

Київський національний 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.
«7» листопада 2019 р.

Завдання на випускню кваліфікаційну роботу студентів

Єрохину Володимиру Олеговичу

1. Тема випускної кваліфікаційної роботи: «Розробка програмного продукту інтернет магазин з продажу комп'ютерної техніки».

Затверджена наказом ректора від «02» грудня 2019 р. № 4112

2. Строк здачі студентом закінченої роботи _____

3. Цільова установка та вихідні дані до роботи:

Мета роботи: аналіз існуючих платформ хмарних середовищ та існуючих СУБД, які використовуються при створенні інтернет-магазинів, розробка веб-додатку з продажу комп'ютерної техніки.

Об'єкт дослідження: діяльність магазину комп'ютерної техніки.

Предмет дослідження: автоматизація роботи магазину з продажу комп'ютерної техніки.

Консультанти роботи із зазначенням розділів, які консультують:

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

4. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 ОГЛЯД ПРОГРАМНИХ РІШЕНЬ, НЕОБХІДНИХ ДЛЯ
ФУНКЦІОНУВАННЯ ІНТЕРНЕТ МАГАЗИНУ З ПРОДАЖУ КОМП'ЮТЕРНОЇ
ТЕХНІКИ

1.1. Аналіз існуючих хмарних середовищ

1.1.1. Amazon

1.1.2. Azure

1.2. Аналіз існуючих СУБД

1.2.1. Oracle

1.2.2. MySQL

1.2.3. Microsoft SQL Server

1.3. Огляд операційних систем для розробки веб-додатку

1.3.1. Операційна система Linux

1.3.2. Операційна система Windows

1.4. Течнічне завдання для розробки інтернет магазину з продажу комп'ютерної
техніки

1.5. Висновки до розділу 1

РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ, ПРОЕКТУВАННЯ ТА НАЛАШТУВАННЯ
РЕСУРСІВ НЕОБХІДНИХ ДЛЯ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ

2.1. Архітектура програмного забезпечення інтернет магазину з продажу
комп'ютерної техніки

2.2. Постановка задачі для розробки бази даних інтернет-магазину з продажу
комп'ютерної техніки

2.2.1. Проектування логічної моделі бази даних інтернет-магазину з продажу комп'ютерної техніки

2.2.2. Фізична модель бази даних інтернет магазину з продажу комп'ютерної техніки

2.3. Формування технічного оточення інтернет магазину з продажу комп'ютерної техніки

2.4. Організація інфраструктури необхідної для функціонування програмного продукту в хмарному середовищі Amazon

2.4.1. Створення віртуальної машини засобами AWS Services

2.4.2 Створення бази даних засобами AWS Services

2.5 Висновки до розділу 2

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ ІНТЕРНЕТ МАГАЗИНУ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ

3.1. Проектування та реалізація графічного інтерфейсу інтернет магазину з продажу комп'ютерної техніки

3.2. Проектування та реалізація серверної частини інтернет магазину з продажу комп'ютерної техніки

3.3. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Додатки

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	Вибір теми випускної кваліфікаційної роботи		
2.	Вступ та перелік літературних джерел		
3.	Розділ 1. Технічне завдання та дослідження необхідних ресурсів для створення програми «інтернет магазин з продажу комп'ютерної техніки»		
4.	Розділ 2. Постановка задачі та моделювання архітектури необхідних ресурсів для інтернет магазину з продажу комп'ютерної техніки		
5.	Розділ 3. Реалізація веб-додатку «інтернет-магазин з продажу комп'ютерної техніки»		
6.	Висновки		
7.	Здача випускної кваліфікаційної роботи на кафедрі (перша перевірка)		
8.	Підготовка автореферату та презентації доповіді		
9.	Попередній захист випускної кваліфікаційної роботи		
10.	Зовнішнє рецензування випускної кваліфікаційної роботи		
11.	Здача прошитої випускної кваліфікаційної роботи на кафедрі		
12.	Публічний захист випускної кваліфікаційної роботи		

7. Дата видачі завдання «__» _____ 20__ р.

8. Науковий керівник випускної кваліфікаційної роботи

Рзаєва С. Л.

9. Гарант освітньої програми

Цензура М. О.

10. Завдання прийняв до виконання студент

Срохін В. О.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

(*nidnuc*, *data*)

РЗВЄВА С. Л.

(ПИБ, *нідпис*, *дата*)

Випускна кваліфікаційна робота студента Єрохіна В. О. може бути допущена до захисту експертною комісією.

Цензура М. О.

Криворучко О. В.

« » 20 p.

Анотація

Дипломна робота на тему розробка інтернет-магазину з продажу комп'ютерної техніки містить 56 сторінок, 15 рисунків та 1 додаток. Перелік посилань налічує 24 найменування.

Мета дослідження: аналіз існуючих платформ хмарних середовищ та існуючих СУБД, які використовуються при створенні інтернет-магазинів, розробка веб-додатку з продажу комп'ютерної техніки.

Об'єкт дослідження: діяльність магазину комп'ютерної техніки.

Предмет дослідження: автоматизація роботи магазину з продажу комп'ютерної техніки.

У першому розділі складено технічне завдання для створення інтернет магазину з продажу комп'ютерної техніки, оглянуто існуючі хмарні середовища, системи управління базами даних та операційні системи, все те, що необхідно для побудови програмного продукту.

У другому розділі спроектовано логічну та фізичну моделі для бази даних, організовано інфраструктуру у хмарному середовищі Amazon, для функціонування веб додатку, налаштовано середовище для запуску віртуальної машини, на якій буде хоститися додаток та під'єднано його до бази даних MySQL.

У третьому розділі спроектовано макети веб сторінок для графічного інтерфейсу інтернет магазину, розроблено сам графічний інтерфейс за створеними макетами та створено серверну частину за встановленими рекомендаціями.

Annotation

Graduate work on the development of an online store for the sale of computer equipment contains 56 pages, 15 images and 1 attachment. The list of links includes 24 names.

The purpose of the study: analysis of existing cloud platforms and existing databases used in the creation of online stores, development of a web application for the sale of computer equipment.

Object of research: the activities of a computer store.

Subject of research: automation of the store selling computer equipment.

The first section outlines the terms of reference for creating an online store selling computer equipment, examines the existing cloud environments, database management systems and operating systems, everything you need to build a software.

The second section designs the logical and physical models for the database, organized the infrastructure in the Amazon cloud environment in which runs the web application, configured the environment to run the virtual machine that will host the application, and connected it to the MySQL database.

In the third section, web page layouts are designed for the graphical interface of the online store, the graphical interface itself is developed according to the created layouts and the server part is created according to the established recommendations.

Київський національний торговельно-економічний університет

Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

*«Розробка програмного продукту інтернет магазин з продажу
комп'ютерної техніки»*

Студента 4 курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»

Єрохіна Володимира
Олеговича

Науковий керівник
кандидат технічних наук,
доцент

Рзаєва Світлана
Леонидівна

Гарант освітньої програми
кандидат технічних наук,
доцент

Цензура Микола
Олександрович

КИЇВ — 2020

Зміст

Зміст	2
ВСТУП.....	5
Розділ 1 ОГЛЯД ПРОГРАМНИХ РІШЕНЬ, НЕОБХІДНИХ ДЛЯ ФУНКЦІОНУВАННЯ ІНТЕРНЕТ МАГАЗИНУ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ.....	7
1.1 Аналіз існуючих хмарних середовищ.....	7
1.1.1 Amazon	7
1.1.2 Azure	9
1.2 Аналіз існуючих СУБД.....	10
1.2.1 Oracle.....	10
1.2.2 MySQL	11
1.2.3 Microsoft SQL Server	13
1.3 Огляд операційних систем для розробки веб-додатку.....	14
1.3.1 Операційна система Linux	14
1.3.2 Операційна система Windows	14
1.4 Технічне завдання для розробки інтернет магазину з продажу комп'ютерної техніки	15
1.5 Висновки до розділу 1	17
РОЗДІЛ 2 ПОСТАНОВКА ЗАДАЧІ, ПРОЕКТУВАННЯ ТА НАЛАШТУВАННЯ РЕСУРСІВ НЕОБХІДНИХ ДЛЯ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ	19

					<i>КНТЕУ 121 07-21.БР</i>			
					Розробка інтернет магазину з продажу комп'ютерної техніки	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум	Підпис	Дата		3	2	56
Зав. кафедри	Криворучко О.В.					Факультет інформаційних технологій, 4 курс, 6 група		
Керівник	Рзаєва С.Л.							
Гарант	Цензура М.О.							
Розроб.	Єрохін В.О.				Зміст			

2.1. Архітектура програмного забезпечення інтернет магазину з продажу комп'ютерної техніки	19
2.2. Постановка задачі для розробки бази даних інтернет-магазину з продажу комп'ютерної техніки	19
2.2.1. Проектування логічної моделі бази даних інтернет-магазину з продажу комп'ютерної техніки	20
2.2.2. Фізична модель бази даних інтернет-магазину з продажу комп'ютерної техніки	22
2.3. Формування технічного оточення інтернет-магазину з продажу комп'ютерної техніки	24
2.4. Організація інфраструктури необхідної для функціонування програмного продукту в хмарному середовищі Amazon	25
2.4.1 Створення віртуальної машини засобами AWS Services	26
2.4.2 Створення бази даних засобами AWS Services	28
2.5 Висновки до розділу 2.....	29
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ	30
3.1. Проектування та реалізація графічного інтерфейсу інтернет магазину з продажу комп'ютерної техніки	30
3.2. Проектування та реалізація серверної частини інтернет магазину з продажу комп'ютерної техніки	38
3.3 Висновки до розділу 3.....	42
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	43
Список використаних джерел.....	45
Додатки	48

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		3

					КНТЕУ 121 06-15.БР	Аркуш
						4
Зм.	Аркуш	№ докум	Підпис	Дата		

ВСТУП

Актуальність. На сьогоднішній день майже всі споживачі будь-яких товарів та послуг, все більше віддають перевагу придбанню продукції та замовленню послуг через мережу Інтернет. По-перше це зручно, тому що можна в будь-яку зручну годину зробити замовлення та відправити його на обробку, а представник компанії вже зможе обробити це замовлення у робочий час. По-друге інтернет замовлення можна зробити із будь якої точки світу і відповідно замовляти доставку у будь яке зручне місце.

Автоматизація допомагає збільшити продуктивність підприємств та установ, оскільки кількість роботи в одиницю часу яку може зробити машина значно перевищує продуктивність людини. Збільшення продуктивності відповідно призводить до отримання більших прибутків, що і є головною метою кожного підприємця.

Саме тому запропоновано автоматизувати процес продажу комп'ютерної техніки, оскільки саме комп'ютери дозволяють створювати інноваційні продукти, та модернізувати сучасний світ. Отже можна із впевненістю сказати, що попит на цей програмний продукт буде великий.

Метою дослідження є аналіз існуючих платформ хмарних середовищ та існуючих СУБД, які використовуються при створенні інтернет-магазинів, розробка веб-додатку з продажу комп'ютерної техніки.

Об'єкт дослідження – діяльність магазину комп'ютерної техніки.

Предмет дослідження – автоматизація роботи магазину з продажу комп'ютерної техніки.

Завдання випускної кваліфікаційної роботи:

					КНТЕУ 121 07-21.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка інтернет магазину з продажу комп'ютерної техніки	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					В	5	56
Керівник	Рзаєва С.Л.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М.О.							
Розроб.	Єрохін В.О..							
					Вступ			

- аналіз СУБД для розробки веб-додатку з продажу комп'ютерної техніки;
- аналіз доступних платформ для публікації веб-додатку у хмарному середовищі;
- вибір операційної системи для побудови веб-додатку;
- опис технічного завдання для програмного забезпечення;
- розробка бази даних інтернет-магазину з продажу комп'ютерної техніки;
- проектування графічного інтерфейсу інтернет магазину з продажу комп'ютерної техніки;
- проектування серверної частини інтернет магазину з продажу комп'ютерної техніки;
- розробка веб-додатку з продажу комп'ютерної техніки.

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		6

Розділ 1

ОГЛЯД ПРОГРАМНИХ РІШЕНЬ, НЕОБХІДНИХ ДЛЯ ФУНКЦІОНУВАННЯ ІНТЕРНЕТ МАГАЗИНУ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ

1.1 Аналіз існуючих хмарних середовищ

1.1.1 Amazon

У 2006 році Amazon (AWS) почав пропонувати такі послуги ІТ-інфраструктури для бізнесу, як веб-сервіси - тепер загальновідомі як хмарні обчислення. Однією з ключових переваг хмарних обчислень є можливість замінити передові капітальні витрати на інфраструктуру з низькими змінними витратами, які відповідають масштабам вашого бізнесу. За допомогою хмари, бізнесу більше не потрібно планувати та закуповувати сервери та іншу ІТ-інфраструктуру за тиждень чи місяці. Натомість вони можуть миттєво розгортати сотні чи тисячі серверів за лічені хвилини та та відповідно збільшувати продуктивність бізнесів.

Сьогодні AWS забезпечує високонадійну, масштабовану, недорогу інфраструктурну платформу у хмарі, яка забезпечує сотні тисяч підприємств у 190 країнах світу.

Amazon Web Services пропонує широкий набір глобальних хмарних продуктів, включаючи обчислення, зберігання, бази даних, аналітику, мережеві послуги, мобільні пристрої, інструменти для розробників, інструменти управління, IoT, безпеку та додатки для підприємств: із оплатою за запитом, посекундно, за час користування. Від зберігання даних, до інструментів розгортання, каталогів для доставки контенту, доступні понад 140 сервісів AWS. Нові послуги можна надати швидко, без попередніх капітальних витрат. Це

					<i>КНТЕУ 121 07-21.БР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка інтернет магазину з продажу комп'ютерної техніки	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					Р1	7	56
Керівник	Рзаєва С.Л.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М.О.							
Розроб.	Єрохін В.О..							
					Розділ 1			

дозволяє підприємствам, стартапам, малому та середньому бізнесу та клієнтам у державному секторі отримати доступ до будівельних блоків, які вони потребують, щоб швидко реагувати на мінливі вимоги бізнесу.

З точки зору людини яка має відношення до розробки програмного забезпечення, AWS – це прекрасний інструмент, для розгортання усієї необхідної інфраструктури яка потрібна для функціонування будь-якого додатку, оскільки ця платформа надає безліч корисних сервісів, котрі можна інтегрувати у додаток без необхідності налаштовувати їх окремо. Наприклад: майже кожен веб додаток має взаємодіяти із базою даних, у звичайному випадку, коли хостинг здійснюється у датацентрі, необхідно вручну налаштовувати все від самого початку: встановити операційну систему на обраний сервер, скачати всі необхідні програми для запуску створеного веб додатку, встановити ядро бази даних та налаштувати взаємодію між додатком та базою даних. Це мінімальний набір інструкцій які потрібно зробити для того щоб створена програма почала коректно працювати. А якщо для роботи додатку необхідна не тільки база даних, а ще й інші програми, якщо було здійснено інтеграцію із іншими компаніями які надають свої програмні продукти у використання іншим проектам, для цього потрібно буде знову пройти всі шаги із встановлення та налаштування необхідного софту, конфігурування мережі та налаштування контролю доступу.

Найбільша перевага AWS це як раз таки різноманітність послуг які можна знайти на цій платформі для використання у своєму додатку. Тепер, якщо у якості хостингу обраний Амазон, не треба думати про половину роботи яку потрібно було робити у датацентрах, відтепер для запуску свого додатку необхідно лише обрати декілька характеристик що описують віртуальну машину, на якій буде виконуватись вся робота, додати список бажаних сервісів із якими необхідна взаємодія, та запустити процес створення обраних ресурсів. Після того як усі ресурси були створені, можна додати моніторинг процесів та налаштувати

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

повідомлення про критичні сбої в системі, і відтепер, додаток самостійно функціонує у хмарі, та контролює процес роботи програми.

Висновок: AWS ідеально підходить для усіх видів програмних продуктів та будь-якої сфери діяльності, оскільки різноманітність пропонованих сервісів дозволяє гнучко налаштувати розробляємий продукт, із мінімальними зусиллями за короткий проміжок часу.

1.1.2 Azure

Azure – платформа хмарних обчислень від Microsoft. Це об'єднання служб для створення, розгортання та управління програмами, які працюють у захищених світових центрах обробки даних Microsoft. Набір сервісів може зробити практично будь-яке програмне рішення можливим - від великого роздрібного націлення на глобальну аудиторію до об'ємних процесорів, наукових обчислень даних, до простих резервних копій даних. Практично будь-яке програмне забезпечення, яке можна уявити, може бути написане, щоб скористатися послугами Azure і може працювати в Microsoft Cloud.

Microsoft Azure була запущена в 2010 році і є одним з найбільших комерційних постачальників хмарних послуг. Ця платформа пропонує такий спектр інтегрованих хмарних сервісів та функціональних можливостей як аналітика, обчислювальна техніка, мережа, база даних, сховище даних, мобільні та веб-додатки, які безперебійно інтегруються з вашим середовищем для досягнення ефективності та масштабованості.

Microsoft, з як правило, є популярним вибором у керівників рівня С, які мають давні стосунки з постачальником і знають, що вони можуть споживати велику кількість своїх корпоративних потреб у комп'ютері в одному місці, починаючи від продуктивності та корпоративного програмного забезпечення.

Оскільки Azure розпочав свою діяльність трохи пізніше ніж Amazon, платформа має не таким широким асортиментом сервісів як AWS але має свої переваги. Microsoft можна позиціонувати як номер два, після AWS, завдяки поєднанню

					КНТЕУ 121 06-15.БР	Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата		

Azure, Office 365 та Teams. Тобто перше що можна виділити при розгляданні цієї платформи – це зручна інтеграція із іншими службами від Microsoft, що також деколи казується на цінній політиці, оскільки ліцензійовані продукти від своєї ж компанії, обходяться дешевше ніж у конкурента. Також не можна не відзначити той факт, що Microsoft виробником системи Windows, одже використання цієї операційної системи теж коштує дешевше ніж в амазоні.

Висновок: Azure – більш молода платформа на відміну від амазону, та має менший асортимент сервісів які можна застосувати у процесі розробки програмного продукту, але не зважаючи на це – можна із впевненістю сказати що це фінансово більш вигідне рішення, у разі якщо використовуюється інтеграція із іншими продуктами від Microsoft.

1.2 Аналіз існуючих СУБД

1.2.1 Oracle

Не дивно, що корпорація Oracle пропонує однойменний продукт, з якого зазвичай починається розгляд варіантів популярних СУБД. Перша версія Oracle була створена в кінці 70-х років, маючи на даний момент блискучу репутацію. Крім того, існує кілька версій цього продукту для задоволення потреб конкретної організації.

Актуальна версія Oracle на даний момент – призначена для хмарних середовищ і може бути розміщена на одному або декількох серверах, це дозволяє управляти базами даних, які містять мільярди записів. Деякі з функцій новітньої версії Oracle включають в себе gridframework і використання як фізичних, так і логічних структур. Це означає, що фізичне управління даними не впливає на доступ до логічних структур. Крім того, безпека в цій версії доведена до найвищого рівня, тому що кожна транзакція ізольована від інших.

Переваги

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		10

- Найсвіжіші інновації та вражаючий функціонал вже впроваджені в цьому продукті, оскільки компанія Oracle прагне тримати планку навіть на тлі інших розробників СУБД.
- СУБД від Oracle є вкрай надійною, фактично це еталон надійності серед подібних систем.

Недоліки

- Вартість Oracle може виявитися непомірно високою, особливо для невеликих організацій.
- Система може зажадати значних ресурсів вже відразу після установки, тому можливо буде потрібно модернізувати обладнання для впровадження Oracle.

Висновок: ідеально підходить для великих організацій, які працюють з величезними базами даних і різноманітними функціями.

1.2.2 MySQL

MySQL – одна з найпопулярніших баз даних для веб-додатків. Фактично, є стандартом defacto для веб-серверів, які працюють під управлінням операційної системи Linux. MySQL – це безкоштовний пакет програм, однак нові версії виходять постійно, розширюючи функціонал і покращуючи безпеку. Існують спеціальні платні версії, призначені для комерційного використання. У безкоштовній версії найбільший наголос робиться на швидкість і надійність, а не на повноту функціоналу, який може стати і гідністю і недоліком - в залежності від області застосування.

Розробка та підтримка сайту MySQL здійснює корпорація Oracle, яка отримала права на торговельну марку разом з поглиненої SunMicrosystems, яка раніше придбала шведську компанію MySQLAB. Продукт поширюється як під GNUGeneralPublicLicense, так і під власною комерційною ліцензією. Крім цього, розробники створюють функціональність за замовленням ліцензійних

					КНТЕУ 121 06-15.БР	Аркуш
						11
Зм.	Аркуш	№ докум	Підпис	Дата		

користувачів. Саме завдяки такому замовленню майже в найраніших версіях з'явився механізм реплікації.

Ця СУБД дозволяє вибирати різні двигики для системи зберігання, які дозволяють змінювати функціонал інструменту і виконувати обробку даних, що зберігаються в різних типах таблиць. Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Більш того, СУБД MySQL поставляється із спеціальним типом таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі і GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць. Вона також має простий у використанні інтерфейс, і пакетні команди, які дозволяють зручно обробляти величезні обсяги даних. Система неймовірно надійна і не прагне підпорядкувати собі всі доступні апаратні ресурси.

Переваги:

- Розповсюджується безкоштовно.
- Прекрасно документована.
- Пропонує багато функцій, навіть у безкоштовній версії.
- Пакет MySQL включений в стандартні репозиторії найбільш поширених дистрибутивів операційної системи Linux, що дозволяє встановлювати її елементарно.
- Підтримує набір призначених для користувача інтерфейсів.
- Може працювати з іншими базами даних, включаючи DB2 і Oracle.

Недоліки:

- Доведеться витратити багато часу і зусиль, щоб змусити MySQL виконувати нескладні завдання, хоча інші системи роблять це автоматично, наприклад: створювати інкрементні резервні копії.
- Відсутня вбудована підтримка XML або OLAP.

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		12

- Для безкоштовної версії доступна тільки платна підтримка.

Висновки: ідеально підходить для: організацій, яким потрібен надійний інструмент управління базами даних, але безкоштовний.

1.2.3 Microsoft SQL Server

Ще однією з популярних СУБД є програмний продукт MicrosoftSQL-сервер. Це система управління базами даних, движок якої працює на хмарних серверах, а також локальних серверах, причому можна комбінувати типи застосовуваних серверів одночасно. Незабаром після випуску Microsoft SQL Server 2016. Microsoft адаптувала продукт для операційної системи Linux, а на Windows-платформі він працював спочатку.

Однією з унікальних особливостей версії 2016 року є temporal data support (тимчасова підтримка даних), яка дозволяє відстежувати зміни даних з плином часу. Остання версія MicrosoftSQL-сервер підтримує dynamicdatamasking (динамічну маскування даних), яка гарантує, що тільки авторизовані користувачі будуть бачити конфіденційні дані.

Переваги:

- Продукт дуже простий у використанні.
- Поточна версія працює швидко і стабільно.
- Движок надає можливість регулювати і відслідковувати рівні продуктивності, які допомагають знизити використання ресурсів.
- Ви зможете отримати доступ до візуалізації на мобільних пристроях.
- Він дуже добре взаємодіє з іншими продуктами Microsoft.

Недоліки:

- Ціна для юридичних осіб виявляється неприйнятною для більшої частини організацій.
- Навіть при ретельній налаштування продуктивності корпорація SQLServer здатний зайняти всі доступні ресурси.

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

- Повідомляється про проблеми з використанням служби інтеграції для імпорту файлів.

Висновок: ідеально підходить для: великих організацій, які вже використовують ряд продуктів Microsoft.

1.3 Огляд операційних систем для розробки веб-додатку

1.3.1 Операційна система Linux

Linux – узагальнена назва, на честь програміста Лінуса Торвальдса, який написав так зване ядро операційної системи. Різновидів операційних систем на базі Linux існує величезна кількість, подивитися цей список можна на вікіпедії.

Операційні системи на базі Linux створені програмістами для програмістів. З плюсів всіх дистрибутивів Linux варто відзначити наявність програми термінал. Ця проста на вигляд програма дозволяє повністю управляти комп'ютером і серверами. Багато операцій виробляти через термінал набагато швидше, ніж через програми з графічним інтерфейсом. Варто зауважити, що велика частина сайтів в інтернеті працює на серверах на базі Linux.

Коли потрібно програмно виконати багато команд, та налаштувати операційну систему на режим роботи серверу, Linux – дуже зручний та простий інструмент.

1.3.2 Операційна система Windows

Windows дуже популярна операційна система і для Windows є практично всі програми, працює практично з усіма пристроями і на противагу цьому мінус, що через велику популярність під неї пишуться велика кількість вірусів, бекдорів та рекламних програм.

Під Windows є всілякі редактори коду, інструменти для розробки, як і в Linux, але у них є свої особливості і мінуси. Наприклад, якщо у вас Windows Home, вам будуть потрібні додаткові дії при роботі з Docker – популярною програмою контейнеризації, або функціонал деяких інструментів для розробки не буде повноцінно працювати.

					КНТЕУ 121 06-15.БР	Аркуш
						14
Зм.	Аркуш	№ докум	Підпис	Дата		

Windows часто поставляється разом з комп'ютером, проте з часом поточна редакція Windows вам може не підійти для деяких завдань і доведеться купувати потрібну редакцію Windows.

1.4 Технічне завдання для розробки інтернет магазину з продажу комп'ютерної техніки

1. Загальні відомості: Програмний продукт інтернет магазин з продажу комп'ютерної техніки призначений для автоматизування діяльності стаціонарного магазину з продажу комп'ютерної техніки.

1.1 Найменування системи: Інтернет магазин з продажу комп'ютерної техніки.

1.1.1 Повне найменування системи: Інтернет магазин з продажу комп'ютерної техніки “Store”.

1.1.2 Скорочене найменування системи: Інтернет магазин “Store”.

1.2 Планові терміни початку та закінчення робіт: початок робіт – 01.02.2020, закінчення робіт: 01.06.2020

1.3 Потенційні користувачі системи: усі користувачі мережі інтернет.

2. Мета та призначення створення системи.

2.1 Призначення системи: Полегшення користування послугами з продажу комп'ютерної техніки.

2.2 Мета створення системи: автоматизування діяльності стаціонарного магазину з продажу комп'ютерної техніки.

3. Вимоги до системи.

3.1 Вимоги до системи в цілому: Програмний продукт має бути розроблений як веб додаток, мати зручний графічний інтерфейс користувача, відокремлений від серверної логіки, передбачити необхідні рівні безпеки та ізоляції програмного коду від втручання сторонніх осіб.

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

- 3.1.1 Вимоги до структури та функціонування системи, перелік підсистем: Програмний продукт має бути розділений на дві основні частини: графічний інтерфейс користувача та серверна частина. Серверна частина в свою чергу повинна мати архітектуру DDD(Domain Driven Design).
- 3.1.2 Вимоги до пристосовності системи до змін: система повинна бути розроблена таким чином, щоб кожна функціональна частина могла бути замінена іншою, із мінімальними працезатратами.
- 3.1.3 Вимоги до збереження працездатності системи в різних ймовірних умовах: система повинна мати захист від непередбачених ситуацій таких як: технічні несправності фізичних серверів, неполадки із мережею інтернет, тощо.
- 3.1.4 Вимоги до ергономіки та технічної естетики: веб додаток повинен мати зручний та інтуїтивно зрозумілий інтерфейс користувача.
- 3.1.5 Вимоги до інформаційної безпеки: веб додаток має бути захищений від несанкціонованого доступу до програмного коду, мати захист від SQL-ін'єкцій, та найрозповсюджених хакерських атак.
- 3.1.6 Розмежування відповідності ролей при доступі: веб додаток повинен мати публічний доступ для перегляду та придбання товарів для усіх користувачів мережі інтернет, доступ для оновлення інформації в магазині – тільки для адміністраторів.
- 3.1.7 Вимоги щодо застосування систем управління базами даних(СУБД): СУБД може бути використана від виробника будь-якої компанії.
4. Вимоги до програмного забезпечення: Програмний продукт має бути розроблений за допомогою платформи .Net та мови програмування с# для серверної частини та платформи Node.JS та мови програмування TypeScript або JavaScript для графічного інтерфейсу користувача.

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

5. Вимоги до технічного забезпечення: розроблений програмний продукт має бути доступний у мережі інтернет шляхом публікування програмного коду у хмарному середовищі.

1.5 Висновки до розділу 1

У першому розділі були розглянуті існуючі хмарні середовища, такі як Amazon Web Services та Microsoft Azure. Було визначено, що Amazon надає більш функціональну платформу за рахунок того що компанія довше знаходиться на ринку, має інтеграцію із багатьма корисними програмами та зарекомендувала себе як надійне рішення для розробки веб-додатків у хмарному середовищі.

Microsoft в свою чергу надає не менш надійне рішення для побудови програмного продукту, але як можна було очікувати, компанія орієнтується в більшості на інтеграцію із платформою Windows та сервісів які співпрацюють із цією операційною системою. З точки зору фінансового питання AWS вигідніше використовувати якщо програмний продукт будується на базі операційної системи Linux, в той час як Azure – у випадку використання Windows.

Окрім цього були проаналізовані існуючі системи управління базами даних, такі як Oracle, MySQL та MsSQL. Виявлені недоліки та переваги кожної з систем та підбиті проміжні підсумки що до їх використання.

Було оглянуто найбільш використовувані варіанти операційних систем для побудови програмних продуктів веб-напрямку та обрана операційна система Linux, оскільки саме на ній будується більшість веб-додатків та серверів, окрім цього варто зазначити, що ця платформа краще інтегрується із програмою контейнеризації – Docker, яка також є найрозповсюдженим варіантом для побудови не тільки веб-додатків, а й інших систем завдяки простоті налаштування та немалої компактності.

В якості хмарного середовища було обрано AWS, як більш функціональне рішення, а також за рахунок того що програмний продукт буде розроблятися на базі операційної системи Linux.

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

Оскільки програмний продукт буде розташований у хмарному середовищі AWS віддати перевагу якійсь СУБД з технічної точки зору немає можливості, адже всі оглянуті рішення можуть бути використані однаково ефективно, тому у якості бази даних було обрано MySQL.

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

РОЗДІЛ 2

ПОСТАНОВКА ЗАДАЧІ, ПРОЕКТУВАННЯ ТА НАЛАШТУВАННЯ РЕСУРСІВ НЕОБХІДНИХ ДЛЯ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ

2.1. Архітектура програмного забезпечення інтернет магазину з продажу комп'ютерної техніки

Для створення програмного продукту - інтернет магазину з продажу комп'ютерної техніки, необхідні такі елементи архітектури програмного забезпечення як: графічний інтерфейс користувача розроблений на базі платформи Node.JS та мови програмування TypeScript або JavaScript, серверна частина, розроблена на базі платформи .Net та мови програмування с#, база даних будь-якої компанії розробника СУБД, хмарне середовище, обране за результатами огляду доступних варіантів та технічне оточення необхідне для функціонування веб додатку у хмарному середовищі, яке являє собою мережеві налаштування, налаштування режимів доступу до програмного коду веб додатку, та публічного доступу у мережі інтернет.

2.2. Постановка задачі для розробки бази даних інтернет-магазину з продажу комп'ютерної техніки

Перед тим як побудувати базу даних необхідно проаналізувати та розробити її структуру. До бази даних інтернет-магазину з продажу комп'ютерної техніки необхідно включити такі сутності: категорії комп'ютерної техніки, виробники комп'ютерної техніки, товари, заклади, окремо треба створити сутність – зображення, оскільки один товар може мати багато зображень, коментарі – відгуки які будуть залишати покупці та самі покупці. Ці дані допоможуть зробити модель бази даних максимально інформативною гнучкою та структурованою.

					<i>КНТЕУ 121 07-21.БР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка інтернет магазину з продажу комп'ютерної техніки	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					Р2	19	56
Керівник	Рзаєва С.Л.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М.О.							
Розроб.	Єрохін В.О.							
					Розділ 2			

2.2.1. Проектування логічної моделі бази даних інтернет-магазину з продажу комп'ютерної техніки

Перед тим, як розробляти логічну модель бази даних треба спроектувати таблиці, поля таблиць, та інформація, яка буде міститись в базі.

По-перше і найголовніше – це таблиця товарів, у якій будуть міститися всі інформація про преїскурант продуктів інтернет-магазину, та матиме назву *Products*. Ця таблиця має містити наступні поля: назва товару, опис товару, оцінка від одного до п'яти яку може залишити покупець цього товару (важливо зазначити що це поле повинно бути опціональним, оскільки не кожен покупець бажає залишати оцінку придбаному товару) ціна товару, доступність товару – вказівка на те, наявний даний товар на складі чи ні, категорія товару – власне до якої категорії відноситься даний товар: периферія, комплектуючі до комп'ютера, чи то лептоп, виробник товару, та коментарі – які теж мають бути опціональним полем по тій самій причині що і оцінка товару,

Необхідно зазначити що таблиця має містити зовнішні ключі *CategoryId* та *ManufacturerId*, які будуть зв'язувати продукт із сутностями «Категорія товару» та «Виробник товару» відповідно.

По-друге, база даних має містити вищезгадані таблиці категорії товару та виробника товару які матимуть назву *Categories* та *Manufacturers* відповідно. Структура цих таблиць повністю однакова, та буде виглядати наступним чином: первинний ключ та назва категорії чи виробника. На перший погляд може показатися, що немає сенсу створювати такі неінформативні таблиці, але вони мають більш стратегічне значення аніж інформативне. За допомогою них буде досягнута необхідна гнучкість та розширюваність бази даних.

По третє, таблиця заказів, яку можна назвати *Orders* буде представляти собою заказ, який оформлює покупець. Ця таблиця має містити наступні поля: первинний ключ, статус заказу: у черзі на обробку; вже оформляється; доставлений; та містити поле дата замовлення.

					КНТЕУ 121 06-15.БР	Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата		

На останнє, в базу даних треба додати таблиці коментарі та картинки, де коментарі будуть мати такі поля як: первинний ключ, текст коментаря, оцінка від одного до п'яти (важливо не спутати ту оцінку яка знаходиться у таблиці товарів, оскільки та оцінка буде середнім арифметичним з усіх оцінок в коментарях), кількість лайків коментаря – для того щоб можна було зробити сортування коментарів за корисністю, поле *ProductId* – зовнішній ключ який дозволяє віднести певний коментар до конкретного продукту, та поле батьківський коментар – адже можна завжди залишити коментар на коментар.

Таблиця картинок міститиме декілька полів, таких як: ключ картинки та *ProductId* – зовнішній ключ, який виконуватиме ту саму роль що і відповідне поле у коментарях. Чому саме ключ картинки, а не сама картинка, буде розглянуто у третьому розділі при реалізації програмного продукту.

Підсумовуючи все вищесказане ми отримуємо не тільки чітко структуровану та легко розширювану базу даних але ще й нормалізовану за третьою нормальною формою.

На рис. 2.1. подано логічну модель бази даних інтрент магазину з продажу комп'ютерної техніки.

					КНТЕУ 121 06-15.БР	Аркуш
						21
Зм.	Аркуш	№ докум	Підпис	Дата		

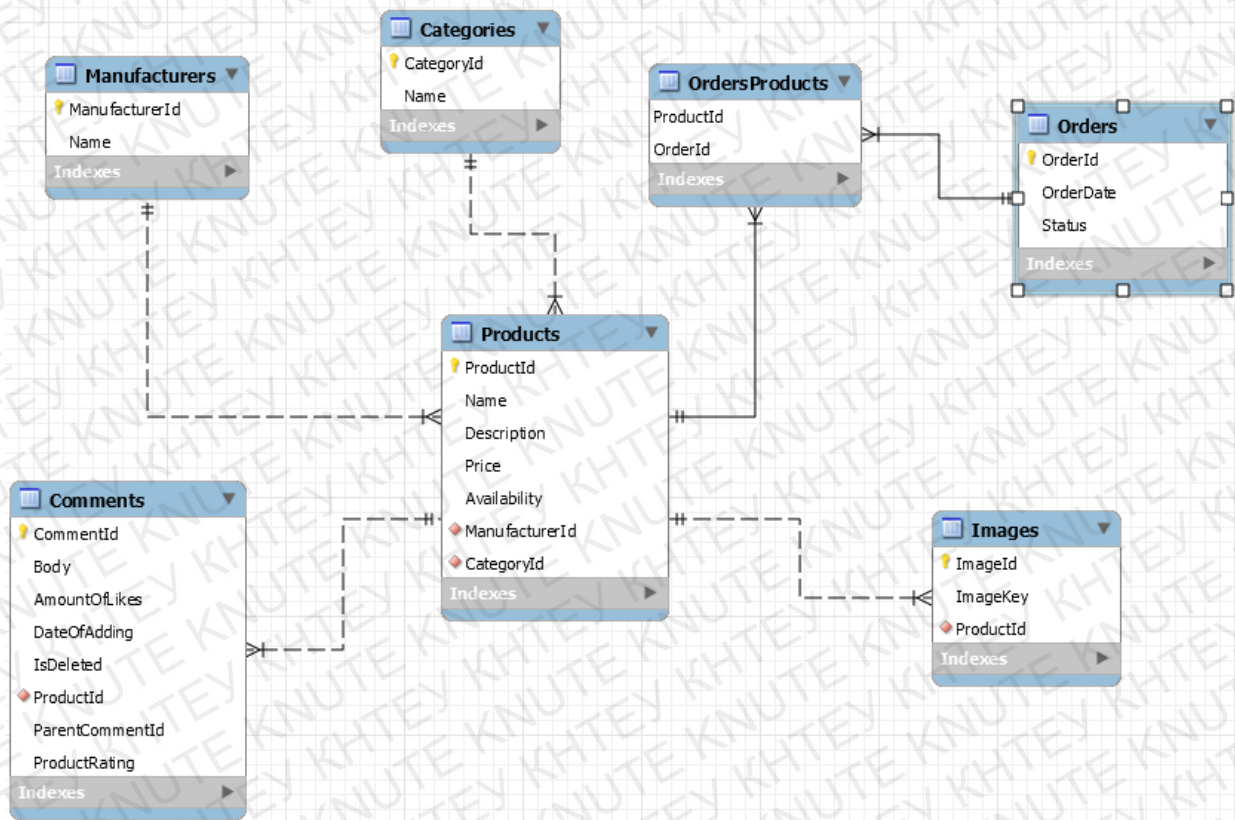


Рис. 2.1. Логічна модель бази даних інтернет магазину з продажу комп'ютерної техніки

2.2.2. Фізична модель бази даних інтернет-магазину з продажу комп'ютерної техніки

Перед розробкою фізичної моделі треба проаналізувати який тип даних будуть мати комірки у таблицях та яким чином буде здійснюватися зв'язок між таблицями. Отож всі поля, які пов'язані із цифровими значеннями, будуть мати тип даних *int*, в основному це поля пов'язані із первинними або зовнішніми ключами, окрім поля *Price* в таблиці продуктів, оскільки ціна не завжди має бути цілим числом, то для цього поля буде використано тип даних *double*. Окрім цього можна виділити такий тип даних як *DateTime*, який використовується для поля дата заказу таблиці заказів і уявляє собою календарний формат часу. Всі інші поля, які якимось чином пов'язані із зберіганням текстових даних матимуть формат даних *nvarchar(max)*, де *nvarchar* – означає текстовий тип даних який дозволяє писати,

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		22

як українською так і англійською мовами, а параметр у дужках – це максимальна кількість символів яку можна записати в дане поле, в даному випадку залишмо це значення за замовченням в сорок п'ять символів. Як було зазначено вище, деякі поля можуть бути не обов'язковими, а деякі не можна залишати не заповненими, оскільки такий підхід може зламати всю систему взаємодії таблиць одна із одною. Таким чином, до тих полів, які повинні бути обов'язково заповнені, треба застосувати обмеження поля *not null*.

Якщо проходитися детально по кожному зв'язку, то вони мають наступний вигляд: таблиця *Manufacturers* зв'язана з таблицею *Products* за ключовим полем *ManufacturerId* та має зв'язок один до багатьох, оскільки один виробник може мати багато товарів які він виготовляє. Таблиця *Categories* зв'язана з таблицею *Products* за ключовим полем *CategoryId* та має зв'язок один до багатьох, оскільки в одній категорії може бути безліч товарів. Таблиці *Comments* та *Images* зв'язані з таблицею *Products* за ключовим полем *ProductId*, та мають зв'язок один до багатьох, оскільки в одного продукту може бути багато коментарів, а також в одного продукту може бути багато малюнків. Окремо варто виділити таблицю *Orders*, яка зв'язана із таблицею *Products* через проміжну таблицю *OrdersProduct*, такий підхід обумовлено тим, що ці таблиці мають зв'язок багато до багатьох, оскільки в одного замовлення може бути безліч товарів і, в той самий час, один, і той самий товар, може належати до багатьох замовлень. Проміжна таблиця має складний первинний ключ створений за допомогою полів *ProductId* та *OrderId*, в свою чергу ця проміжна таблиця зв'язана з таблицею *Products* за ключовим полем *ProductId* і має зв'язок один до багатьох, а також зв'язана із таблицею *Orders* за ключовим полем *OrderId*. Саме таким чином організовується зв'язок між таблицями багато до багатьох.

Отже за попередньо описаною концепцією створення фізичної моделі бази даних інтернет магазину з продажу комп'ютерної техніки можна побачити наступну схему (рис. 2.2).

					КНТЕУ 121 06-15.БР	Аркуш
						23
Зм.	Аркуш	№ докум	Підпис	Дата		

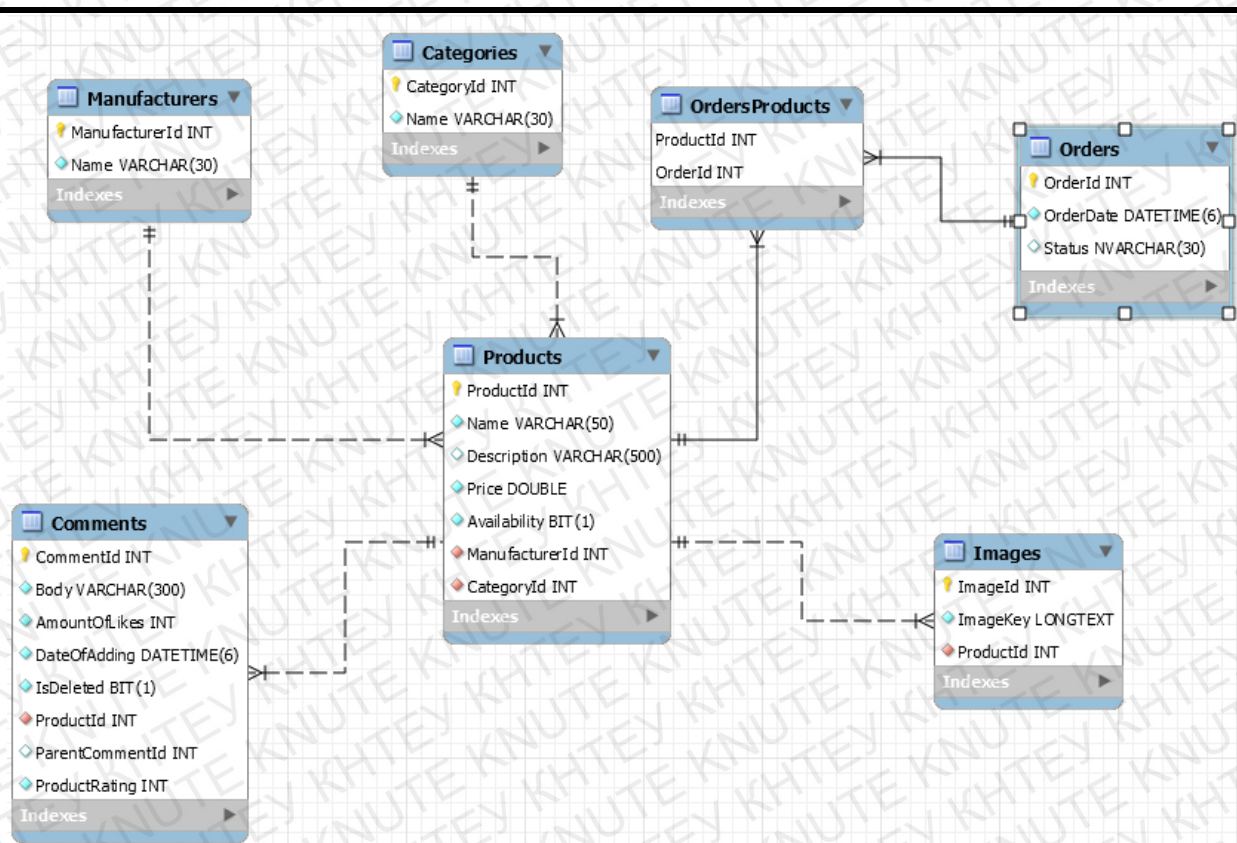


Рис. 2.2. Фізична модель бази даних інтернет-магазину з продажу комп'ютерної техніки

2.3. Формування технічного оточення інтернет-магазину з продажу комп'ютерної техніки

Програмний продукт інтернет-магазину з продажу комп'ютерної техніки повинен складатися з двох головних частин: по-перше, це бек-енд частина яка являє собою основну логіку додатку, в ній відбуваються такі дії, як звернення додатку до бази даних, виконання необхідних операцій з даними у відгук на дії користувача, виконання валідації даних, організація інформаційної безпеки програмного продукту та своєчасне інформування розробників або власника продукту про непередбачувані збої в роботі програми. По-друге, це фронт-енд частина, яка являє собою графічний інтерфейс користувача, що полегшує користування додатком. За своєю суттю графічний інтерфейс лише передає дані від користувача до сервера, та не виконує ніяких логічних обчислень або інших

маніпуляцій, такий підхід використовується для мінімізування можливості використання продукту вірусними програмами, запобігання неправомірного використання іншими користувачами, та, зокрема, для розширюваності й легкості модифікації. Так, наприклад, якщо необхідно зробити якісь правки у графічному інтерфейсі, серверна частина ніяк не чіпається, що гарантує стабільну роботу та попередження можливих збоїв у роботі. Так само, якщо необхідно змінити логіку сервера, графічний інтерфейс залишиться гарантовано у не змінному робочому стані. Не можна не відмітити те, що таким чином полегшується робота для розробника даного програмного продукту.

Для розробки серверної частини інтернет-магазину з продажу комп'ютерної техніки було обрано платформу *.Net* та мову програмування *c-sharp*. Версія платформи на якій повинна базуватися серверна частина – *.NetCore 2.2* та необхідна для цього технологія – *RESTWebApi*. Ця технологія являє собою програмну бібліотеку у якій наявний перелік доступних для виклику функцій, кожна функція має свою унікальну адресу за якою до неї можна звернутися.

Графічний інтерфейс повинен бути розроблений на платформі *Node.js* за допомогою фреймворку *Angular 8*. Така зв'язка є розповсюдженим рішенням для розробки веб-додатків, оскільки *Angular* працює із мовою програмування *JavaScript*, яка в свою чергу прекрасно взаємодіє із форматом даних *JSON*. Цей формат використовує *REST WebApi* на базі *.Net*.

2.4. Організація інфраструктури необхідної для функціонування програмного продукту в хмарному середовищі Amazon

Для побудови програмного продукту потрібно дві різні платформи: *.Net* для серверної частини та *Node.js* для графічного інтерфейсу. Встановлювати кожен операційну систему на віртуальну машину у хмарному середовищі не має жодного сенсу, оскільки це не оправдані ресурсо- та праце-затрати. У такому випадку доведеться встановлювати всі необхідні програми та налагоджувати комунікацію між всією інфраструктурою вручну, що займе чимало часу та сил,

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		25

також у випадку будь-якої непередбачуваної помилки, або у необхідності оновлення системи, доведеться кожен раз все виправляти заново. Спеціально для таких випадків, коли мова йде про хмарне середовище, а особливо про веб-розробку, використовуються технології контейнеризації. Контейнеризація – можливість розгорнути програмний продукт на будь якій платформі за лічені хвилини. Такий результат досягається завдяки особливій технології: розробник створює конфігураційний файл, у якому вказує необхідну операційну систему, перелік команд для налагодження інфраструктури та запуску програми. Сам контейнер являє собою звичайну операційну систему, але у мінімальній комплектації, до неї входять лише ті компоненти, які необхідні для запуску та функціонування програмного продукту. Все це робить контейнери дуже малими за обсягом та дозволяє швидко встановлювати, запускати необхідне програмне забезпечення.

Одна із популярних технологій контейнеризації – *Docker*. Для завантаження *Docker-контейнера*, необхідно *Docker-ядро* – спеціальну програму для запуску та контролю контейнерів.

2.4.1 Створення віртуальної машини засобами AWS Services

Хостинг у хмарному середовищі *Amazon* виконується за допомогою сервісу *EC2 – ElasticCloudCompute*. Цей сервіс надає віртуальній машини із різними характеристиками, встановлювати необхідне програмне забезпечення. Від початку сервіс вже має перелік популярних імеджів – заздалегідь заготовлених операційних систем, оскільки як було зазначено раніше у якості операційної системи було обрано *Linux*, для роботи програмного продукту буде досить віртуальної машини у мінімальній комплектації із передінстальованою операційною системою *Linux*, розрядність системи повинна бути шістдесят чотири. Так як встановити *Docker-ядро* на операційній системі *Linux* можна однією командою, то за для цих потреб можна використати звичайний імедж *Linux* системи у мінімальній комплектації, а саме *t2.micro*, який має одне ядро та

					КНТЕУ 121 06-15.БР	Аркуш
						26
Зм.	Аркуш	№ докум	Підпис	Дата		

один гігабайт оперативної пам'яті, постійної пам'яті буде досить восьми гігабайт, що є значенням за замовченням.

Окрім вибору технічних характеристик віртуальної машини ще є конфігурування мережових налаштувань та безпеки, які є найважливішими параметрами, оскільки забезпечують необхідний рівень захисту та ізоляції програмного продукту від постороннього втручання.

Для того щоб мати можливість розгорнути будь-які ресурси у середовищі *Amazon*, необхідно для початку створити віртуальне приватне середовище – *VPC*. *VPC* являє собою діапазон *Ip*-адрес, які можна використовувати у межах свого проекту, таким чином кожен користувач має свій власний діапазон *Ip*-адрес, які не перетинаються одна з одною, що забезпечує безпроблемне функціонування безлічі ресурсів. Після створення віртуального приватного середовища, треба визначитись чи потрібне авто-призначення *Ip*-адресу для даного імеджа, чи треба створити власну адресу та обрати її в налаштуваннях.

Для того щоб віртуальна машина мала доступ до якихось певних дій та операцій у інших сервісах платформи *Amazon*, необхідно створити роль із певними повноваженнями, призначити цю роль на створювану віртуальну машину. Взагалі, будь-який ресурс який створюється у межах *Amazon*, повинен мати свою роль яка надає можливості конкретному сервісу зберігати належний рівень безпеки. У такому разі неможливо раптово та необережно зайти туди куди не треба та зламати весь функціонал, оскільки для цього банально немає повноважень. Отож для віртуальної машини потрібна роль, яка матиме доступ до таких же віртуальних машин, та до бази даних, оскільки база даних теж створюється засобами хмарного середовища.

Наостанок, перед створенням віртуальної машини необхідно обрати групу безпеки – *SecurityGroup*, яка регулює доступ до створюваних ресурсів із конкретних *Ip*-адрес та через певні типи з'єднання: чи то *TCP*, *UDP*, *SSH* тощо. За допомогою цих груп можна обмежити доступ наприклад для всіх клієнтів які

					КНТЕУ 121 06-15.БР	Аркуш
						27
Зм.	Аркуш	№ докум	Підпис	Дата		

мають *Ip-адресу* із діапазону 187.187.0.6/16 та намагаються підключитися через Інтернет по TCP-протоколу.

Створення віртуальної машини, як для серверної частини так і для графічного інтерфейсу, ніяк не відрізняється, оскільки це лише конфігурування необхідного технічного оточення для коректної роботи програмного продукту. Різниця у налаштуваннях буде описана далі, при описі створення конфігураційного файлу для побудови контейнера.

2.4.2 Створення бази даних засобами AWS Services

Для створення бази даних засобами хмарного середовища *Amazon* необхідно скористуватися сервісом *RDS – Relational Database Service*. Цей сервіс має інтеграцію із усіма популярними базами даних: *MySQL*, *MSSQL*, *MariaDB*, *PostgreSQL*, *Oracle* та *Amazon Aurora* – це поліпшена версія SQL бази даних за рахунок деякої оптимізації, яка може працювати у режимі сумісності із *MySQL* та *PostgreSQL*, тобто запити у цій СУБД будуть виконуватися трішки швидше, ніж в оригінальних СУБД. Але ця продуктивність не дуже важна для програмного продукту: інтернет-магазин з продажу комп'ютерної техніки, оскільки немає цінності у збереженні декількох мілісекунд на виконання запиту.

Отже для функціонування бази даних необхідно вказати версію яка буде використовуватися, в даному випадку це повинна бути остання стабільна версія ядра СУБД – *MySQL 5.7.22*. Далі необхідно вказати назву майбутньої бази даних – *ComputerStoreDb*, також для створення бази даних необхідно створити користувача: придумати ім'я та пароль, які будуть використовуватися для авторизації. Наостанок залишається обрати приватне віртуальне середовище у якому буде розміщено базу даних та кількість файлового сховища, для даного програмного продукту вистачить стандартний розмір у сто гігабайт.

					КНТЕУ 121 06-15.БР	Аркуш
						28
Зм.	Аркуш	№ докум	Підпис	Дата		

2.5 Висновки до розділу 2

У другому розділі була сформована задача розробки бази даних, створено логічну та фізичну моделі бази даних, проаналізовано всі зв'язки між таблицями та визначено типи даних створених полів.

Окрім цього, було обґрунтовано вибір програмних платформ, на яких буде функціонувати інтернет-магазин з продажу комп'ютерної техніки, як для серверної частини, так і для графічного інтерфейсу, обрано технологію контейнеризації для розгортання додатку в хмарному середовищі, у протилежність до ручного конфігурування.

Для створення ресурсів у хмарному середовищі *Amazon* було сформовано перелік дій: створення технічного оточення хмарного середовища, а саме ролі для контролю доступу до ресурсів, віртуальне приватне середовище у якому буде функціонувати програмний додаток та групи безпеки, які контролюють доступ із Інтернету до працюючого у хмарі програмного продукту. Визначено порядок створення віртуальних машин, на яких будуть розгорнуті контейнери із програмним кодом, та необхідні кроки щодо створення бази даних засобами *AWS Services*.

					КНТЕУ 121 06-15.БР	Аркуш
						29
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ

3.1. Проектування та реалізація графічного інтерфейсу інтернет магазину з продажу комп'ютерної техніки

Перед тим як розпочати розробку серверної частини програмного продукту, необхідно спроектувати графічний інтерфейс користувача, а вже тільки після цього відштовхуватися далі, оскільки повну картину необхідного функціоналу можна побачити тільки тоді, коли вже визначені всі потреби користувача через розроблений макет веб сторінки.

Головна сторінка являє собою «обличчя» програмного продукту, отож має бути простою та лаконічною. У верхній частині повинен знаходитися «header» - область інтерфейсу, яка буде наявною на всіх доступних сторінках програмного продукту, оскільки містить у собі стратегічно важливі елементи: телефон центрального офісу інтернет магазину, контакти – посилання на географічне положення, карту та телефони всіх доступних стаціонарних магазинів по всій країні, допомога – посилання на сторінку зв'язку із технічною допомогою, перемикач мови відображення контенту на сайті Інтернет магазину з продажу комп'ютерної техніки, поточне обране користувачем місто, яке згодом використовується при формуванні замовлення, посилання на особистий кабінет у якому можна переглядати минулі замовлення та статус поточних заказів, історію коментарів, тощо, додатково верхня частина інтерфейсу має містити у собі графу пошуку бажаного продукту, кнопку меню – яку можна визвати перебуваючи на будь-якій сторінці, а також кнопку корзина – у якій можна знайти перелік всіх замовлених продуктів у даний момент, але ще не підтверджений остаточно. Із не

					<i>КНТЕУ 121 07-21.БР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка інтернет магазину з продажу комп'ютерної техніки	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					РЗ	30	56
Керівник	Рзаєва С.Л.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М.О.							
Розроб.	Єрохін В.О..							
					Розділ 3			

функціонального контенту у верхній частині можна виначити назву магазину – «Store» та логотип.

Таким чином, підсумовуючи все вищезгадане, «header» програмного продукту інтернет магазину з продажу комп'ютерної техніки повинен мати вигляд, як на рис. 3.1.



Рис. 3.1. «header» – веб сайту з продажу копажу комп'ютерної техніки.

Головна сторінка сайту матиме наступний вигляд: зліва повинне знаходитися меню усіх наявних категорій товарів які продаються в інтернет магазині, таким чином досягається найкращий користувацький досвід, оскільки відвідувач сайту одразу бачить весь перелік доступних категорій та може без зайвого пошуку обрати те що йому потрібно, надалі всю центральну частину займатиме великий слайдер у якому будуть зазначені найгарячіші пропозиції інтернет магазину, тобто товари із найбільшими знижками які можна наразі придбати. Слайдер та ліве меню категорій повинні знаходитися на одному рівні, якби представляти частину одного рядка, розділеного на дві частини.



Рис. 3.2. Головна сторінка сайту із лівим меню-категорій та слайдером у центрі.

					КНТЕУ 121 06-15.БР		Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			31

Такий підхід концентрує увагу відвідувача одразу на двох важливих речах: що він може придбати на цьому сайті та на які товари можна отримати немало знижку, це такий собі маркетинговий хід із приваблювання клієнта залишитись на сайті більше ніж він хотів, та переглянути інші товари які йому не були потрібні, але зі знижкою він не проти їх придбати.

Під першим рядком головної сторінки, який складається з слайдера та меню категорій товарів, знаходиться другий рядок, теж розділений на дві частини, лівий контент якого складається з короткого опису компанії-власника інтернет магазину, це потрібно для того щоб нові користувачі відчули довіру до нового інтернет магазину який раніше ніхто не бачив, вміст цього опису повинен бути коротким та змістовним. Друга частина нижнього рядка – це перелік компаній партнерів, що вже працюють із інтернет магазином. Цей блок також має стратегічне значення, оскільки демонструє ще не дуже довірливому відвідувачу що саме із цим магазином вже працюють досить великі компанії, внаслідок чого рівень зацікавленості та довіри зростає у декілька разів. Перелік компаній партнерів краще всього виконувати у вигляді блока логотипів компаній, це робить дизайн одночасно красивішим інформативнішим та лаконічнішим, оскільки логотипи великих компаній знає кожен другий громадянин.



Рис. 3.3. Нижня частина головної сторінки із інформацією про компанію та перелік компаній партнерів.

Наостанок в нижній частині сайту треба закріпити «footer» - так називають нижню частину веб сторінки яка зазвичай не несе в собі корисної інформації, а виконує якусь формальну роль. Зазвичай у футері роблять або дерево сайту, або

					KHTEY 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		32

записують якусь не значну інформацію, або просто залишають відмітку про належність права власності даного програмного продукту. У нашому випадку буде використано саме останній варіант: у футері буде розміщено повідомлення про те що право власності на інтернет магазин з продажу комп'ютерної техніки належить до компанії Store. Реалізація цього підходу виглядатиме як вузька полоса закріплена внизу веб сторінки на якій розташовано текстове повідомлення.

О нас

Store - молодая платформа, специфицирующая на продаже компьютерной техники и периферии. Мы работаем с крупнейшими поставщиками компьютерной техники и электроники по всему миру. Мы продаем не только отдельное железо или готовые решения, мы работаем с клиентом в индивидуальном порядке: возможно персональная консультация и подбор уникальных сборок специально подготовленных под ваши потребности. Вся наша продукция имеет гарантийный срок, а так же наши специалисты всегда готовы принять ваш товар на диагностику для дальнейшего его восстановления или замены.

Наши партнеры



Copyrights 2020 by Store Company

Рис. 3.4. Нижня частина веб додатку із повідомленням про належність права власності до компанії Store

Після того як користувач попадає на головну сторінку у нього є один варіант розвитку подій – це зробити те, за чим він відвідав інтернет магазин: обрати бажану категорію товарів, для пошуку необхідної техніки, або напряму увести назву шуканого товару у строку пошуку. Для початку розглянемо перший варіант.

При натисканні на певну категорію товарів, користувач переходить на сторінку товарів, наявних у цій категорії. Сторінка товарів має наступний вигляд: з лівої частини розташовано меню доступних фільтрів, які можна використовувати для сортування існуючих товарів. Перелік фільтрів має у собі такі опції: сортування за фірмою-виробником, по заданому ціновому діапазону, та можливість упорядкування за популярністю, ціною, та рейтингом.

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		33

Центральна частина сторінки товарів заповнена відповідно самими продуктами, які наявні в інтернет магазині. Кожен товар має окремий блок, виділений рамкою, у якому міститься наступна інформація про товар: головна картинка товару, фірма-виробник, назва товару, рейтинг цього товару від одного до п'яти, кількість відгуків – які оформлені як посилання, для можливості одразу перейти до їх перегляду, ціна товару та наявність на складі, яка виражається написом: є в наявності або немає в наявності. Такий мінімалістичний дизайн так званих товарних карток обран не випадково, у цьому випадку користувач бачить лише те що найбільше турбує кожного покупця: скільки коштує товар, що про нього думають інші власники та чи можна його придбати саме зараз. Вся додаткова інформація типу детального опису та коментарі – не є першочерговою, одже її можна буде знайти тільки якщо саме цей товар зацікавив користувача.

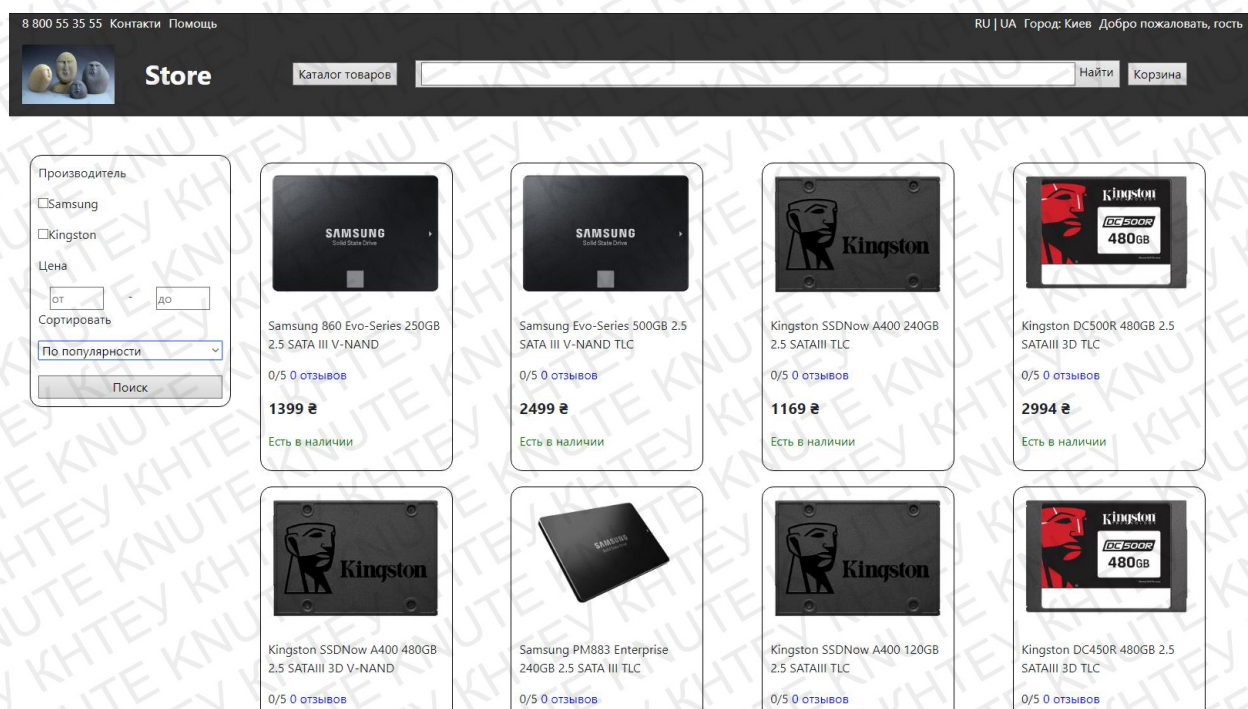


Рис. 3.5. Сторінка товарів наявних у категорії SSD.

Одразу після відвідування сторінки товарів, можна переглянути лише шістдесят товарів (п'ятнадцять рядків по чотири товара в рядку), що є досить достатньою кількістю для середньостатистичного користувача, якщо ж категорія

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		34

має більше товарів ніж можна відобразити за один запит, використовується пагінація.

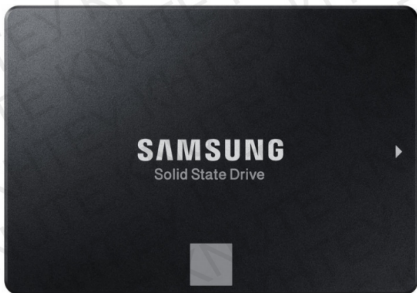
В результаті сторінка товарів має наступний вигляд (Рис. 3.4)

Обравши бажаний товар користувачу потрібно лише натиснути на відповідну картку товару, після чого він буде переадресований на сторінку конкретного товару.

Сторінка товару має наступні елементи: вгорі шістнадцятим жирним шрифтом написані виробник та назва товару, під назвою знаходиться рейтинг цього товару від нуля до п'яти, нижче, більшу частину лівої половини екрану займає зображення бажаного товару, а справа від цього зображення надрукована деяка інформація що до придбання товару, а саме: наявність товару на складі, ціна товару, кнопка для придбання товару, ціна розрахована для придбання товару на виплату у період на вісімнадцять місяців, та табличка доступних варіантів для отримання товару: самовивіз зі складу, самовивіз із нової пошти, або доставка кур'єром, та розцінки на вказані послуги. Вищезгадані елементи знаходяться якби в одному рядку, та компактно розташовані один біля одного.

Samsung 860 Evo-Series 250GB 2.5 SATA III V-NAND

0/5



Есть в наличии

Цена 1399 ₴

Купить

от 87 ₴ / месяц при покупке товара в кредит или оплате частями

Самовывоз из наших магазинов	Забрать сегодня до 19:00	Бесплатно
Самовывоз из Новой Почты	Отправим 19.05.2020	Бесплатно
Курьер по вашему адресу	Доставка завтра с 10:00	60 ₴

Характеристики

Объем
250 Гб
Скорость чтения
до 550 МБ/сек

Комментарии

Рис 3.6. Сторінка обраного товару.

					КНТЕУ 121 06-15.БР		Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			35

Наступний рядок являє собою перемикач між двома вкладками: детальні характеристики товару та коментарі. При натисканні на першу кнопку з'являється звичайний перелік характеристик вказаний фірмою-виробником. При виборі вкладки коментарів відповідно з'являється область із відображеними існуючими коментарями, та можливістю залишити свій. Блок коментаря складається з рейтингу товару, який залишає користувач якщо бажає, оскільки це опціональна функція, ім'я користувача який залишив даний коментар, дата створення коментаря, можливість оцінити коментар за допомогою відмітки «Мені подобається», та кнопка для того щоб залишити відповідь на коментар. Таким чином приклад графічного інтерфейсу для вищеописаних вкладок можна знайти на малюнках 3.7 та 3.8.

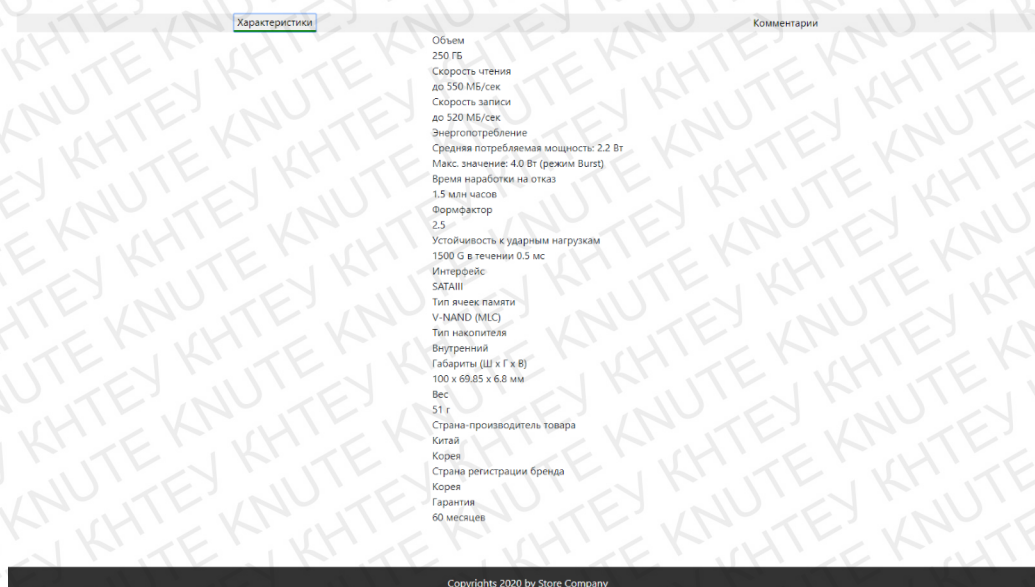


Рис. 3.7 Вкладка «Характеристики» на сторінці обраного товару.

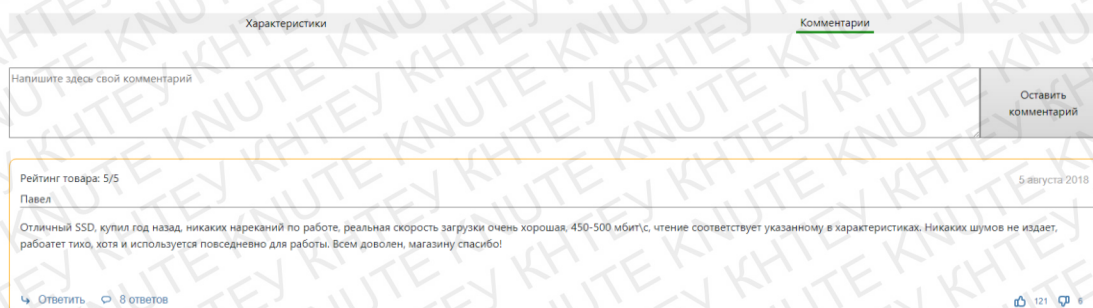


Рис 3.8. Вкладка «Коментарі» на сторінці обраного товару.

									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата				КНТЕУ 121 06-15.БР	36

Відтепер можна повернутися до другого варіанту попадання на сторінку товару – через строку пошуку. Як вже зазначалося строка пошуку – це ключовий елемент інтернет магазину, отож він наявний із будь-якої сторінки. Для того щоб знайти необхідний товар, користувачу необхідно лише ввести назву у поле. Після того як назва товару була введена, виконується запит до серверу, на якому за допомогою пошукової системи Elasticsearch виконується пошук товару. Elasticsearch - тиражуєма вільна програмна пошукова система, за станом на середину 2010-х років - найпопулярніша в своїй категорії. Оскільки сервіси амазону як вже зазначалося мають велику кількість доступних для інтеграції систем, elasticsearch звісно є у цьому переліку.



Рис. 3.9. Рядок пошуку із запропонованими результатами запиту.

Як можна побачити на рис. 3.9, після того як користувач ввів бажану назву товару, в результатах пошуку будуть запропоновані знайдені варіанти, натиснувши на які відбудеться переадресація на сторінку відповідного товару яка була описана вище. Якщо ж користувач відводить фокус від строки пошуку клацнувши мишкою у вільному місці, спливаюче поле із результатами зникає, повертаючи початковий стан елемента.

Таким чином залишається ще одна можливість яка відкривається користувачу зі сторінки товару – це кошик покупця. Після того як було обрано бажаний товар, та натиснути на кнопку придбання товару, у верхній частині сторінки, на кнопці «Корзина» з'являється кількість товарів, які покупець додав у кошик. Натиснувши на цей кошик, користувач потрапляє на сторінку оформлення заказів, де можна знайти перелік всіх товарів які були додані у кошик, кожен

					КНТЕУ 121 06-15.БР	Аркуш
						37
Зм.	Аркуш	№ докум	Підпис	Дата		

товар має свою картку як на сторінці категорії товарів, але трішки з зменшеною інформативністю: у картці наявне зображення товару, назва, виробник та ціна товару, окрім картки товару в кінці переліку можна знайти підсумкову суму, на яку було куплено товарів.

Корзина



SSD Samsung 860 Evo-Series 250GB 2.5 SATA III V-NAND
1399 грн

Дополнительные услуги

☐ +1 или +2 года гарантии

Продление гарантии на 2 года (1001-2000) 179 грн

— 1 + ✕



SSD Samsung Evo-Series 500GB 2.5 SATA III V-NAND TLC
2499 грн

Дополнительные услуги

☐ +1 или +2 года гарантии

Продление гарантии на 1 год (1001-2000) 139 грн

— 1 + ✕



SSD Kingston SSDNow A400 240GB 2.5 SATAIII TLC
1169 грн

Дополнительные услуги

☐ +1 или +2 года гарантии

Продление гарантии на 1 год (1001-2000) 139 грн

— 1 + ✕

Итого: 5067

Оформить заказ

Рис. 3.10. Сторінка «Кошик покупця»

Отже, після того як було сформовано всі вимоги до інтерфейсу інтрнет магазину з продажу комп'ютерної техніки були сформовані, макети намальовані. Можна перейти до серверної частини.

3.2. Проектування та реалізація серверної частини інтернет магазину з продажу комп'ютерної техніки

Для створення серверної частини на базі платформи .Net Core буде використовуватись DDD(Domain Driven Design) – це такий підхід коли вся логіка програмного продукту будується навколо предметної області. Відповідно, по-перше, необхідно визначити цю предметну область. Насправді вона вже визначена на моменті проектування бази даних, отже можна ще раз перелічити необхідні сутності: товар, виробник, категорія товару, заказ, коментар та малюнок

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		38

товару. Відтепер навколо кожної сутності необхідно будувати архітектуру програмного продукту.

Як було зазначено раніше, архітектура буде створена на базі REST API, отже розпочати можна зі створення контролерів зі стандартними CRUD(create, read, update, delete) операціями для кожної сутності. Далі перейти до створення відповідних сервісів, а потім відповідних репозиторіїв.

```
12 namespace WebApi.Controllers
13 {
14     [Route(template: "api/product")]
15     [ApiController]
16     1 reference
17     public class ProductController : ControllerBase
18     {
19         private readonly IProductService _productService;
20         private readonly ICommentService _commentService;
21
22         0 references | 0 exceptions
23         public ProductController(IProductService productService, ICommentService commentService)
24         {
25             _commentService = commentService;
26             _productService = productService;
27         }
28
29         [HttpGet]
30         [AllowAnonymous]
31         0 references | 0 requests | 0 exceptions
32         public async Task<ActionResult<List<ProductResponse>>> Get()
33         {
34             var gamesList = await _productService.GetAll();
35             return Ok(gamesList.Select(p => p.ToProductResponse()));
36         }
37
38         [HttpGet]
39         [Route(template: "{productId:int:min(1)}")]
40         [AllowAnonymous]
41         0 references | 0 requests | 0 exceptions
42         public async Task<ActionResult<ProductResponse>> GetGame(int productId)
43         {
44             var game = await _productService.GetById(productId);
45             return Ok(game.ToProductResponse());
46         }
47     }
48 }
```

Рис 3.11. Контролер для виклику функцій пов'язаних із сутністю «продукт».

Одразу необхідно зазначити що все згадані архітектурні рівні потрібно відокремлювати один від одного, розбиваючи програмний продукт на окремі прошарки. Такого підходу необхідно дотримуватись одразу за декількох причин: полегшення тестування програмного продукту, оскільки можна протестувати як окремо взятий компонент, так і всю систему загалом, по-друге це дотримання архітектурного шаблону, який дозволяє зробити програмний продукт

					КНТЕУ 121 06-15.БР	Аркуш
						39
Зм.	Аркуш	№ докум	Підпис	Дата		

слабозв'язним, що в свою чергу дозволяє відкласти прийняття конкретних рішень, та якнайдовше використовувати абстракції, і наостаннє це завищує масштабованість програмного продукту, що є одним із найважливіших показників при створенні веб-додатків.

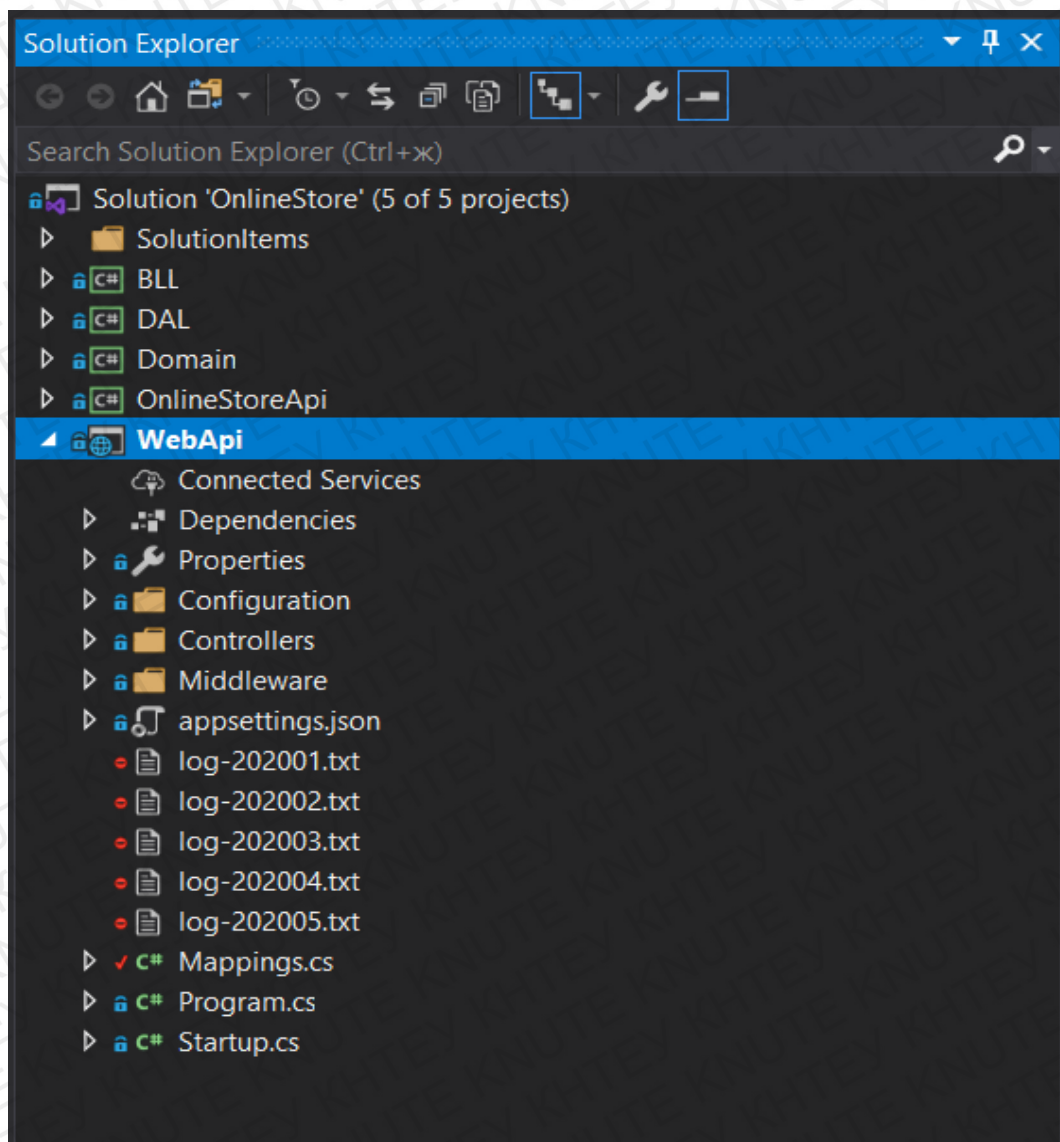


Рис 3.12. Структура проекту серверної частини інтернет магазину з продажу комп'ютерної техніки.

Таким чином на рис. 3.12. можна побачити структуру програмного коду серверної частини з дотриманням вищезгаданих рекомендацій що до масштабованості та слабозв'язності. Отже у такому випадку у проекті відокремлені такі елементи: WebApi – який відповідає за отримання та обробку

					КНТЕУ 121 06-15.БР	Аркуш
						40
Зм.	Аркуш	№ докум	Підпис	Дата		

запитів від графічного інтерфейсу, BLL – логічна частина, яка виконую всю логіку запиту відповідно, DAL – компонент зв'язку логіки програмного продукту із базою даних, звідси спочатку формуються всі запити до бази даних а потім відповідно виконуються, Domain – компонент сутностей навколо яких відбувається побудова усієї логіки програмного додатку, у ньому знаходяться сутності, перелічені раніше, які являють собою звичайні класи із переліком властивостей притаманних даній сутності, та OnlineStoreApi – так званий API інтерфейс програмного продукту, який має більш стратегічне значення ніж практичне, оскільки розроблений для майбутнього інтегрування із іншими додатками, у ньому наявний перелік всіх функцій які можна визвати у даному проекті, цей перелік доступний для виклику іншими додатками у майбутньому, та зроблений спеціально для цього, оскільки без нього неможливо викликати будь-яку функцію із інших частин проекту. Такий підхід зроблений спеціально для забезпечення інформаційної безпеки.



Рис 3.13. Структура компоненту Domain та приклад сутності «продукт»

Дотримання кращих практик при створенні серверної частини допоможе принцип інверсії залежності, який виконує вищеописану функцію та із коробки наявний у кожному проекті створеному на базі .Net Core.

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		41

Після того як серверна частина розроблена, можна завантажувати готовий програмний продукт на платформу Amazon. Створення необхідних для роботи додатку ресурсів було описано у другому розділі, отож після цього залишається лише додати створені бібліотеки в архів та завантажити на віртуальну машину, після чого, скористувавшись адресою за якою розташована віртуальна машина можна перейти у браузер та побачити працюючий інтернет магазин з продажу комп'ютерної техніки вже на хостингу, та доступний в мережі інтернет.

3.3 Висновки до розділу 3

У третьому розділі було спроектовано та розроблено графічний інтерфейс інтернет-магазину з продажу комп'ютерної техніки на базі платформи Node.JS та мови програмування TypeScript, визначено та пріоретизовано основні та другорядні елементи веб сторінок, розроблено макети веб сторінок за якими повинна відбуватися верстка сайту.

Серверна частина була розроблена на базі платформи .Net та мови програмування с#, також було визначено, що для побудови програмного продукту необхідно використовувати підхід DDD, який полегшує процес розробки та тестування, окрім цього було визначено необхідність розділити структуру проекту на окремі логічні одиниці, що в свою чергу також полегшує тестування та найголовніше масштабування програмного продукту.

Отже, у даному розділі було спроектовано та реалізовано практичну частину програмного продукту – інтернет-магазину з продажу комп'ютерної техніки, завдяки чому не можна не відмітити практичне значення розробленої програми, а саме – готове автоматизоване рішення для власників стаціонарних магазинів техніки, які вже зараз можуть перейти на освоєння інтернет-торгівлі за допомогою розробленого продукту.

					КНТЕУ 121 06-15.БР	Аркуш
						42
Зм.	Аркуш	№ докум	Підпис	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Проаналізувавши ситуацію на сучасному ринку було визначено що все більше набирає популярність перехід стаціонарних бізнесів у мережу Інтернет. Було визначено що автоматизація не тільки полегшує процес придбання товарів та послуг для звичайних користувачів, а й ще збільшує обороти самих бізнесів, що і є головною метою підприємців. Отже було запропоновано автоматизувати процес роботи магазину з продажу комп'ютерної техніки, розробивши програмне забезпечення відповідно інтернет-магазин з продажу комп'ютерної техніки.

В ході дослідження було визначено, що для створення інтернет-магазину необхідна хостинг-платформа, отож було розглянуто перелік хмарних середовищ які дозволяють підтримувати роботу веб-додатку. У якості бажаної платформи обрано Amazon Web Services, оскільки саме ця платформа має найширший діапазон доступних сервісів для інтеграції із власним програмним продуктом.

Оскільки інтернет-магазин з продажу комп'ютерної техніки передбачає зберігання певної кількості даних про наявні товари, було оглянуто та проаналізовано існуючі СУБД які можна використати при розробці веб-додатку, та обрано MySQL.

Після чіткої постановки задачі, було спроектовано та розроблено логічну та фізичну моделі бази даних, на основі яких виконана побудова самої бази даних, налаштування зв'язків між таблицями та встановлені типи даних полів у таблицях.

Так як інтернет-магазин являє собою перш за все веб-додаток, було розроблено макети веб-сторінок, за якими повинна відбуватися верстка сайту, та відповідно розроблено сам графічний інтерфейс за створеними макетами на базі платформи Node.JS та мови програмування TypeScript. Серверна частина була

					<i>КНТЕУ 121 06-15.БР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка інтернет магазину з продажу комп'ютерної техніки	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					ВП	43	56
Керівник	Рзаєва С.Л.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М.О.							
Розроб.	Єрохін В.О..							
					<i>Висновки та пропозиції</i>			

розроблена на базі платформи .Net Core та мови програмування C#, повинна базуватися на DDD підході, який полегшує процес розробки та тестування, окрім цього було визначено необхідність розділити структуру проекту на окремі логічні одиниці, що в свою чергу також полегшує тестування та найголовніше масштабування програмного продукту.

Врешті решт дотримуючись технічного завдання та використовуючи кращі архітектурні рішення, було спроектовано та розроблено програмний продукт – інтернет-магазин з продажу комп'ютерної техніки, який хоститься на платформі Amazon та повністю готовий до використання.

Серед пропозицій щодо розширення функціоналу програмного продукту можна додати наступне: у майбутніх релізах можна розширити кількість сторінок доступних у графічному інтерфейсі, наприклад, додати сторінку порівнянь товарів за характеристиками, додати вкладку відео-переглядів товарів, додати можливість додавати відео із розпакуванням та фото отриманого продукту від клієнтів та покупців. Із точки зору серверної частини, вдосконалити API інтерфейс проекту та інтегруватися із іншими програмними продуктами.

За допомогою розробленого програмного продукту будь-який власник стаціонарного магазину з продажу комп'ютерної техніки може легко перейти на інтернет-торгівлю використовуючи створений веб-додаток, що тільки збільшить фінансові обороти підприємця та надасть більшої популярності саме цьому магазину, оскільки у час сучасних технологій надання послуг та товарів все швидше та швидше переходить у мережу Інтернет.

					КНТЕУ 121 06-15.БР	Аркуш
						44
Зм.	Аркуш	№ докум	Підпис	Дата		

Список використаних джерел

Основний

1. Herbert Schildt The Complete Reference C# 4.0: by The McGraw-Hill Companies / Herbert Schildt, 2010. – 976 pages.
2. William R. Stanek Microsoft SQL Server 2008: Administrator's Pocket Consultant : Microsoft Press / William R. Stanek, 2008. – 720 pages.
3. Kalen Delaney, Paul S. Randal Microsoft® SQL Server® 2008 Internals (Developer Reference): Microsoft Press / Kalen Delaney, Paul S. Randal, 2009. - 784 pages.
4. Nicholas Petreley, Alex Khaloghli, Scott Mace. Ashton-Tate/Microsoft SQL Server: PC databases enter a new era: InfoWorld Media Group, Inc. / Nicholas Petreley, Alex Khaloghli, Scott Mace, 1989. — 841 pages.
5. Karen Watterson. Soul of a new machine. Data Based Advisor: Advisor Publications / Karen Watterson, 1989 – 486 pages.
6. Ashton-Tate and Microsoft form long-term strategic alliance to market SQL Server for local area networks: Microsoft Press/ Microsoft, Ashton-Tate (13 January 1988) – 755 pages.
7. Microsoft and Ashton-Tate ship SQL Server Network Developer's Kit: Microsoft Press / Microsoft, Ashton-Tate (1 November 1988) – 750 pages.
8. Scott Mace. Microsoft Ships SQL Server 1.1: Info World Media Group, Inc. / Scott Mace, 1990. — 460 pages.
9. Scott Mace. Microsoft's SQL Server Faces Slow Initial Sales: InfoWorld Media Group, Inc. / Scott Mace, 1990. — 290 pages.
10. Scott Mace, Kristi Coale. Q+E 2.5 Delivers DDE Link for Excel-SQL Server: InfoWorld Media Group, Inc. / Scott Mace, Kristi Coale, 1990. — 90 pages.

					КНТЕУ 121 06-15.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка інтернет магазину з продажу комп'ютерної техніки	Стадія	Аркуш	Аркушів
Зав. кафедри		Криворучко О.В.				СД	45	56
Керівник		Рзаєва С.Л.				Факультет інформаційних технологій, 4 курс, 6 група		
Гарант		Цензура М.О.						
Розроб.		Єрохін В.О..			Список використаних джерел			

11. Louise Fickel. Microsoft offers client/server developers' group: InfoWorld Media Group, Inc. / Louise Fickel, 1991. — 90 pages.
12. Scott Mace. SQL Server sports better LAN supports. Microsoft offers SQL Commander: InfoWorld Media Group, Inc. / Scott Mace, 1991. — 91 pages.
13. Уолтерс Р. Коулс. SQL Server 2008: прискорений курс для професіоналів. / Р. Уолтерс, М. Коулс – М.: «Вільямс», 2008. – С. 768.
14. Роберт Вієйра. Програмування баз даних Microsoft SQL Server 2005. Базовий курс = Beginning Microsoft SQL Server 2005 Programming. - М.: «Діалектика»/ Роберт Вієйра, 2007 – 500 с.
15. Майк Гандерлой, Джозеф Джорден, Дейвід Чанц. Освоєння Microsoft SQL Server 2005 = Mastering Microsoft SQL Server 2005. - М.: «Діалектика»/ Майк Гандерлой, Джозеф Джорден, Дейвід Чанц, 2007. – 650 с.
16. Microsoft® SQL Server™ 2005. Реалізація і обслуговування. Навчальний курс Microsoft (Іспит 70-431). / Microsoft®, 2007. - С. 767.
17. Петкович Д. Microsoft SQL Server 2008. Керівництво для початківців. - Microsoft®/ Петкович Д, 2009. - С. 752.
18. Джеймс Р. Грофф, Пол Н. Вайнберг, Ендрю Дж. Опель. SQL: повне керівництво, 3-є видання SQL: The Complete Reference, Third Edition. - М.: «Вільямс» / Джеймс Р. Грофф, Пол Н. Вайнберг, Ендрю Дж. Опель, 2014. - 960 с.
19. Кріс Фіайлі. SQL: Керівництво по вивченню мови. - М.: Peachpit Press / Кріс Фіайлі, 2003. - 456 с.
20. Аллен Тейлор. SQL для чайників, 8-е видання SQL For Dummies, 8th edition. - М.: «Діалектика» / Аллен Тейлор, 2014. - 416 с.
21. К. Дж. Дейт. Введення в системи баз даних Пер. з англ. - 8-е изд. - М.: Вільямс / К. Дж. Дейт, 2005. - 1 328 с.
22. Chamberlin, Donald D. Early history of SQL. IEEE Annals of the History of Computing / Chamberlin, Donald D, (2012): 78-82.

					КНТЕУ 121 06-15.БР		Аркуш
							46
Зм.	Аркуш	№ докум	Підпис	Дата			

Інтернет ресурси

1. Порівняння сучасних СУБД. [Електронний ресурс] – Режим доступу : <http://drach.pro/blog/hi-tech/item/145-db-comparison>
2. Порівняння хмарних середовищ Amazon та Azure. [Електронний ресурс] – Режим доступу : <https://www.educba.com/aws-vs-azure/>

					КНТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		47

Додатки

Додаток А

```

namespace WebApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class OrderController : ControllerBase
    {
        private readonly IOrderService _orders;

        public OrderController(IOrderService order)
        {
            _orders = order;
        }

        [HttpPost]
        public async Task<ActionResult<int>> AddOrder([FromBody] CreateOrderRequest request)
        {
            var orderedProductsIds = request.ProductsIds;
            var orderId = await _orders.AddOrder(orderedProductsIds);

            if (orderId == 0)
                return BadRequest("You can not make an order for a non-existent game");

            return Ok(orderId);
        }

        [HttpPut]
        [Route("{orderId:int:min(1)}")]
        public async Task<IActionResult> EditOrder(int orderId, [FromBody] EditOrderRequest request)
        {
            await _orders.EditOrder(orderId, request.ProductsIds);
            return NoContent();
        }
    }
}

```


[HttpGet]

public async Task<ActionResult<List<OrderResponse>>> GetAllOrders()

{

var (orders, products) = **await** _orders.GetAll();

var ordersResponse = orders.Select(p => **new** OrderResponse

{

OrderId = p.OrderId,

OrderDate = p.OrderDate,

Status = p.Status,

OrderedProducts = products.Where(x => p.Products.Select(y =>

y.ProductId).Contains(x.ProductId)).Select(

z => z.ToProductResponse()).ToList()

}).ToList();

return Ok(ordersResponse);

}

[HttpGet]

[Route("history")]

[Authorize(Roles = "manager")]

public async Task<ActionResult<List<OrderResponse>>> GetOrdersHistory()

{

var ordersHistory = **await** _orders.GetOrdersHistory();

return Ok(ordersHistory.ToOrderResponse());

}

[HttpGet]

[Route("{id:int:min(1)}")]

public async Task<ActionResult<OrderResponse>> GetOrderById(int id)

{

var order = **await** _orders.GetOrderById(id);

return Ok(order.ToOrderResponse());

}

}


```

    }
    namespace BLL.Services
    {
        public class OrderService : IOrderService
        {
            private readonly IUnitOfWork _unit;

            public OrderService(IUnitOfWork unit)
            {
                _unit = unit;
            }

            public async Task<int> AddOrder(int[] productIds)
            {
                var foundProducts = await CheckProductsAvailability(productIds);
                var utcNow = DateTime.UtcNow;

                var orderRequest = new Order { OrderDate = utcNow };
                var order = await _unit.OrderRepository.Add(orderRequest);

                var products = foundProducts.Select(p => new OrdersProduct { OrderId = order.OrderId,
                ProductId = p.ProductId }).ToList();
                order.Products = products;

                await _unit.Commit();

                return order.OrderId;
            }

            public async Task EditOrder(int orderId, int[] productsIds)
            {
                var products = await CheckProductsAvailability(productsIds);

                await _unit.OrderRepository.Update(orderId, products);
                await _unit.Commit();
            }
        }
    }

```

```
}
```

```
public async Task<(IList<Order>, IList<Product>)> GetAll()
```

```
{
```

```
    var orders = await _unit.OrderRepository.GetAll();
```

```
    var products = await _unit.ProductRepository.GetProductsByIds(orders.SelectMany(p =>
        p.Products.Select(x => x.ProductId)).ToArray());
```

```
    return (orders, products);
```

```
}
```

```
public async Task<IList<Order>> GetOrdersHistory()
```

```
{
```

```
    var result = await _unit.OrderRepository.GetOrdersHistory();
```

```
    return result.ToList();
```

```
}
```

```
public async Task<Order> GetOrderById(int orderId)
```

```
{
```

```
    var tempOrder = await _unit.OrderRepository.GetById(orderId);
```

```
    return tempOrder;
```

```
}
```

```
private async Task<List<Product>> CheckProductsAvailability(int[] productIds)
```

```
{
```

```
    var foundProducts = await _unit.ProductRepository.GetProductsByIds(productIds);
```

```
    if (foundProducts.Count != productIds.Length)
```

```
        throw new ArgumentException(nameof(productIds));
```

```
    foreach (var product in foundProducts)
```

```
    {
```

```
        if (!product.Availability)
```

```
            throw new ArgumentException(nameof(productIds));
```

```
    }
```



```

        return foundProducts;
    }
}

namespace DAL.Repositories
{
    internal sealed class ProductRepository : IProductRepository
    {
        private readonly StoreContext _context;

        public ProductRepository(StoreContext context)
        {
            _context = context;
        }

        public async Task<List<Product>> GetProductsByManufacturerId(int manufacturerId)
        {
            var games = await _context.Products.AsNoTracking().Where(p => p.ManufacturerId ==
manufacturerId)
                .GetAllProductProperties()
                .ToListAsync();
            return games;
        }

        public async Task<List<Product>> GetProductsByIds(int[] productsIds)
        {
            return await _context.Products
                .GetAllProductProperties()
                .Where(p => productsIds.Contains(p.ProductId)).ToListAsync();
        }

        public async Task<List<Product>> ProductsWithCategoryId(int categoryId)
        {
            var games = await _context.Products.AsNoTracking().Where(p => p.CategoryId ==
categoryId)
                .GetAllProductProperties()

```



```

        .ToListAsync();

    return games;
}

public async Task<Product> GetById(int productId)
{
    var product = await _context.Products
        .GetAllProductProperties()
        .FirstOrDefaultAsync(p => p.ProductId == productId);

    return product ?? throw new EntryNotFoundException($"Product with ProductId={productId}
was not found");
}

public async Task<Product> FindById(int productId)
{
    var game = await _context.Products
        .GetAllProductProperties()
        .FirstOrDefaultAsync(p => p.ProductId == productId);

    return game;
}

public async Task<List<Product>> GetAll()
{
    var result = await _context.Products.AsNoTracking()
        .GetAllProductProperties()
        .ToListAsync();

    return result;
}

public async Task<Product> Add(Product game)
{
    var gameEntity = await _context.Products.AddAsync(game);

```

```

        return gameEntity.Entity;
    }

    public async Task UploadImages(int productId, List<string> imagesNames)
    {
        var images = imagesNames.Select(p => new Image()
        {
            ImageKey = p,
            ProductId = productId
        });
        await _context.Images.AddRangeAsync(images);
    }

    public async Task UploadImage(int productId, string imageName, bool isMain)
    {
        var image = new Image()
        {
            ImageKey = imageName,
            ProductId = productId,
            IsMain = isMain
        };
        await _context.Images.AddRangeAsync(image);
    }

    public async Task Update(int productId, Product updatedGame)
    {
        var modifiedObj = await _context.Products.FirstOrDefaultAsync(p => p.ProductId ==
productId);
        updatedGame.ProductId = productId;
        _context.Entry(modifiedObj).CurrentValues.SetValues(updatedGame);
    }

    public async Task Delete(int productId)

```



```

    {
        await _context.Database.ExecuteSqlCommandAsync("DELETE FROM Products WHERE
ProductId=@productId",
        new MySqlParameter("@productId", productId));
    }

    public async Task<List<Category>> GetGameGenres(int productId)
    {
        var result = await _context.Categories.FromSql(
            "SELECT ge.CategoryId, ge.Name FROM Categories ge INNER JOIN Categories gg ON
ge.CategoryId = gg.CategoryId WHERE gg.ProductId = @productId",
            new MySqlParameter("@productId", productId)).ToListAsync();
        return result;
    }

    public async Task LeaveComment(int productId, Comment comment)
    {
        await _context.Comments.AddAsync(comment);
    }

    public async Task<List<Product>> GetGamesWithFilters(FilterModel filter)
    {
        IOrderedEnumerable<Product> filteredResult = null;

        var result = await _context.Products.Where(p =>
            filter.Publishers.Contains(p.Manufacturer.Name) &&
            p.Price > filter.PriceFilter.From && p.Price <= filter.PriceFilter.To).ToListAsync();

        if (filter.IsMostCommented)
        {
            filteredResult = result.OrderBy(p => p.Comments.Count);
        }
        else if (filter.IsMostPopular)
        {
            filteredResult = result.OrderBy(p => p.AmountOfViews);
        }
    }

```



```
    }  
    else if (filter.ByPriceAscending)  
    {  
        filteredResult = result.OrderBy(p => p.Price);  
    }  
    else if (filter.ByPriceDescending)  
    {  
        filteredResult = result.OrderByDescending(p => p.Price);  
    }  
    else if (filter.ByDateAscending)  
    {  
        filteredResult = result.OrderBy(p => p.DateOfAdding);  
    }  
    else if (filter.ByDateDescending)  
    {  
        filteredResult = result.OrderByDescending(p => p.DateOfAdding);  
    }  
    return filteredResult?.Skip((filter.Page - 1) * filter.Size).Take(filter.Size).ToList();  
}  
}  
}
```