритми роботи програми.

5. Реалізація

5.1. Проектування бази даних:

- Модель користувача
- Модель товару
- Модель корзини
- Модель JWT-токенів для користувача
- Модель замовлення
- Модель рейтингу

5.2. Реалізація основних маніпуляцій над товаром та корзиною.

5.3 Створити систему авторизації для користувача:

- Використовується 2 JWT-токенів. AccessToken для авторизації на сайті, та RefreshToken для оновлення AccessToken`а

5.4 Створити клієнтську частину та сторінки сайту використовуючи ReactJS

- Сторінка авторизації
- Сторінка реєстрації
- Головна сторінка з товарами
- Сторінка товару
- Сторінка корзини
- Сторінка адміна
- Сторінка замовлень
- Сторінка з статистикою

Проведемо порівняльну характеристику декількох розповсюджених систем управління базами даних та визначимо ту що найбільш задовільнить наші потреби. Для порівняльної характеристики було обрано три бази даних: Microsoft SQL Server, Oracle, та MongoDB.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Функціональне порівняння СУБД

Характеристика	MongoDB	Oracle	Microsoft SQL Server
Адміністративне керування	Відмінно	Відмінно	Добре
Графічні інструменти	Відмінно	Відмінно	Відмінно
Простота обслуговування	Відмінно	Відмінно	Відмінно
Механізм даних	Добре	Добре	Добре
Робота с декількома ЦП	Добре	Відмінно	Задовільно
Одночасний доступ декількох користувачів	Відмінно	Відмінно	Добре
Повнотекстовий пошук	Задовільно	Відмінно	Добре
Інтеграція з іншими СУБД	Відсутня	Добре	Добре
Мова SQL	Відсутня	Відсутня	Відмінно
Засоби завантаження	Відмінно	Відмінно	Задовільно
Засоби аналізу	Задовільно	Задовільно	Добре

Переваги та недоліки сервісів**MongoDB :**

Переваги:

- Open Source
- Дає змогу зберігати багато неструктурованих даних
- Дуже легко масштабується
- Висока швидкість читання, запису даних

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Недоліки:

- Відсутні зв'язки між документами
- Виникають ситуації коли данні ніяк не зв'язані і це неможливо виправити

Oracle :

Переваги:

- Висока надійність та відмово стійкість
- Доступна документація
- Багато найновіших функцій

Недоліки:

- Ціна
- Потребує багато ресурсів
- Складність налаштування та розгортання.

Microsoft SQL Server :

Переваги:

- Висока надійність та відмово стійкість
- Доступна докладна документація
- Підтримка локальних і хмарних БД

Недоліки:

- Ціна
- Залежність від ОС, працює лише з Windows

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Обґрунтування вибору засобів реалізації.

В якості основних засобів реалізації було обрано Node.js, мову програмування, та MongoDB, система управління базами даних, з використанням необхідних бібліотек.

MongoDB - це популярна Open Source документоорієнтована система керування базами даних, це означає, що вона зберігає дані у схожих на JSON (JavaScript Object Notation) документах. MongoDB має потужну мову запитів, багатомарний розподіл даних, є швидкою та зручною у використанні та гарно пристосована для використання у web додатках. На відміну від інших СУБД, в MongoDB різні документи однієї і тієї ж колекції можуть мати різні поля і типи. В нашому випадку це дуже корисно. Використовувана модель документів зберігання даних простіше кодується, а внутрішнє угруповування релевантних даних забезпечує додатковий вииграш в швидкодії.

Для реалізації даного програмного продукту чудово підходить IDE WebStorm, яка є гнучкою та ідеально підходить для використання нашої БД. Це дозволяє швидко знаходити і виправляти помилки, що сильно спрощує написання великої кількості програмного коду. Для побудови серверної частини було обрано платформу Node.js, і додаткові бібліотеки, зокрема mongoose для спрощення взаємодії з базою даних. Для частини клієнту було обрано бібліотеку React і готові стилі для спрощення та прискорення розробки.

Всі потрібні інструменти були встановлені на локальній машині і виконували на ній.

Висновок до 1 розділу:

Проаналізувавши переваги та недоліки існуючих рішень, було поставлено цілі до проекту, а саме критерії функціоналу, обґрунтовано вибір СУБД та проведено необхідні дослідження

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ БАЗИ ДАНИХ ІНТЕРНЕТ МАГАЗИНУ ВІНІЛОВИХ ПЛАТІВОК

2.1 Аналіз інформаційних процесів.

Блок схему загального алгоритму нашого проекту:

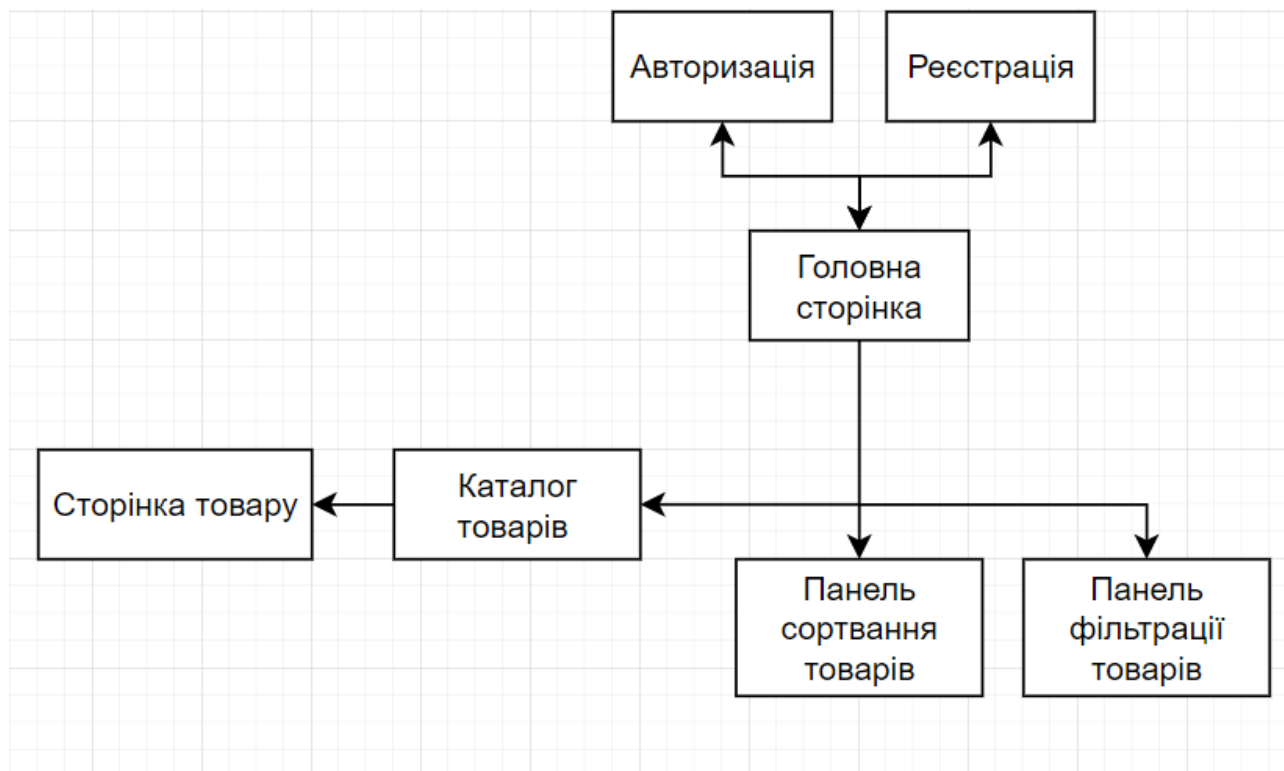


Рис. 2.1 Можливості неавторизованого користувача.

Можливості неавторизованого користувача досить обмежені – він може переглядати список товарів, включно з використанням фільтрів та сортування, зареєструватись та увійти у свій акаунт.

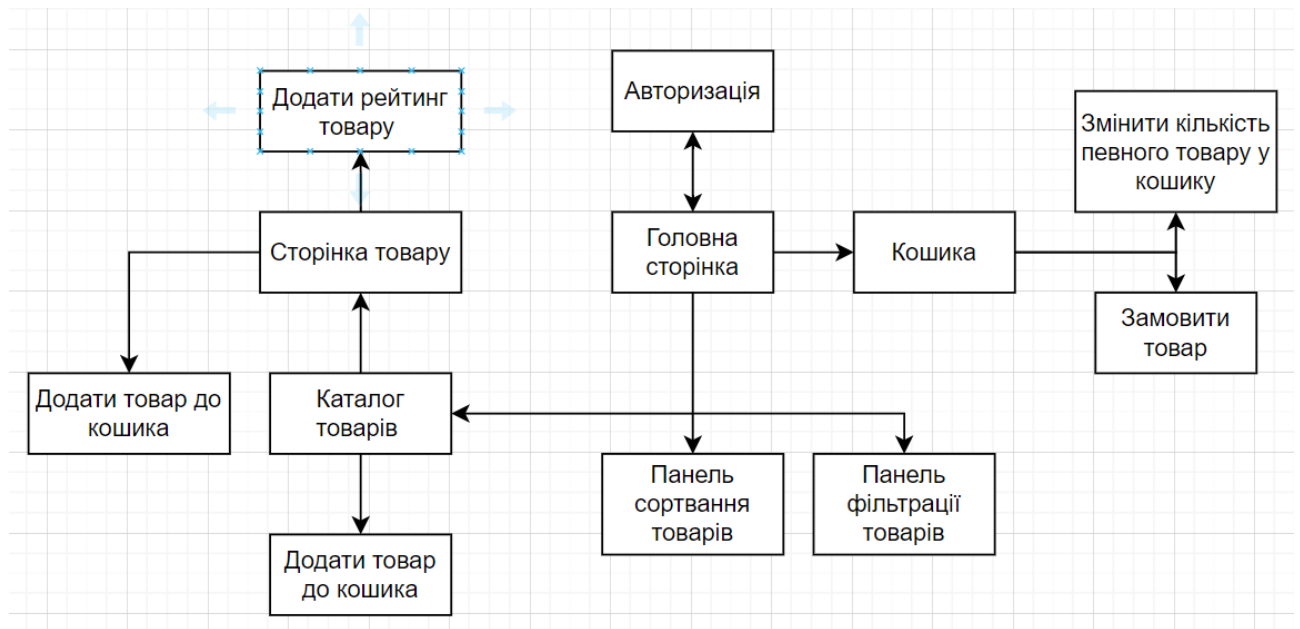


Рис. 2.2 Можливості авторизованого користувача.

Можливості авторизованого користувача значно більші – він може все що може неавторизований користувач, також додавати товари до корзини, залишати рейтинг до товару, змінювати кількість товарів в корзині, створювати замовлення.

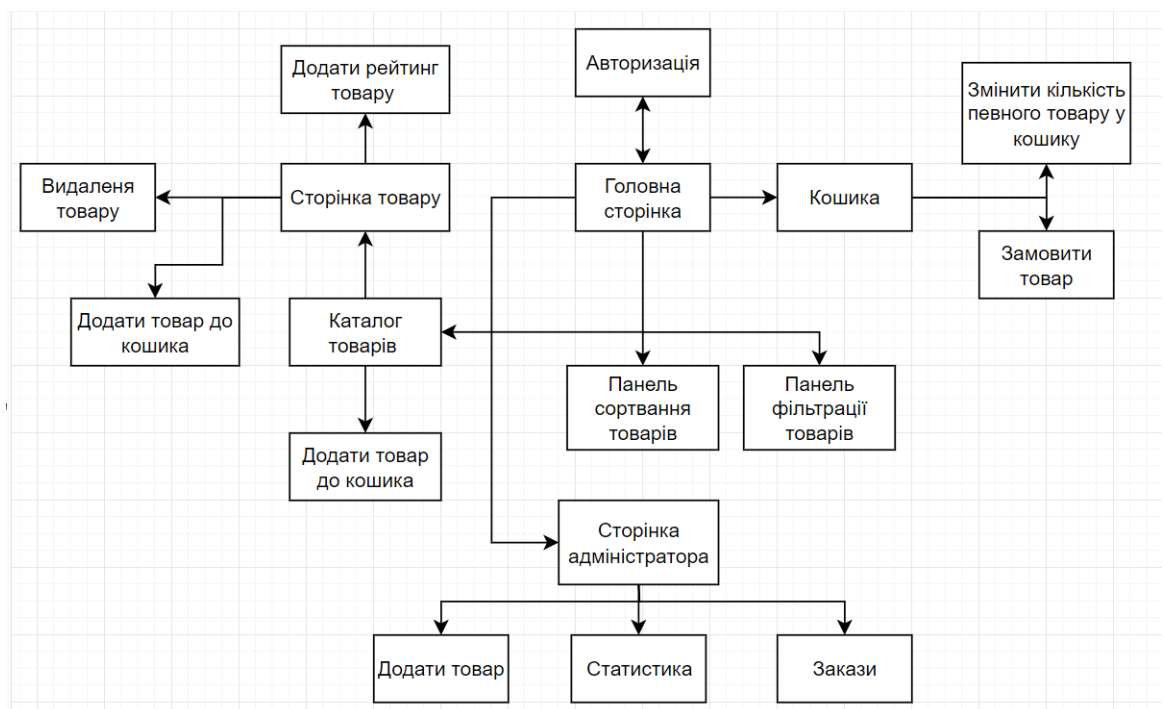


Рис. 2.3 Можливості адміністратора.

Можливості адміністратора – він може все, що було передбачено програмою. Всі функції звичайних користувачів, плюс, маніпуляції товарами, перегляд статистики, та перегляд і маніпуляція заказами.

Після запуску програми завантажуються початкова сторінка. Після чого очікується активність користувача.

Можуть бути виконані такі дії:

Дії на головній сторінці

- Можна переглянути всі товари та застосувати фільтри і сортування
- Можна перейти до сторінки товару
- Можна авторизуватися
- Якщо користувач авторизувався, він може одразу додати товар до кошику.

Дії на сторінці товару

- Можна додати товар до кошику, якщо користувач авторизований
- Можна додати рейтинг до товару
- Якщо користувач АДМІН, він може видалити товар

Дії на сторінці авторизації

- Якщо користувач зареєстрований, він може авторизуватися
- Якщо користувач не має обліковий запис, він може перейти до сторінки реєстрації натиснувши на кнопку «Register»

Дії на сторінці реєстрації

- Якщо користувач не зареєстрований, він може зареєструватися
- Якщо користувач має обліковий запис, він може перейти до сторінки авторизації натиснувши на кнопку «Log In»

Дії на сторінці корзини

- Можна додати або відняти кількість товару в корзині
- Можна оформити замовлення

		Дії на сторінці адміна			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

- Можна додати товар
- Можна перейти до сторінки статистики
- Можна перейти до сторінки заказів

Дії на сторінці статистики

- Можна передивлятися статистику магазину та перейти на головну сторінку.

Дії на сторінці заказів

- Можна передивлятися закази магазину, видаляти їх та перейти на головну сторінку.

2.2 Проектування структури бази даних сайту

Залишилось детальніше розглянути алгоритми роботи основних функцій програми.

Проект складається з двох частин Client та Server. Спочатку розглянемо серверну частину. Почали з того, що встановили всі необхідні для роботи та розробки пакети. Потім створили структуру проекту, папки Controllers – там де лежать контролери, Models - там де лежать моделі, Routes - там де лежать роутери, Services - там де лежать основні функції, Static - там де лежать файли які ми загрузаємо, Middleware - там де лежать функції перевірок вхідних запити.

Після цього ми створили всі моделі, userModel – всі поля які потребує користувач, productModel – поля для товару, tokenModel – поля для зберігання токенів, cartModel – поля для корзини, ratingModel – поля для рейтингу товарів, orderModel – поля для зберігання заказів. Ось ці поля:

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

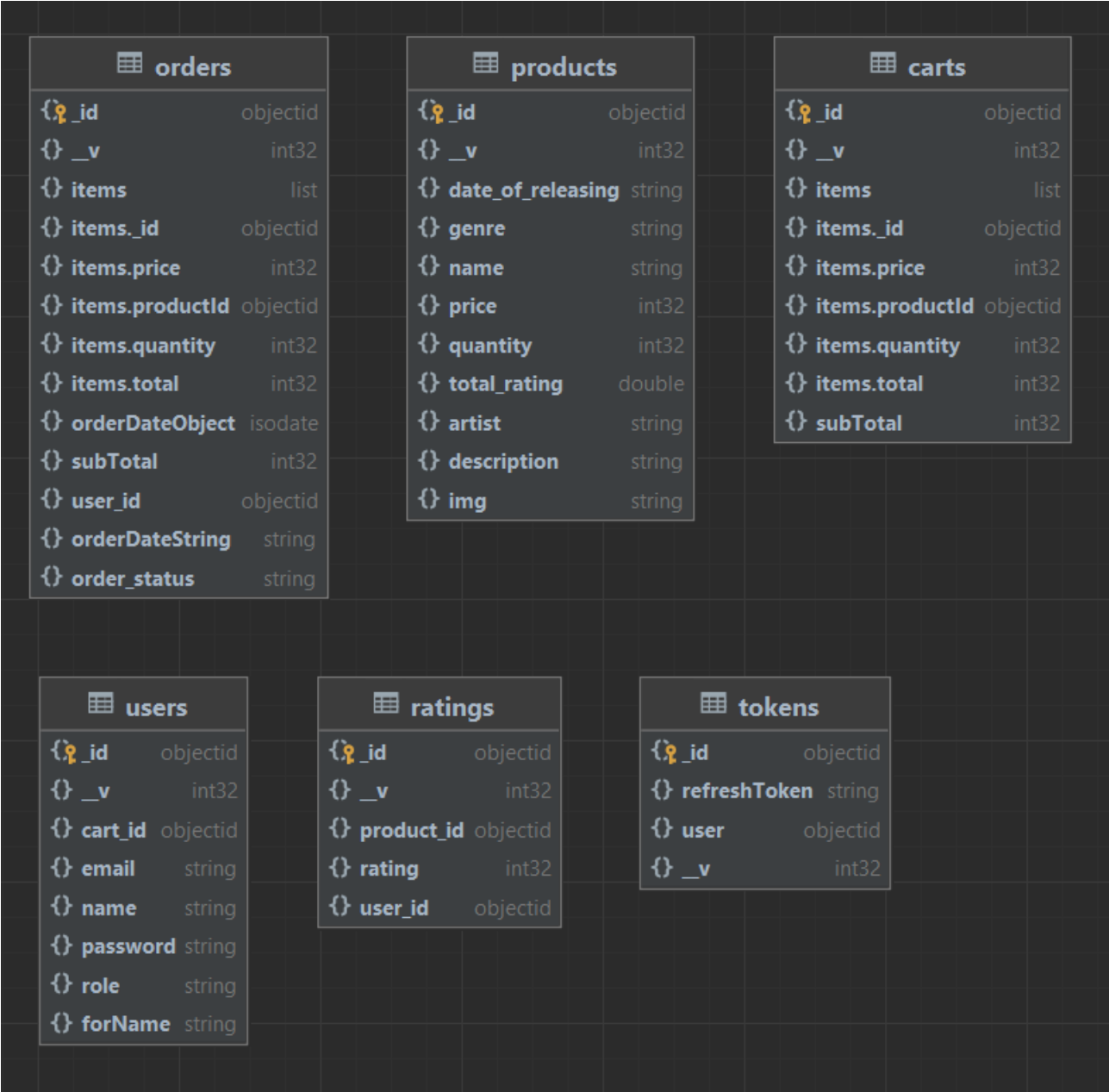


Рис.2.3 Діаграма бази даних

Розглянемо кожну колекцію нашої бази даних. І розберем за який функціонал кожна із них відповідає.

Таблиця 2.1

Структура колекції "users"

Назва			Тип даних		Опис поля			
_id			ObjectId		Id користувача			
name			String		Ім'я			
forname			String		Прізвище			
		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ			Арк.
		Кравченко С.М.						17
Змн.	Арк.	№ докум.	Підпис	Дата				

email	String	Електронна адреса вона ж логін користувача
password	String	Пароль в захешованому вигляді
cart_id	ObjectId	Id корзини
role	String	Ролі користувача

Колекція «users» відповідає за користувачів. Вона буде використовуватись під час реєстрації користувачів, та їхньої аунтифікації.

Таблиця 2.2

Структура колекції "tokens"

Назва	Тип даних	Опис поля
_id	ObjectId	Id токена
user	ObjectId	Id користувача якому належить токен.
refreshToken	String	Рефреш токен користувача.

Колекція «tokens» відповідає за токени. Це частина просунутої авторизації користувачів, де один із токенів зберігається в БД.

Таблиця 2.3

Структура колекції "carts"

Назва	Тип даних	Опис поля
_id	ObjectId	Id кошика
subTotal	Int32	Загальна вартість обраних товарів.
items	Array	Список обраних товарів.
items.id	ObjectId	Id продукту
items.price	Int32	Ціна продукту
items.productId	Int32	Id товару
items.quantity	Int32	Кількість товару
items.total	Int32	Вартість товару у корзині

Колекція «carts» відповідає за кошики. Варто зазначити що в масиві items товари зберігаються в оболонковому.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.4

Структура колекції "orders"

Назва	Тип даних	Опис поля
_id	ObjectId	Id замовлення
user_id	ObjectId	Id користувача якому належить кошик.
orderDateString	String	Загальна вартість обраних товарів.
orderDateObject	Date	Список обраних товарів.
items	List	Список обраних товарів.
order_status	String	Статус замовлення
subTotal	Int32	Загальна вартість замовлення.

Колекція «orders» відповідає за замовлення. В масиві items товари зберігаються в аналогічному вигляді як у кошику. Дата замовлення зберігається в двох форматах, це потрібно для виводу дати в зрозумілому для користувача форматі, і для сортування та формування статистики.

Таблиця 2.5

Структура колекції "product"

Назва	Тип даних	Опис поля
_id	ObjectId	Id товару
date_of_realeasing	String	Дата випуску альбому
genre	String	Жанр
name	String	Ім'я
price	Int32	Ціна товару
quantity	Int32	Кількість товару
total_rating	Double	Рейтинг
artist	String	Артист
description	String	Опис товару.
img	String	Фото

Колекція «products» зберігає данні про товар

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.10

Структура колекції "rating"

Назва	Тип даних	Опис поля
_id	ObjectId	Id відгуку
user_id	ObjectId	Id користувача який залишив відгук
product_id	ObjectId	Id товару до якого залишили відгук
rating	Int32	Оцінка

Колекція «reviews» зберігає данні про відгуки.

Висновки до 2 розділу:

Спроектовано схему роботи програми. Детально описано поля колекцій та їх призначення. Описано деякі нюанси роботи програми. Показано структуру проекту. Приведено логічну діаграму бази даних.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ ІНТЕРНЕТ МАГАЗИНУ

3.1 Проектування інтерфейсу обробки даних

У реалізації серверної частини було використано патерн MVC, з написанням RESTapi. Для частини клієнту використовувався React, для асинхронних запитів до сервера використовується Axios, інформація передається в вигляді JSON.

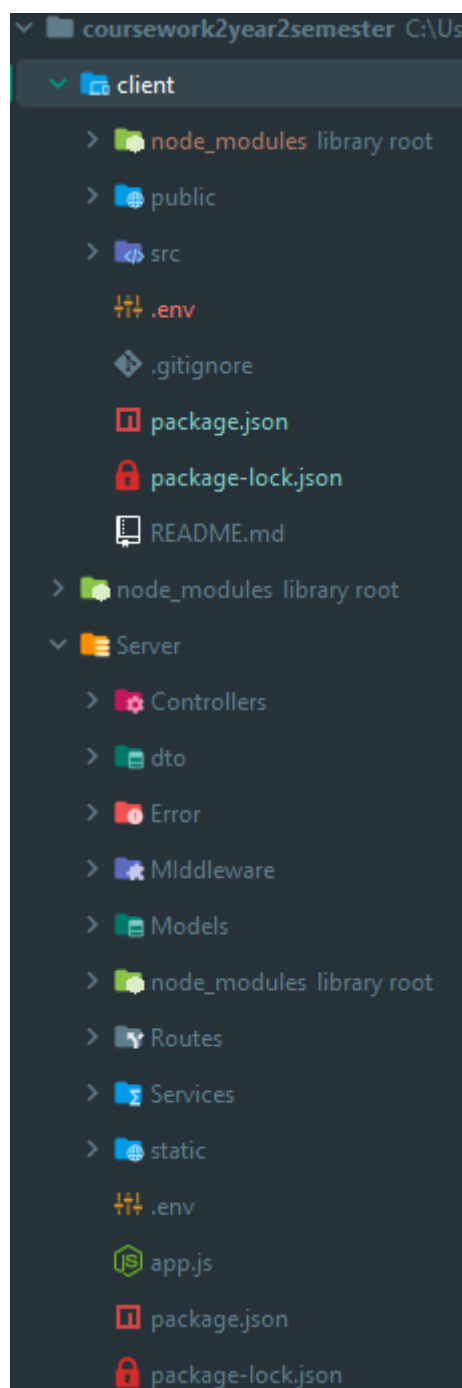


Рис. 3.1. Загальна структура проекту

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

Файлів в проєкті досить багато тому на рисунку лише папки. В сервері головні частини це models – моделі документів бази даних, controllers – контролери які в залежності від запиту викликають потрібну функцію з сервісів, services – сервіси, вони ж вся головна логіка серверу взаємодії з базою даних, формуванню відповіді, валідація тощо, routes – описують маршрути по яким буде відпрацьовувати той чи інший контроллер.

З початку користувач потрапляє на головну сторінку (рис. 3.1), де він може переглядати товари, фільтрувати товари, переглянути інформацію про товар (рис 3.2) та авторизуватись. (рис 3.3) .

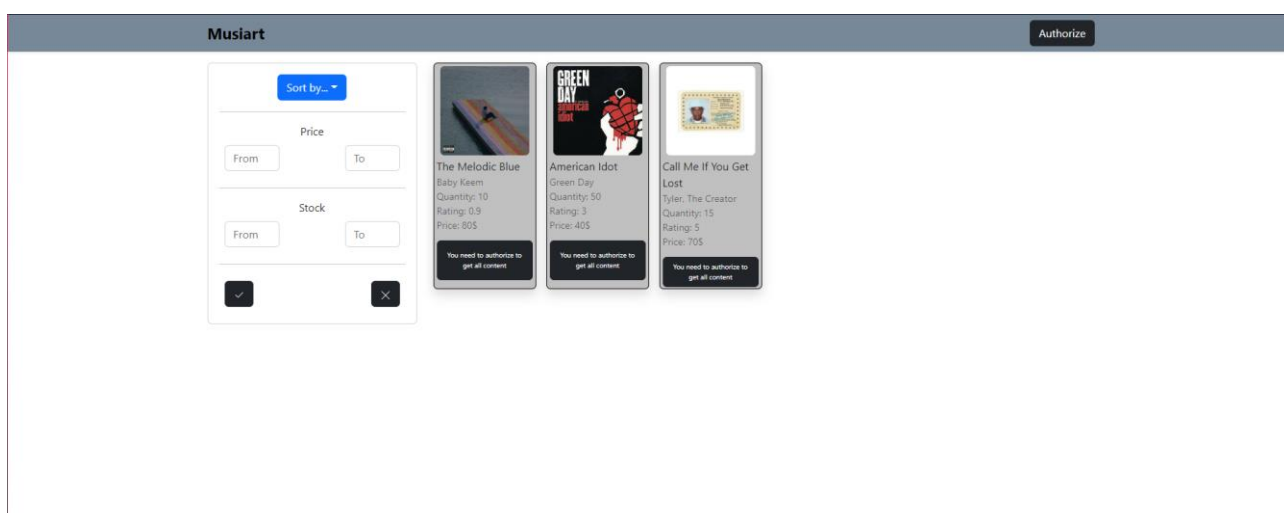


Рис. 3.2. Головна сторінка



Рис. 3.3. Інформація про товар.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

Musiart

Authorize

Registration

Enter name

Enter forename

Enter email

Enter password

You have account [Log In](#)

Sign Up

Рис. 3.4. Сторінка авторизації.

Натомість, коли користувач авторизований з'являється кнопка “корзина”(рис 3.4), и под кожним товаром кнопка “додати до корзини.

Musiart

Cart Log out

Cart

Call Me If You Get Lost

Count: 1

Price 70 \$

Total: 70.00 \$

delete add

Total: 70 \$

Order

Рис. 3.5. Сторінка корзини

Musiart

Admin Button Cart Log out

Add item

Orders

Statistic

Рис. 3.6. Сторінка адміністратора

Якщо ккористувач адміністратор в нього з`являється сторінка адміністратора (рис 3.6), в якій він може переглянути статистику та прцювати з даними на сервері.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.7.Панель додавання товару.

Musiart		Admin Button	Cart	Log out
Order Number - 62e11b62129fd6d77acc33a3		27.07.2022, 14:02:35		
Order Status - Is Processed				
Orders price - 100\$		Delete		
Order Number - 62e02b9eec3114a9fb858e35		26.07.2022, 20:58:55		
Order Status - Is Processed				
Orders price - 80\$		Delete		
Order Number - 62e02d0eec3114a9fb858e64		26.07.2022, 20:58:55		
Order Status - Is Processed				
Orders price - 82\$		Delete		
Order Number - 62e02aab76b0023d7f7124b0		26.07.2022, 20:40:57		
Order Status - Is Processed				
Orders price - 80\$		Delete		
Order Number - 62e02afa76b0023d7f7124e1		26.07.2022, 20:40:57		
Order Status - Is Processed				
Orders price - 160\$		Delete		

Рис. 3.9.Сторінка заказів.

Musiart		Admin Button	Cart	Log out
Statistic				
Top 5 per week				
Sold	Photo	Name		
1		The Melodic Blue		
Number of orders by week :		5		
Number of orders per month :		5		
Total income per day :		502\$		

Рис. 3.9.Сторінка статистики.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Сайт має зручний та інтуїтивно зрозумілий інтерфейс та не створює сильного навантаження на користувача, що є його перевагою над деякими конкурентами.

3.2 Реалізація операцій обробки даних з БД

Для реалізації проекту було використано СУБД MongoDB. Сервер БД було хмарний. Для написання використовувалась програма Webstorm. Більшість операцій були виконані у близькому до ідентичного варіанту, тому демонстрація коду буде лише частковою.

Розглянемо підключення до БД:

```
require('dotenv').config()
const express = require('express')

const start = async () => {
  try{
    await connect(process.env.DATA_BASE)
    app.listen(PORT, () => console.log(`Server started on PORT ${PORT}`))
  }catch (e) {
    console.log(e)
  }
}
start()
```

Крім підключення до бази в файлі наявний і інший код, який потрібний для роботи сервера, вставляти його в звіт немає сенсу.

Для роботи з БД використовується фреймворк Mongoose. Опис документів відбувається в моделях.

Розглянемо це на прикладі моделі Product.

```
const mongoose = require('mongoose');

const productSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
    unique: true
  },
  genre: {
    type: String,
    required: true
  },
  artist: {
    type: String,
    required: true
  },
  date_of_releasing: {
    type: String,
    required: true
  },
})
```

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    price: {
      type: Number,
      required: true
    },
    description: {
      type: String,
      required: true
    },
  },

  quantity: {
    type: Number,
    required: true,
    default: 0
  },
  img: {
    type: String,
    required: true
  },
  total_rating: {
    type: Number,
    default : 0
  },
},
}))

module.exports = mongoose.model('Product', productSchema)

```

Для пошуку даних у документі використовуються різні функції в залежності від ситуації. Розглянемо це на прикладі пошуку кандидата при створенні продукту та фільтрацію товарів за певними параметрами.

```

async create(name, genre, artist, date_of_releasing, price, description, quantity, img,
total_rating) {
  const candidate = await productModel.findOne({name: name})
  if (candidate) {
    throw ApiError.BadRequest('This product already exist')
  }
  return await productModel.create({name, genre, artist, date_of_releasing, price,
description, quantity, img, total_rating})
}

```

```

async getAll(params) {
  let filter = { }

  if(params.minPrice && params.maxPrice){
    filter.price = {$gte: +params.minPrice, $lte: +params.maxPrice}
  }
  if (params.minStock && params.maxStock){
    filter.quantity = {$gte: +params.minStock, $lte: +params.maxStock}
  }
  if(params.name){
    filter.name = {$regex: params.name, $options:'i'}
  }

  let parameters = {};
  if(+params.page === 1){
    parameters = await this.getParams(filter)
  }
  let pipeline = [
    { $match : filter },

```

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    { $project: { _id: 1, name: 1, genre: 1, artist: 1, price: 1, img: 1, quantity: 1,
total_rating: 1 } },
  ]
  if(params.sort_rating && (+params.sort_rating === 1 || +params.sort_rating === -1)){
    pipeline.unshift({$sort: {total_rating: +params.sort_rating} })
  }
  if(params.sort_price && (+params.sort_price === 1 || +params.sort_price === -1)){
    pipeline.unshift({ $sort: {price: +params.sort_price} })
  }

  const products = await productModel.aggregate(pipeline)
  return {products, parameters}
}

```

Для керування доступу створено потрібний checkRoleMiddleware.

```

const tokenService = require('../Services/tokenService')
const ApiError = require('../Error/apiError')

module.exports = function (role)
{
  return function (req, res, next) {
    try{
      const accessToken = req.headers.authorization.split(' ')[1];
      const userData = tokenService.validateAccessToken(accessToken);
      if(userData.role !== role){
        return next(ApiError.Forbidden());
      }
      next();
    }catch (e) {
      return next(ApiError.UnauthorizedError());
    }
  };
}

```

В цьому middleware ми перевіряємо роль користувача. одразу отримуємо всі параметри які має користувач, і далі перевіряємо чи є серед них потрібний дозвіл, якщо так передаємо керування далі, якщо ні повертаємо помилку.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3 Організація звітності системи

Для адміністратора доступний додатковий функціонал в тому числі статистика магазину. В нашому випадку це 5 найпопулярніших товарів, кількість замовлення за тиждень, за місяць та загальна сума проданих товарів за день. Дані виводяться в таблицю на сторінці статистики (Рис. 3.10). Також адміністратор може продивлятися всі заклади на сайті на сторінці закладів (Рис 3.11).

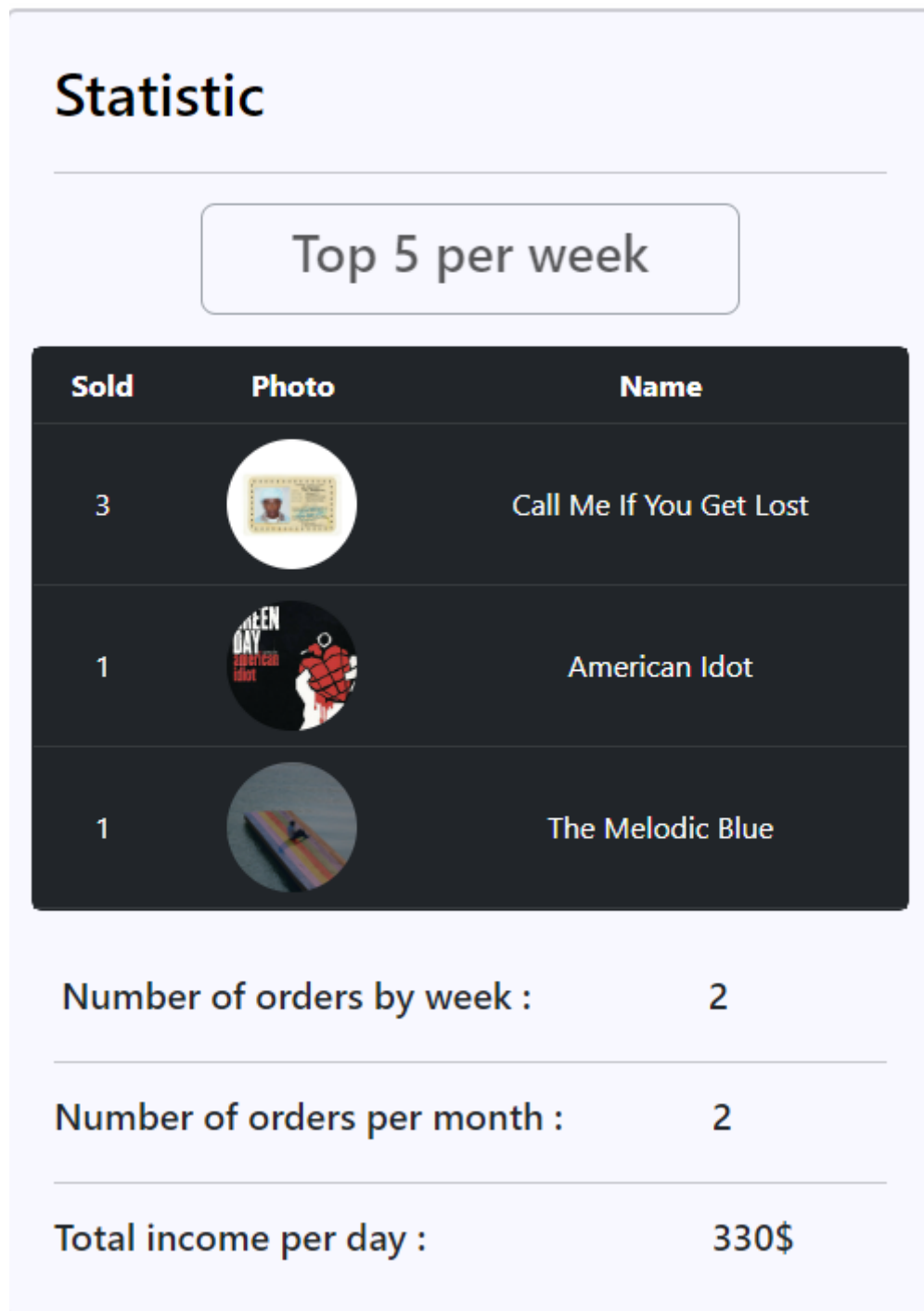


Рис. 3.10. Сторінка статистики.

Musiart		Admin Button	Cart	Log out
Order Number - 62e156c3469fdae23094e9		27.07.2022, 14:04:00		
Order Status - Is Processed				
Orders price - 250\$		Delete		
Order Number - 62e156c7469fdae23094fb		27.07.2022, 14:04:00		
Order Status - Is Processed				
Orders price - 80\$		Delete		

Рис. 3.11. Сторінка заказів.

Висновки до 3 розділу

Було детально описано роботу сайту, розписаний весь функціонал сайту та було продемонстровано його роботу відповідними рисунками, дії користувача та їх наслідки.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ

4.1 Розробка заходів захисту інформації в БД

Для доступу до бази даних було створено систему користувачів з двома ролями. Головна роль ADMIN має необмежені права. Користувач що має цю роль має повний доступ до всіх методів. Також має можливість видаляти товари та заклади.

Нижче приведено матрицю доступу:

0 – немає доступу

1 – читання

2 – вставка

3 – редагування

4 – видалення

5 – повний доступ

Таблиця 4.1

	Користувач	Адміністратор
carts	5	5
methods	0	5
orders	0	1, 4
products	1	1, 2, 4
reviews	1, 2	1, 2
tokens	0	0

4.2 Налаштування параметрів роботи в MongoDB

Для реалізації даного проекту нам потрібна база даних і місце де вона буде зберігатися. Для цього було вирішено розмістити базу даних в MongoDB Atlas. Це всесвітня служба хмарних баз даних для сучасних додатків, яка забезпечує найкращу у своєму класі автоматизацію, а перевірені практики гарантують доступність, масштабованість та відповідність найвищим стандартам безпеки даних та конфіденційності. Для початку роботи з MongoDB Atlas нам потрібно створити особистий акаунт, після входу ми створюємо відповідну організацію в якій розташовуємо наш проект.(рис 4.2)

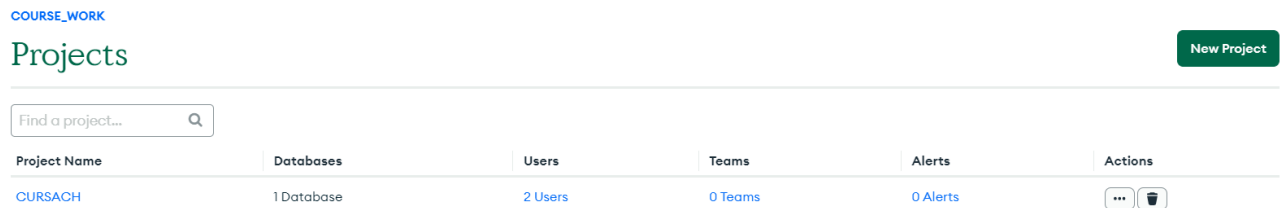
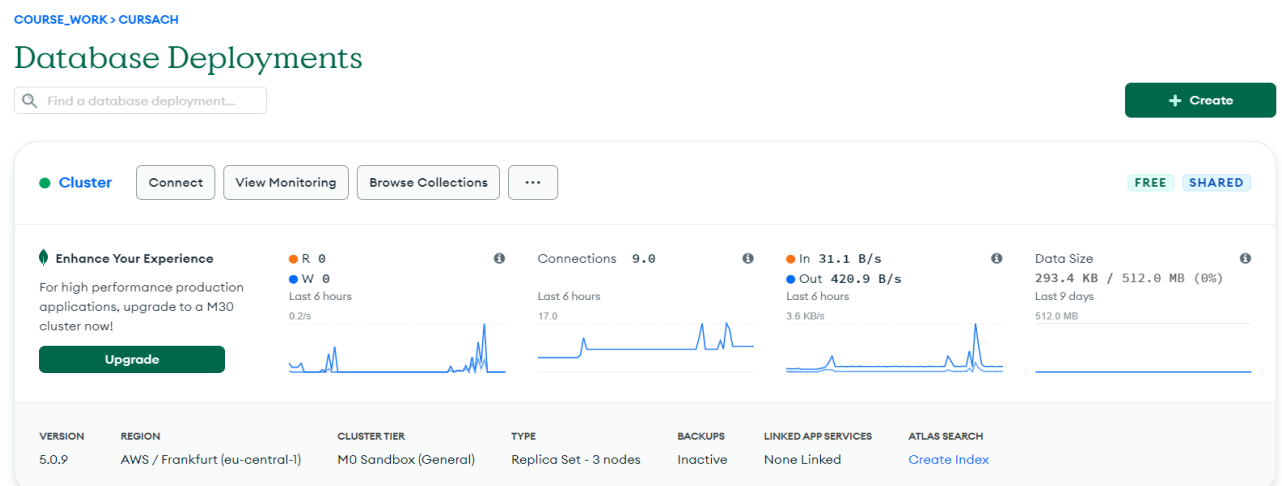


Рис. 4.2. Створений проект

Далі ми створюємо кластер який дозволить базі даних MongoDB здійснювати або горизонтальне масштабування на багатьох серверах з використанням шардингу, або копіювати дані, забезпечуючи високу доступність за допомогою наборів реплік MongoDB, таким чином підвищуючи загальну продуктивність і надійність кластера MongoDB. Тобто кластер це група серверів де буде розміщуватись наша база даних.



		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис 4.3 Кластер MongoDB

Далі нам потрібно встановити айпі адресу підключення та створити користувача який матиме доступ до бази даних

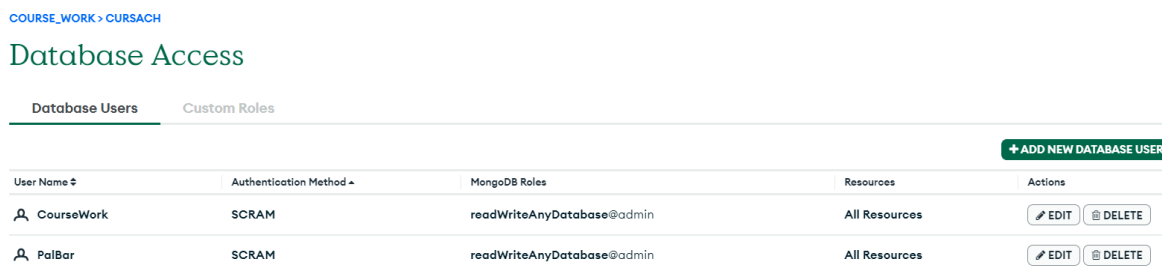


Рис 4.4 Створений користувач



Рис 4.5 Створена айпі адреса

Далі нам потрібно натиснути на кнопку «COLLECTIONS» де ми і створимо нашу базу даних.

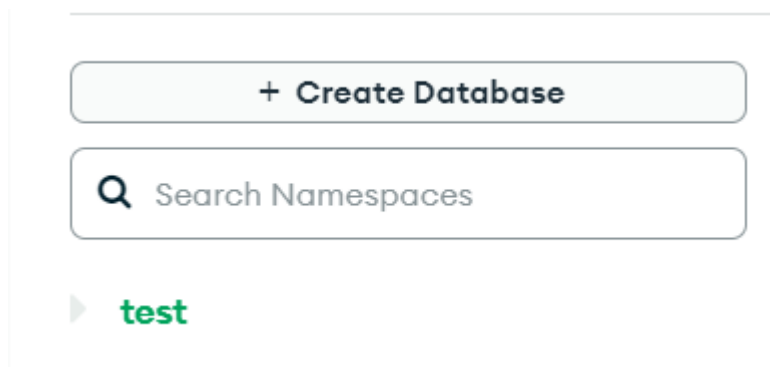


Рис 4.6 Створення бази даних

Нам залишається підключити базу даних в проект і ми готові працювати з нею. Це робиться за допомогою кнопки «CONNECT» де буде запропоновано декілька способів підключення, так як нас цікавить підключення в наш додаток ми вибираємо відповідний пункт, далі буде сформовано посилання за допомогою якого і буде здійснюватись підключення до бази даних.

Для підключення потрібно замінити елементи посилання (Рис 4.7) myFirstDatabase на відповідні дані, а саме на пароль користувача, а myFirstDatabase на назву бази даних до якої ми будемо підключатися.

Connect to Cluster

✓ Setup connection security ✓ Choose a connection method Connect

1 Select your driver and version

DRIVER: Node.js VERSION: 4.1 or later

2 Add your connection string into your application code

☐ Include full driver code example

```
mongodb+srv://<username>:<password>@cluster.vhqkf.mongodb.net/?
retryWrites=true&w=majority
```

Replace **<password>** with the password for the **<username>** user. Ensure any option params are [URL encoded](#).

Having trouble connecting? [View our troubleshooting documentation](#)

Go Back Close

Рис 4.7 Згенероване посилання для підключення до бази даних

Висновки до 4 розділу

Було використано засоби захисту інформації в БД створенням різних ролей. Розкрито можливості кожної групи користувачів та їх відмінності.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

При виконанні курсового проекту у першому розділі досліджувалися потреби та задачі проекту, визначено переваги та недоліки різних СУБД.

Далі було сформовано загальний алгоритми роботи програми, продумані сторінки сайту та можливі дії користувача. Було сформовано алгоритм розподілу ролей між користувачами. Була спроектована база даних.

Було спроектовано інтерфейс обробки даних, що дозволяє нам спостерігати його роботу додатку на практиці. Спроектований інтерфейс обробки даних що дозволяє нам дуже зручно і просто взаємодіяти з додатком.

Було детально описано роботу програми, очікувані дії користувача та їх наслідки.

Створено звіт та презентацію, і завершено проект. В результаті було отримано готовий проект який реалізовує всі поставлені задачі.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mongoose [Електронний ресурс] – Режим доступу до ресурсу: <https://www.npmjs.com/package/mongoose>.
2. MongoDB документація [електронний ресурс] / MongoDB – 2022. – Режим доступу до ресурсу: <https://docs.mongodb.com/manual/>
3. ReactJS документація [електронний ресурс] / ReactJS Tutorial – 2022. – Режим доступу до ресурсу: <https://reactjs.org/tutorial/tutorial.html>
4. Керівництво по NodeJS [електронний ресурс] / Metanit – 2022. . – Режим доступу до ресурсу: <https://metanit.com/web/nodejs/>
5. Електронний посібник «Web Development with Node & Express», 2018.
Режим доступу до ресурсу:
https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.vanmeggern.de/fileadmin/user_upload/PDF/Web_Development_with_Node_Express.pdf&ved=2ahUKEwiNrYrMv7j1AhUXKEQIHQdoAhIQFnoECAQQAQ&sqi=2&usg=AOvVaw3lQb9HWBJLFj54HrM0V4jZ
6. Express/Node introduction js [Електронний ресурс]. Режим доступу до ресурсу: https://developer.mozilla.org/ru/docs/Learn/Serverside/Express_Nodejs/Introduction
7. React-Bootstrap документація [електронний ресурс] / React-Bootstrap Documentation – 2022 - Режим доступу до ресурсу: <https://react-bootstrap.github.io/getting-started/introduction/>

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

ТЕХНІЧНЕ ЗАВДАННЯ

1. Загальне положення**1.1. Найменування програмного засобу**

Повне найменування програмної системи: «База даних Інтернет магазину вінілових платівок» (надалі "програма"). Коротка назва програмної системи – «Musiart».

1.2. Призначення розробки та область застосування

Програмна система "База даних інтернет-магазину вінілових платівок" призначена для відображення даних про товари за тематикою «вінілових платівок». В програмі передбачено використання механізму фільтрації та сортування по параметрам товарів.

Система може бути впроваджена у роботу офлайн-магазину для збільшення клієнтської бази.

1.3. Мета

Програмна система "База даних інтернет-магазину вінілових платівок" дозволить підвищити ефективність, повноту та швидкість роботи працівників магазину.

1.4. Найменування розробника та замовника.

Розробник даного продукту – студент групи ІПЗ-20-2 Палій Іван (надалі "розробник"). Замовник програмного продукту - кафедра інженерії програмного забезпечення дуржавного університету «Житомирська політехніка» в межах виконання курсової з дисципліни «Бази даних» Сугоняк Інна Іванівна.

2. Підстава для розробки**2.1. Документ на підставі якого ведеться розробка**

Робота ведеться на підставі навчального плану за напрямом 121 «Інженерія програмного забезпечення».

3. Вимоги до програми**3.1. Вимоги до функціональних характеристик.****3.1.1. Загальні вимоги**

		Програма має забезпечувати:			Арк.
		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ
		Кравченко С.М.			
Змн.	Арк.	№ докум.	Підпис	Дата	
					37

- можливість дистанційної роботи з робочих станцій локальної та глобальної мережі підприємства;
- інтерфейс користувача, що не залежить від операційної системи
- постійний доступ користувачів до БД;
- можливість доступ до бази даних;
- аутентифікацію користувачів та захист інформації від несанкціонованого доступу;

3.1.2. Склад виконуваних функцій

Розробити базу даних, що використовується у веб-сайті, який підтримує виконання наступних операцій:

1. Перегляд усіх даних, згрупованих по категоріям
2. Пошук по багатьом параметрам
3. Авторизація користувача
4. Якщо користувач є адміністратором – з'являється доступ до режиму адміністрування
5. Вставка, перегляд та видалення товарів та категорій.
6. Зберігання даних про замовлення користувачів
7. Формування звітності щодо наявних даних. Перелік звітних форм наступний:

- Звіт про дохід магазину за заданий термін
- Звіт про кількість замовлень за задний термін
- Звіт про 5 найзамовлюваних товарів

3.1.3. Організація вхідних і вихідних даних

Вхідними даними є інформація про користувача та інформація, що вноситься в систему при додаванні або зміні значень колекцій.

Організація вхідних і вихідних даних повинна відповідати інформаційній структурі виконуваних з операцій, вхідним та вихідним паперовим документами.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

Введення оперативних даних повинно виконуватися з використанням діалогових екранних форм, побудованих на основі візуальних компонентів.

3.1.4. Часові характеристики і розмір пам'яті, необхідної для роботи програми.

Час реакції програми на дії користувача (маніпуляції з пристроями введення даних) не повинен перевищувати 0,25 с.

Час виконання команд меню не більше 1 с.

Відображення масивів даних за запитами не більше 3 хвилин.

Доступність БД – 90% цілодобово.

Операції з'єднання з БД не більше 1 хвилини. Обсяг оперативної пам'яті, необхідний для роботи програми не менше 2Гб.

Надійне збереження даних та можливість відновлення даних у випадку непередбачуваних збоїв системи;

3.2. Вимоги до надійності.

3.2.1. Вимоги до надійного функціонування

Програма повинна нормально функціонувати при безперебійній роботі ПК. Доступність БД 90% при одночасному доступі 30 користувачів. При апаратних збоях, відновлення нормальної роботи програми повинне виконуватися після:

а) апаратні збої сервера - перезавантаження ОС сервера, запуск сервера БД (запуску резервного сервера, використання технологій RAID для збереження даних);

б) апаратні збої робочої станції – перезавантаження ОС ПК, запуск виконуваного файлу програми.

в) БД повинна повертатись в найближчий несуперечний стан – передбачити точки відновлення.

При збоях програмного забезпечення:

г) система повинна забезпечувати можливість відновлення даних та фіксацію і «відкат» транзакцій.

д) в системі має бути реалізована коректна обробка виняткових ситуацій.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2.2.Контроль вхідної і вихідної інформації

Для контролю коректності вхідної інформації та захисту від помилок оператора:

- Перевірка відповідності даних доменам інформаційних атрибутів;
- Використанням механізму авто заповнення та вибору за переліком для зв'язаних даних;
- Захист від помилок оператора (залипань, випадкових символів тощо).

Визначені некоректні дії повинні супроводжуватись повідомленнями про помилку і блокуванням операцій оновлення даних. В системі має бути передбачений захист від загального блокування.

Для вихідної інформації передбачити:

- відсутність блокування даних через багато користувальницький доступ;
- постійне оновлення даних у відображених на екрані звітних формах.
- повідомлення про використання режиму монопольного доступу супервайзерами БД.

3.2.3. Час відновлення після відмови.

Час відновлення після відмови, не пов'язаною з роботою програми, повинен складатися із: часу перезапуску операційної системи; часу запуску сервера БД (підключення до сервера) запуску виконуваного файлу, часу повторного введення або зчитування даних.

3.3. Вимоги до інформаційної і програмної сумісності.

3.3.1. Вимоги до методів рішення і мов програмування

Вибір методів рішення здійснюється розробникам без узгодження з замовником. СУБД обирається у відповідності до характеристик визначених в п.3. З замовником погоджується вибір варіанту за вартісною ознакою.

3.3.2. Вимоги до системи програмних засобів.

Вимоги до програмного забезпечення сервера: ОС – серверна версія ОС Windows не нижче Windows 2000 Server, IIS (або інший сервер),

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

СУБД визначається встановлюється та налаштовується розробником (ліцензування СУБД виконується замовником).

Вимоги до програмного забезпечення робочої станції:

Всі найновіші, стабільні релізи браузерів и платформ.

3.4. Вимоги до складу і параметрів технічних засобів.

Вимоги до складу технічних засобів:

1. Сервер:

- сервер у базовій конфігурації із підтримкою RAID;
- засоби для під'єднання до локальної мережі.

2. Робоча станція:

- ПК на базі процесорів Intel, AMD у стандартній комплектації;
- засоби для під'єднання до локальної мережі. засоби для під'єднання до локальної мережі.
- Pentium III з тактовою частотою 750 МГц;
- оперативна пам'ять - 128Мбайт;
- об'єм дискової пам'яті - 20Гбайт

4. Вимоги до програмної документації

Програмна документація повинна включати наступні відомості:

1. "Інструкція по інсталяції (встановленню) програми", складається з опису інсталяційного пакету, переліку етапів інсталяції та їх послідовності, опису встановлених програмних компонентів та режимів їх роботи після інсталяції. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться у 4 розділі.

2. Керівництво користувача" складається з опису послідовності завантаження програми, основних режимів роботи, опису основних екранних форм, переліку виняткових ситуацій та реакції користувача на них, порядку виконання завдань в системі. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться в 3 розділі курсової роботи.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

3. "Керівництво адміністратора баз даних", складається з опису складу таблиць бази даних та правил доступів до них, опису та послідовності робіт по обслуговуванню бази даних (архівування, резервне копіювання, з вказівкою періодичності виконання та засобів, що для цього використовуються), порядку налаштування серверу та клієнтських додатків. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться в 2,3 та 4 розділах курсової роботи згідно плану.

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

productModel

```
const mongoose = require('mongoose');

const productSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
    unique: true
  },
  genre: {
    type: String,
    required: true
  },
  artist: {
    type: String,
    required: true
  },
  date_of_releasing: {
    type: String,
    required: true
  },
  price: {
    type: Number,
    required: true
  },
  description: {
    type: String,
    required: true
  },
  quantity: {
    type: Number,
    required: true,
    default: 0
  },
  img: {
    type: String,
    required: true
  },
  total_rating: {
    type: Number,
    default : 0
  },
},
})

module.exports = mongoose.model('Product', productSchema)
```

productRouter

```
const Router = require('express').Router;
const checkRole = require('../Middleware/checkRoleMiddleware')
const authMiddleware = require('../Middleware/authMiddleware')

const productRouter = new Router();

const productController = require('../Controllers/productController')

productRouter.post('/create', checkRole('ADMIN'), authMiddleware,
productController.create)
productRouter.delete('/delete/:id', checkRole('ADMIN'), authMiddleware,
productController.delete)
productRouter.get('/allproducts', productController.getAll)
productRouter.get('/:id', productController.getOne)
module.exports = productRouter;
```

productController

ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата
		Кравченко С.М.		
				43

```

const productModel = require('../Models/productModel')
const productService = require('../Services/productService')
const uuid = require('uuid')
const path = require('path')

class ProductController {
  async create(req, res, next)
  {
    try {
      const {name, genre, artist, date_of_releasing, price, description, quantity}
= req.body
      const {img} = req.files
      let imgName = uuid.v4() + ".jpg"
      img.mv(path.resolve(__dirname, '..', 'static', imgName))

      const product = await productService.create(name, genre, artist,
date_of_releasing, price, description, quantity, imgName)

      return res.status(200).json(product)
    } catch (e) {
      next(e);
    }
  }
  async getAll(req, res, next)
  {
    try {
      const params = req.query;
      const result = await productService.getAll(params);
      return res.status(200).json(result)
    } catch (e) {
      next(e)
    }
  }
  async getOne(req, res, next)
  {
    try{
      const {id} = req.params
      const product_id = await productService.getOne(id)

      return res.status(200).json(product_id)
    } catch (e) {
      next(e)
    }
  }
  async delete(req, res, next)
  {
    try {
      const {id} = req.params
      const productDel = await productService.delete(id)

      return res.status(200).json(productDel)
    } catch (e) {
      next(e)
    }
  }
}

module.exports = new ProductController();

```

productService

```

const productModel = require('../Models/productModel')
const ApiError = require('../error/ApiError')
class TovarService {
  async create(name, genre, artist, date_of_releasing, price, description, quantity, img,
total rating)

```

```

{
  const candidate = await productModel.findOne({name: name})
  if (candidate) {

```

Арк.

ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ					44
Змн.	Арк.	Кравченко С.М.	Підпис	Дата	

```

        throw ApiError.BadRequest('This product already exist')
    }
    return await productModel.create({name, genre, artist, date_of_releasing, price,
description, quantity, img, total_rating})
}
async getOne(id) {
    if(!id)
    {
        throw ApiError.BadRequest('Id of product wasn`t entered')
    }
    const candidate = await productModel.findById(id)
    if(!candidate){
        throw ApiError.BadRequest('This product doesn`t exist')
    }
    return candidate
}
async delete(id) {
    if(!id)
    {
        throw ApiError.BadRequest('Id of product wasn`t entered')
    }
    const candidate = await productModel.findById(id)
    if(!candidate){
        throw ApiError.BadRequest('This product doesn`t exist')
    }
    return productModel.findByIdAndDelete(id);
}
async getParams(filter) {
    const parameters = await productModel.aggregate([
        {$match : filter},
        {$group : {
            _id: 'parameters',
            maxPrice: {$max: 'price'}, minPrice: {$min: 'price'},
            maxStock: {$max: 'quantity'}, minStock: {$min: 'quantity'},
            pages:{$count : {}}
        }}
    ])

    return parameters[0];
}
async getAll(params) {
    let filter = { }

    if(params.minPrice && params.maxPrice){
        filter.price = {$gte: +params.minPrice, $lte: +params.maxPrice}
    }
    if (params.minStock && params.maxStock){
        filter.quantity = {$gte: +params.minStock, $lte: +params.maxStock}
    }
    if(params.name){
        filter.name = {$regex: params.name, $options:'i'}
    }

    let parameters = {};
    if(+params.page === 1){
        parameters = await this.getParams(filter)
    }
    let pipeline = [
        { $match : filter },
        { $project: {_id: 1, name: 1, genre: 1, artist: 1, price: 1, img: 1,
quantity: 1, total_rating: 1 } },
    ]
    if(params.sort_rating && (+params.sort_rating === 1 || +params.sort_rating === -
1)){
        pipeline.unshift({$sort: {total_rating: +params.sort_rating} })
    }

    if(params.sort_price && (+params.sort_price === 1 || +params.sort_price === -1)){
        pipeline.unshift({ $sort: {price: +params.sort_price} })
    }
}

```

```

    if(params.sort_price && (+params.sort_price === 1 || +params.sort_price === -1)){
        pipeline.unshift({ $sort: {price: +params.sort_price} })
    }
}

```

					Арк.
Кравченко С.М.					45
Змн.	Арк.	№ докум.	Підпис	Дата	

```

        const products = await productModel.aggregate(pipeline)
        return {products, parameters}
    }
}
module.exports = new ТоварService()

```

cartService

```

const cartModel = require('../Models/cartModel')
const userModel = require('../Models/userModel')
const productModel = require('../Models/productModel')
const ApiError = require('../error/ApiError')

class CartService {
    async create(){
        return await cartModel.create({});
    }
    async addToCart(userId, productId)
    {
        const productData = await productModel.findById(productId)
        if(!productData)
        {
            throw ApiError.BadRequest('No product with such id')
        }
        const userData = await userModel.findById(userId)
        if(!userData)
        {
            throw ApiError.BadRequest('No user with such id')
        }
        const cart = await cartModel.findById(userData.cart_id)
        if(!cart)
        {
            throw ApiError.BadRequest('User doesn`t have cart')
        }
        let indexFound = cart.items.findIndex(item => (item.productId == productId))
        if(indexFound === -1)
        {
            cart.items.push({
                productId: productId,
                quantity: 1,
                price: productData.price,
                total: parseInt(productData.price),
            })
        }
        if(indexFound !== -1)
        {
            cart.items[indexFound].quantity += 1
            let quantity = cart.items[indexFound].quantity
            cart.items[indexFound].total = (productData.price * quantity)
        }
        cart.subTotal = 0;
        cart.items.forEach(item => {
            cart.subTotal += item.total;
        })
        return await cart.save()
    }
    async deleteCartElement(userId, productId)
    {
        const userData = await userModel.findById(userId)
        if(!userData)
        {
            throw ApiError.BadRequest('No user with such id')
        }
    }

```

```

        const cart = await cartModel.findById(userData.cart_id)
        if(!cart)
        {
            throw ApiError.BadRequest('User doesn`t have cart')
        }
    }

```

Арк.

```

    }
    let indexFound = cart.items.findIndex(item => (item.productId == productId))
    if(indexFound === -1)
    {
        throw ApiError.BadRequest('Cart don't have this item')
    }
    const productData = await productModel.findById(productId);
    if(!productData){
        cart.items.splice(indexFound, 1);
        cart.subTotal = 0;
        cart.items.forEach(item => {
            cart.subTotal += item.total;
        })
        return await cart.save();
    }

    if(cart.items[indexFound].quantity !== 0){
        cart.items[indexFound].quantity -= 1;
        let quantity = cart.items[indexFound].quantity;
        if(quantity === 0){
            cart.items.splice(indexFound, 1);
            cart.subTotal = 0;
            cart.items.forEach(item => {
                cart.subTotal += item.total;
            })
            return await cart.save();
        }
        cart.items[indexFound].total = (cart.items[indexFound].price * quantity);
    }
    cart.subTotal = 0;
    cart.items.forEach(item => {
        cart.subTotal += item.total;
    })
    return await cart.save()
}
async getAllCartElements(userId)
{
    const userData = await userModel.findById(userId)
    if(!userData)
    {
        throw ApiError.BadRequest('No user with such id')
    }
    const cartData = await cartModel.findById(userData.cart_id)
    if(!cartData)
    {
        throw ApiError.BadRequest('User doesn't have cart')
    }
    return cartData
}
}
module.exports = new CartService()

```

cartModel

```

const mongoose = require("mongoose")

const itemSchema = new mongoose.Schema({
    productId: {
        type: mongoose.Schema.Types.ObjectId,
        ref: "Product"
    },

    quantity: {
        type: Number,
        required: true,

        default: 0
    },

    price: {

```

```

        type: Number,
        required: true
    },

    total: {
        type: Number,
        required: true
    }
})
const cartSchema = new mongoose.Schema({
    items:[itemSchema],

    subTotal: {
        default: 0,
        type: Number
    }
});

module.exports = mongoose.model('Cart', cartSchema);

```

cartController

```

const cartService = require('../Services/cartService');

class CartController {
    async addElementToCart (req, res, next)
    {
        try{
            const {userId, productId} = req.body
            const cartData = await cartService.addToCart(userId, productId)
            return res.status(200).json(cartData)
        }catch (e) {
            next(e)
        }
    }
    async getAll (req, res, next)
    {
        try{
            const {id} = req.params
            const cartData = await cartService.getAllCartElements(id)
            return res.status(200).json(cartData)
        }catch (e) {
            next(e)
        }
    }
    async deleteElementFromCart (req, res, next)
    {
        try{
            const {userId, productId} = req.query

            const cartData = await cartService.deleteCartElement(userId, productId)
            return res.status(200).json(cartData)
        }catch (e) {
            next(e)
        }
    }
}

module.exports = new CartController()

```

		Палій І.В.			ДУ «Житомирська політехніка». 22.121.8.000 - ПЗ	Арк.
		Кравченко С.М.				48
Змн.	Арк.	№ докум.	Підпис	Дата		