

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**До захисту допущено:
Завідувач кафедри**

_____ Сергій СТИРЕНКО

“ _____ ” _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою

**“Інженерія програмного забезпечення комп’ютерних систем”
спеціальності 121 “Інженерія програмного забезпечення”**

на тему: Програмний застосунок для реалізації процесів електронної комерції.

Виконала:

студентка IV курсу, групи ПП-83
Расюк Альона Володимирівна

Керівник:

д.т.н., професор, лауреат державної премії України
в галузі науки і техніки, професор кафедри ОТ ФІОТ
Писарчук Олексій Олександрович

Консультант (нормоконтроль):

д.т.н, професор
Сімоненко Валерій Павлович

Рецензент:

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка)

Київ – 2022 р.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Інженерія програмного забезпечення комп’ютерних систем”

спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій СТИРЕНКО

“__” _____ 2022 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт студентки

Расюк Альони Володимирівни

1. Тема проєкту Програмний застосунок для реалізації процесів електронної комерції.

керівник проєкту доктор технічних наук, професор, лауреат державної премії України в галузі науки і техніки, професор кафедри ОТ ФІОТ Писарчук Олексій Олександрович, затверджені наказом по університету від _____ 2022 року № _____

2. Термін здачі студенткою закінченого проєкту _____ 2022 р.

3. Вихідні дані до проєкту: технічна документація, існуючі реалізації процесів електронної комерції.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються) аналіз сучасного стану процесів електронної комерції, аналіз відомих рішень щодо реалізації процесів електронної комерції, архітектура та програмна розробка програмного застосунку для реалізації процесів електронної комерції, прикладні

аспекти застосування програмного застосунку для реалізації процесів електронної комерції.

5. Перелік графічного матеріалу: структурна схема програмного продукту, UML-схема функціонування програмного продукту, діаграма класів, програмна документація, презентаційні матеріали

6. Консультанти проєкту, з вказівкою розділів проєкту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Нормоконтроль</i>	<i>д.т.н, проф. Сімоненко В. П.</i>		

7. Дата видачі завдання _____

Календарний план

№ п/п	Найменування етапів дипломного проєкту	Терміни виконання етапів проєкту	Примітки
1.	<i>Затвердження теми проєкту</i>	<i>10.12.2022-15.12.2022</i>	
2.	<i>Вивчення та аналіз сучасного стану процесів електронної комерції та відомих рішень щодо реалізації процесів електронної комерції</i>	<i>15.12.2022-15.03.2022</i>	
3.	<i>Розробка архітектури та загальної структури програмного застосунку для реалізації процесів електронної комерції</i>	<i>15.03.2022-25.03.2022</i>	
4.	<i>Програмна розробка застосунку для реалізації процесів електронної комерції</i>	<i>25.03.2022-15.04.2022</i>	
5.	<i>Оформлення пояснювальної записки</i>	<i>15.04.2022-20.05.2022</i>	
6.	<i>Захист програмного продукту</i>		
7.	<i>Передзахист дипломного проєкту</i>		
8.	<i>Захист дипломного проєкту</i>		

Студент-дипломник

_____ Альона Расюк

Керівник проєкту

_____ Олексій ПИСАРЧУК

АНОТАЦІЯ

В бакалаврському дипломному проєкті проаналізовано сучасний стан процесів електронної комерції. Додатково було детально проаналізовано відомі рішення щодо реалізації процесів електронної комерції. Згідно з результатами досліджень спроектовано архітектуру програмного застосунку для реалізації процесів електронної комерції. Відповідно до архітектури розроблено застосунок для реалізації процесів електронної комерції.

Програма здійснює систематизоване формування та зберігання інформації про основні процеси електронної комерції та реалізацію клієнт-серверних запитів.

Розроблена система орієнтована на стартапи малого бізнесу, та відповідає умовам кросплатформності, швидкої розгортки і надійності.

Складовими частинами системи є серверна частина, база даних, мобільна версія застосунку та інструментарій для інтеграції з різноманітними обліковими системами.

ANNOTATION

The bachelor's thesis project analyzes the current state of e-commerce processes. Additionally, well-known e-commerce solutions were analyzed in detail. According to the results of the research, the architecture of the software application for the implementation of e-commerce processes was designed. in accordance with the architecture, an e-commerce application has been developed.

The program carries out the systematic formation and storage of information about the main processes of e-commerce and the implementation of client-server requests.

The developed system is focused on small business startups, and meets the conditions of cross-platform, fast deployment and reliability.

The components of the system are the server part, database, mobile version of the application and tools for integration with various accounting systems.

ВІДОМІСТЬ ТЕХНІЧНОГО ПРОЄКТУ

Довідки	Формат	Значення	Найменування	Кіл. листів	№ екземпляра	Додаток
			Документація загальна			
	<i>A4</i>	<i>ІАЛЦ.467200.001 ТП</i>	Відомість технічного проекту	1		
	<i>A4</i>	<i>ІАЛЦ.467200.002 ТЗ</i>	Технічне завдання	3		
	<i>A4</i>	<i>ІАЛЦ.467200.003 ПЗ</i>	Пояснювальна записка	55		
	<i>A1</i>	<i>ІАЛЦ.467200.004 Д1</i>	Р-схема алгоритму	1		
	<i>A1</i>	<i>ІАЛЦ.467200.005 Д2</i>	Функціональна схема (діаграма класів)	1		
	<i>A1</i>	<i>ІАЛЦ.467200.006 Д3</i>	Структурна схема	1		
	<i>A4</i>	<i>ІАЛЦ.467200.007 Д4</i>	Опис програми	3		

					ІАЛЦ.467200.001 ТП		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розробив		<i>Расюк А.В.</i>			Літ.	Арк.	Акрушіє
Перевірів		<i>Писарчук О.О.</i>					
Реценз.					НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІП-83		
Н.контроль		<i>Симоненко В.П.</i>					
Затв.							
					Програмний застосунок для реалізації процесів електронної комерції Відомість технічного проекту		

ТЕХНІЧНЕ ЗАВДАННЯ

ДО ДИПЛОМНОГО ПРОЕКТУ

**на тему: «Програмний застосунок для реалізації процесів
електронної комерції»**

Київ – 2022 р.

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ.....	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	2
5.1. Вимоги до розробленого продукту.....	2
5.2. Вимоги до апаратної частини.....	3
6. ЕТАПИ РОЗРОБКИ	3

					ІАЛЦ.467200.002 ТЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розробив		Расюк А.В.			Програмний застосунок для реалізації процесів електронної комерції Технічне завдання	Літ.	Арк.	Акрушів
Перевірів		Писарчук О.О.				Т	1	3
Реценз.						НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІП-83		
Н. контроль		Симоненко В.П.						
Затв.								

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

Це технічне завдання поширюється на розробку програмного застосунку для реалізації процесів електронної комерції. Областю застосування цього проєкту є діяльність торгівельних підприємств. Продукт виступає у якості альтернативи існуючим системам електронної комерції.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврського проєкту по освітньо-професійної програми “Інженерія програмного забезпечення комп’ютерних систем” спеціальності 121 “Інженерія програмного забезпечення”, затверджене кафедрою Обчислювальної техніки Національного технічного Університету України “Київський Політехнічний інститут імені Ігоря Сікорського”.

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проєкту є автоматизація професійної діяльності торгівельних підприємств із впровадженням технологій глобальних інформаційних систем та процесів.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є технічна документація, веб-довідка, технічна література з розробки баз даних, веб-розробки та веб-дизайну.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

- Кросплатформність – програмний продукт повинен функціонувати в Unix- подібних операційних системах, MAC-OS, та в Windows-системах.

					ІАЛЦ.467200.002 ТЗ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

- Надійність – продукт повинен бути відмовостійким.
- Швидке розгортання та зручність використання – продукт повинен бути компактним та простим для користувача.

5.2. Вимоги до апаратної частини

- Оперативної пам'яті не менше 512 Мбайт для клієнта та 1Гб для сервера
- Вільний простір жорсткого диска не менше 800 Мбайт для сервера
- Підключення до Інтернету
- Веб браузер з підтримкою HTML5 та javascript.

6. ЕТАПИ РОЗРОБКИ

Етапи:	Дата
1. Вивчення літератури	20.12.2021
2. Складання і узгодження технічного завдання	15.01.2022
3. Аналіз сучасного стану процесів електронної комерції	27.01.2022
4. Розробка архітектури та програмна розробка застосунку для реалізації процесів електронної комерції	14.04.2022
5. Тестування програмного застосунку	01.05.2022
6. Відлагодження і виправлення помилок	12.05.2022
7. Оформлення документації дипломного проєкту	06.06.2022

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО ДИПЛОМНОГО ПРОЕКТУ

**на тему: «Програмний застосунок для реалізації процесів
електронної комерції»**

Київ – 2022 р.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	2
ВСТУП.....	3
РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ПРОЦЕСІВ ЕЛЕКТРОННОЇ КОМЕРЦІЇ	5
1.1. Аналіз змісту процесів електронної комерції.	5
1.2. Аналіз відомих рішень щодо реалізації процесів електронної комерції.	8
1.3. Формалізація задачі та постановка часткових задач досліджень.	12
Висновок за 1 розділом.	14
РОЗДІЛ 2. АРХІТЕКТУРА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ДЛЯ РЕАЛІЗАЦІЇ ПРОЦЕСІВ ЕЛЕКТРОННОЇ КОМЕРЦІЇ	16
2.1 Розробка архітектури програмного застосунку для реалізації процесів електронної комерції.....	16
2.1.1. Інженерія вимог.	16
2.1.2. Архітектурне проектування програмного застосунку.	18
2.2. Програмна реалізація програмного застосунку для реалізації процесів електронної комерції.....	19
Висновок за 2 розділом.	34
РОЗДІЛ 3. ПРИКЛАДНІ АСПЕКТИ ЗАСТОСУВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ ДЛЯ РЕАЛІЗАЦІЇ ПРОЦЕСІВ ЕЛЕКТРОННОЇ КОМЕРЦІЇ.....	37
3.1. Демонстрація практичних можливостей розробленого програмного застосунку для реалізації процесів електронної комерції.	37
3.2. Програмна документація на програмний застосунок.	39
Висновок за 3 розділом.	50
ВИСНОВОК.	51
СПИСОК ВИКОРИСТАНИХ ПЕРІОДИЧЕРЕЛ – ЛІТЕРАТУРИ	53

ІАЛЦ.467200.003 ПЗ				
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
<i>Розробив</i>		<i>Расюк А.В.</i>		
<i>Перевірів</i>		<i>Писарчук О.О.</i>		
<i>Реценз.</i>				
<i>Н.контроль</i>		<i>Симоненко В.П.</i>		
<i>Затв.</i>				
Програмний застосунок для реалізації процесів електронної комерції Пояснювальна записка				
		<i>Літ.</i>	<i>Арк.</i>	<i>Акрушіє</i>
		Т	1	55
НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІП-83				

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІС – інформаційна система

ОС – операційна система.

OS – operation system.

UI – User Interface, користувацькій інтерфейс, інтерфейс, який забезпечує передачу даних та взаємодію між користувачем та ІС.

APK - Android Package, це формат файлу, який використовується в операційній системі Android для встановлення мобільних додатків.

REST - Representational State Transfer, передача репрезентативного стану

API - Application Programming Interface, прикладний програмний інтерфейс, інструментарій для взаємодії та інтеграції частин ІС.

CMS – Content management system, система управління контентом, технологія для заповнення сайту даними

MVC – Model-View-Controller – (модель-представлення-контролер) архітектура для CMS

B2B - бізнес-бізнес, тип систем е-комерції

B2G - бізнес-держава, тип систем е-комерції

B2C - бізнес-клієнт, тип систем е-комерції

C2C - клієнт-клієнт, тип систем е-комерції

G2B - держава - бізнес, тип систем е-комерції

G2C - держава - клієнт, тип систем е-комерції

C2G - клієнт-держава, тип систем е-комерції

CRUID – Create, Update, Insert, Delete: функції керування даними

СУБД – система управління базою даних

CDN - Content delivery network. Мережа доправлення контенту

JS – JavaScript, мова програмування

МІТ – вид ліцензії

HTTP HyperText Transfer Protocol – протокол Інтернета

UML Unified Modeling Language – узагальнена мова моделювання.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Метою досліджень у дипломному проєкті є автоматизація професійної діяльності торговельних підприємств із впровадженням технологій глобальних інформаційних систем та процесів.

Предметом досліджень є технології глобальних інформаційних процесів.

Об'єктом дипломного проєкту є процес інформаційної взаємодії за технологіями глобальних інформаційних систем. Результатом цього процесу взаємодії є інтеграція за допомогою API між звичними для підприємств системами обліку товарів та розробленим програмним застосунком. Інтеграція за допомогою API дозволяє автоматизувати процес додавання товарів до ІС, актуалізувати ціни на товари та їх кількість у ІС та зменшити таким чином вплив «людського фактора».

Однією з найважливіших тенденцій, які зараз переживає світ бізнесу, є впровадження процесів електронної комерції[1]. Враховуючи стрімкий розвиток Інтернет-технологій, електронна комерція нарощує свої обсяги та формується як окрема галузь економіки. З кожним днем все більше торговельних підприємств впроваджують у свою діяльність процеси електронної комерції, що дозволяє їм розвиватися: збільшувати прибуток та залучати нових клієнтів.

Також росте кількість споживачів, які є учасниками процесів електронної комерції. Таким чином, процеси електронної комерції стають дедалі актуальнішими.

На сьогоднішній день існує багато різноманітних рішень для реалізації процесів електронної комерції для широкого спектру підприємств: від фізичних осіб-підприємців до учасників крупного бізнесу. Але питання зручності, надійності систем та відповідність рівню залишаються актуальними завжди. Орієнтиром для результату в цьому дипломному проєкті є насамперед стартапи малого бізнесу.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

У проекті сформульовані наступні завдання:

1. Аналіз сучасного стану процесів електронної комерції.
2. Аналіз відомих рішень щодо реалізації процесів електронної комерції.
3. Розробка архітектури програмного застосунку для реалізації процесів електронної комерції.
4. Програмна розробка застосунку для реалізації процесів електронної комерції.

Дипломний проект складається зі вступу, трьох розділів, висновків до кожного із розділів та загальних висновків. У першому розділі детально проаналізовано існуючі системи для забезпечення процесів електронної комерції, порівняно рішення, аргументовано вибір технологій для реалізації завдання. У другому розділі описаний процес розробки архітектури та програмної реалізації застосунку. У третьому розділі демонструється результат розробки.

Основними структурними компонентами програмного застосунку є серверна частина, база даних, мобільний застосунок, веб-відображення ІС, АРІ інтеграція.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

РОЗДІЛ 1.

АНАЛІЗ СУЧАСНОГО СТАНУ ПРОЦЕСІВ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

1.1. Аналіз змісту процесів електронної комерції.

Електронна комерція – це термін, який використовується для опису комерційної діяльності в Інтернеті. Забезпечує можливість здійснення покупок, продажу, обслуговування, маркетингової діяльності за допомогою використання комп'ютерних мереж [2].

Електронна комерція (у широкому розумінні) — це підприємницька діяльність, яка здійснює комерційні операції з використанням електронних засобів обміну даними. Об'єктами електронної комерції є те, на що спрямована діяльність систем електронної комерції. Сюди входять різні товари, послуги та інформація.

За об'єктами та суб'єктами діяльності розрізняють такі типи систем електронної комерції: B2B (бізнес-бізнес), B2G (бізнес-держава), B2C (бізнес-клієнт), C2C (клієнт-клієнт), G2B (держава - бізнес), G2C (держава - клієнт), C2G (клієнт-держава) [3, 22].

Сьогодні найпоширенішими є системи B2B і B2C[23, 24].

Модель B2C є найпопулярнішою формою електронної комерції та включає онлайн-роздрібну торгівлю. Системи B2C включають, наприклад:

- веб-вітрини, що представляють розміщені в мережі каталоги товарів або товарів компанії, які мають мінімальні засоби замовлення;

- Інтернет-магазини, які містять, крім вітрини, всю необхідну бізнес-інфраструктуру для управління процесом електронної комерції через Інтернет - так званий бек-офіс. Організація торгівлі через інтернет-магазин дозволяє контролювати весь процес електронної комерції та зменшувати операційні витрати компанії[27,28];

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

- системи інтернет-магазинів, які представляють собою інтернет-магазини, які повністю інтегровані з торговими бізнес-процесами компанії.

Системи B2B включають рішення, які мають всі рівні електронної взаємодії на рівні компанії з використанням спеціальних технологій і стандартів електронного обміну даними. Моделі B2B реалізуються за такими схемами:

- електронні торгові майданчики (торгівельні хаби, електронний ринок);
- електронні склади (системи збуту та постачання, електронні закупівлі).

Основними елементами систем електронної комерції є:

1. Суб'єкти електронної комерції - учасники та замовники систем електронної комерції. Це фінансові установи, господарські організації (комерційні компанії) та споживачі.

2. Системи електронної комерції: системи торгівлі, оплати, арбітражу та доставки.

3. Процеси в електронній комерції:

- маркетингове дослідження;
- продаж товарів, послуг, інформації;
- виконання замовлень;
- розрахунки за угодами;
- підтримка комерційної діяльності.

4. Мережі електронної комерції: глобальний Інтернет, комерційні та корпоративні мережі.

Масштабний розвиток такого явища, як електронна комерція, пов'язано з перевагами, які воно надає як бізнесу, так і клієнтам.

По-перше, це глобальна присутність. Це дозволяє навіть невеликим компаніям вести бізнес по всьому світу.

По-друге – це підвищення конкурентоспроможності. Різні компанії використовують технології електронної комерції, щоб наблизити клієнта. Вони пропонують розширену та післяпродажну підтримку, можуть швидко реагувати на скарги клієнтів [27].

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

По-третє, персоналізація продажів. За допомогою електронної комерції компанії можуть отримувати інформацію про запити кожного окремого клієнта та автоматично надавати продукти та послуги, які відповідають його вимогам.

Крім того, перевірені методи кібернетичного аналізу та моделювання, пристрій штучних нейронних мереж надає електронному бізнесу цінну можливість моніторингу та аналізу даних про ринок і споживачів у режимі реального часу.

Ще одна перевага полягає в тому, що компанії швидко реагують на попит. Компанії пристосовуються до запитів споживачів, можуть швидко реагувати на їхні запити та надавати «зворотній зв'язок» [16].

Це не завжди так, але в деяких випадках можна говорити про зниження витрат. Укладання угоди в електронному вигляді зменшує витрати на технічне обслуговування. Це призводить до зниження цін для клієнтів.

Клієнтам електронної комерції можна надати такі переваги:

1. Світовий вибір. Клієнти отримують можливість вибору товарів і послуг усіх світових постачальників.
2. Можливість швидкого порівняння пропозицій та обміну інформацією з іншими користувачами.
3. Можливість замовляти товари та здійснювати операції в зручний для клієнтів час.

Всесвітня павутина зробила електронну комерцію доступною для компаній будь-якого рівня.

Якщо раніше організація електронного обміну даними вимагала значних вкладень в комунікаційну інфраструктуру і була можливою лише для великих компаній, то сьогодні використання Інтернету дозволяє поповнити ряди «електронних трейдерів» і малого бізнесу.

Кожен бізнес електронної комерції має власні вимоги і потребує індивідуального налаштування. Однак загальний процес електронної комерції можна грубо розділити на 4 кроки:

- 1) прийом замовлення;
- 2) обробка даних про замовлення та покупця;
- 3) доставка товару клієнту;
- 4) формування та оплата рахунку

Усі ці кроки можна автоматизувати. З метою прийому замовлень потрібен сайт з інтернет-магазином. Отримавши замовлення, дані повинні структуровано оброблятися. Наприклад, товар, який замовляє покупець повинен реєструватися та зберігатися. Також повинні зберігатися дані про покупців. Таким чином сайт має бути інтегрований з СУБД для зберігання, керування, та обробки інформації.

З метою автоматизованої доставки система електронної комерції може бути інтегрована за допомогою API з системами логістичних компаній.

Після обробки даних замовлення система повинна створити рахунок-фактуру. Система електронної комерції може бути інтегрована за допомогою API з платіжними системами.

Таким чином, в межах цього дипломного проекту має бути розроблений програмний застосунок, який реалізує такі процеси електронної торгівлі як прийом замовлення, обробка даних про замовлення та покупця, доставка товару клієнту, формування та оплата рахунку.

1.2. Аналіз відомих рішень щодо реалізації процесів електронної комерції.

Класикою додатків з електронної комерції є рішення, які засновані на технології PHP та бази даних MySQL або PostgreSQL. В основі цих систем покладена архітектура MVC. Концепція MVC була описана Трюгве Реенскаугом в 1978 році. Він працював у центрі «Хероx PARC» над мовою програмування «Smalltalk». Пізніше Стів Бурбек реалізував цю концепцію у Smalltalk-80 [20,21].

Остаточна версія концепції MVC була опублікована лише 1988 року. Згодом ця схема проектування стала розвиватися. Подальший виток

					ІАЛЦ.467200.003 ПЗ	Арк. 8
Змн.	Арк.	№ докум.	Підпис	Дата		

популярності привніс розвиток фреймворків, орієнтованих на швидку розгортку, мовами Python, PHP і Ruby. На момент 2017 року, фреймворки з MVC зайняли помітні позиції щодо інших фреймворків без цього шаблону.. (рис. 1.1).

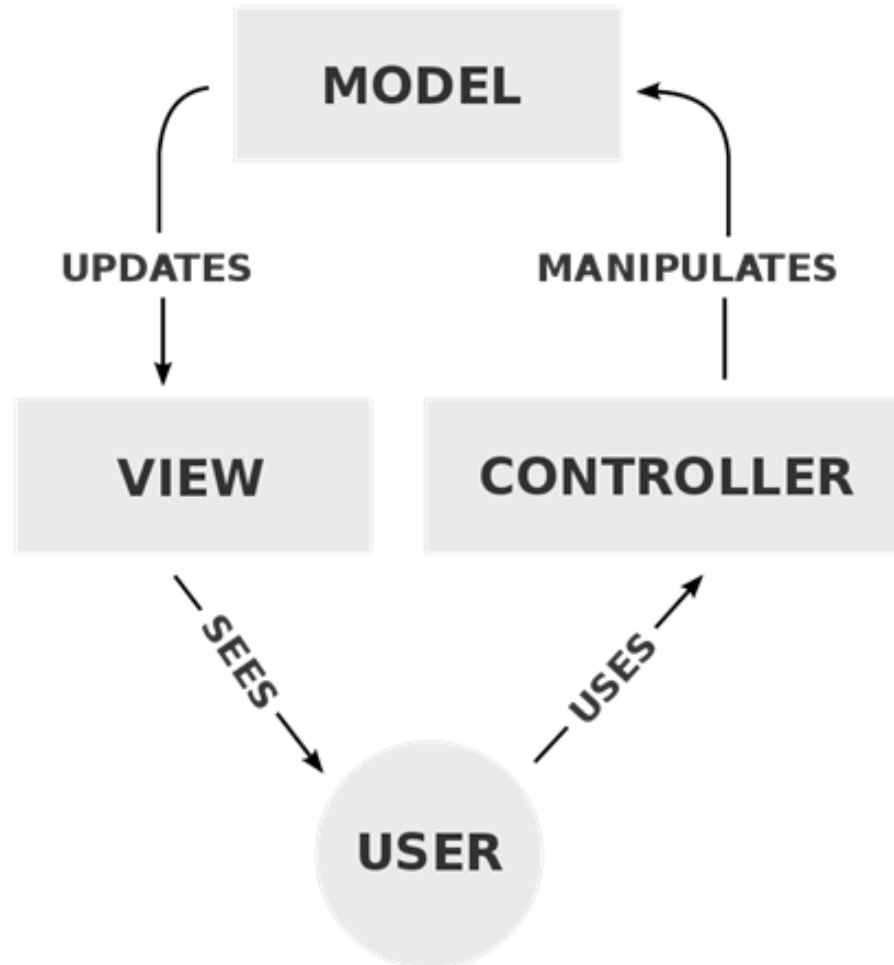


Рисунок 1.1. Архітектура MVC [4]

В даній архітектурі *Модель* – це центральний компонент шаблону. Це динамічна структура даних, яка не залежить від інтерфейсу користувача. Він безпосередньо керує даними та логікою програми.

В цій архітектурі *Вид* є будь-яке відображення інформації, наприклад стаття, діаграма, або таблиця. Можливі кілька переглядів однієї і тієї ж інформації, наприклад, стовпчаста діаграма для управління та табличне представлення для бухгалтерів. До однієї моделі можна приєднати кілька видів.

Змн.	Арк.	№ докум.	Підпис	Дата

Контролер – приймає введення даних та перетворює ці дані в команди для моделі або вида.

Архітектура MVC також детермінує взаємодію між моделю, видом та контролером.

Цю класичну архітектуру крім PHP запозичено у платформах ASP.NET (Core), NodeJS.

Ряд розробників спеціалізується лише в одній із областей: або розробляють графічний інтерфейс, або розробляють бізнес-логіку. Тому можна досягти того, що програмісти, які займаються розробкою бізнес-логіки (моделі), взагалі не будуть обізнані про те, яке використовуватиметься представлення [17,19].

У якості категорій для аналізу програмних рішень з електронної комерції у даному проекті використовуються:

- 1) технологія та СУБД (а саме PHP та MySQL, ASP.NET Core, NodeJS та MongoDB)
- 2) рівень бізнесу (наприклад: малий, середній, великий)
- 3) ліцензія продукту (наприклад, Open Source (OSL 3.0), або комерційна)
- 4) особливості архітектури
- 5) розповсюдженість
- 6) масштабованість
- 7) швидкодія
- 8) трудомісткість процесу розробки

У якості об'єктів для аналізу у даному проекті використовуються найбільш розповсюджені системи електронної комерції:

- 1) Magento від Adobe
- 2) CMS WordPress та плагин WooCommerce
Shopify

3) Grandnode від NopCommerce

4) KeystoneJS

Статистика розповсюдженості приведена на рис. 1.2

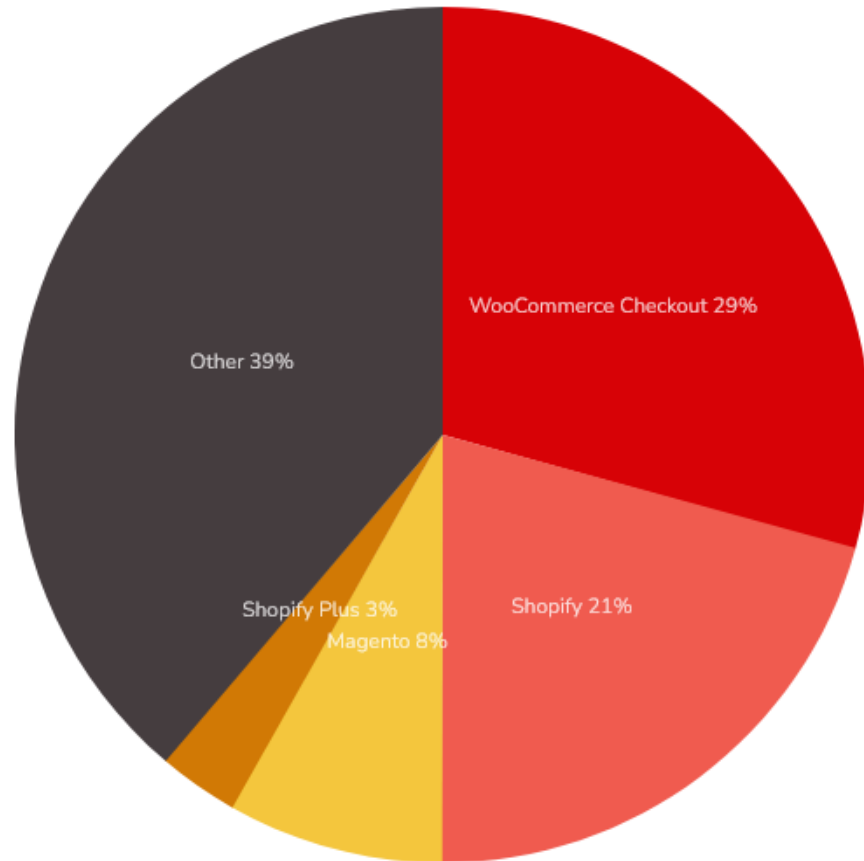


Рисунок 1.2. Статистика розповсюдженості існуючих застосунків для електронної комерції[5].

Magento є багат шаровим програмним забезпеченням. Заснована на PHP та MySQL або PostgreSQL. Найбільш розповсюджена серед крупного бізнесу у якості маркетплейсів.

На найвищому рівні архітектура продукту Magento складається з основного коду продукту та додаткових модулів [6]. Ці додаткові модулі покращують або замінюють основний код продукту.

Модулі організують код, який підтримує конкретне завдання або функцію. Модуль може включати код для зміни зовнішнього вигляду вашої вітрини, а також її фундаментальної поведінки.

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

11

Модулі розробників функціонують з основним кодом продукту Magento, який організовано за шарами.

Розуміння багатoshарової моделі програмного забезпечення є важливим для розуміння базової організації продукту Magento.

Архітектурна діаграма Magento представлена на рис. 1.3

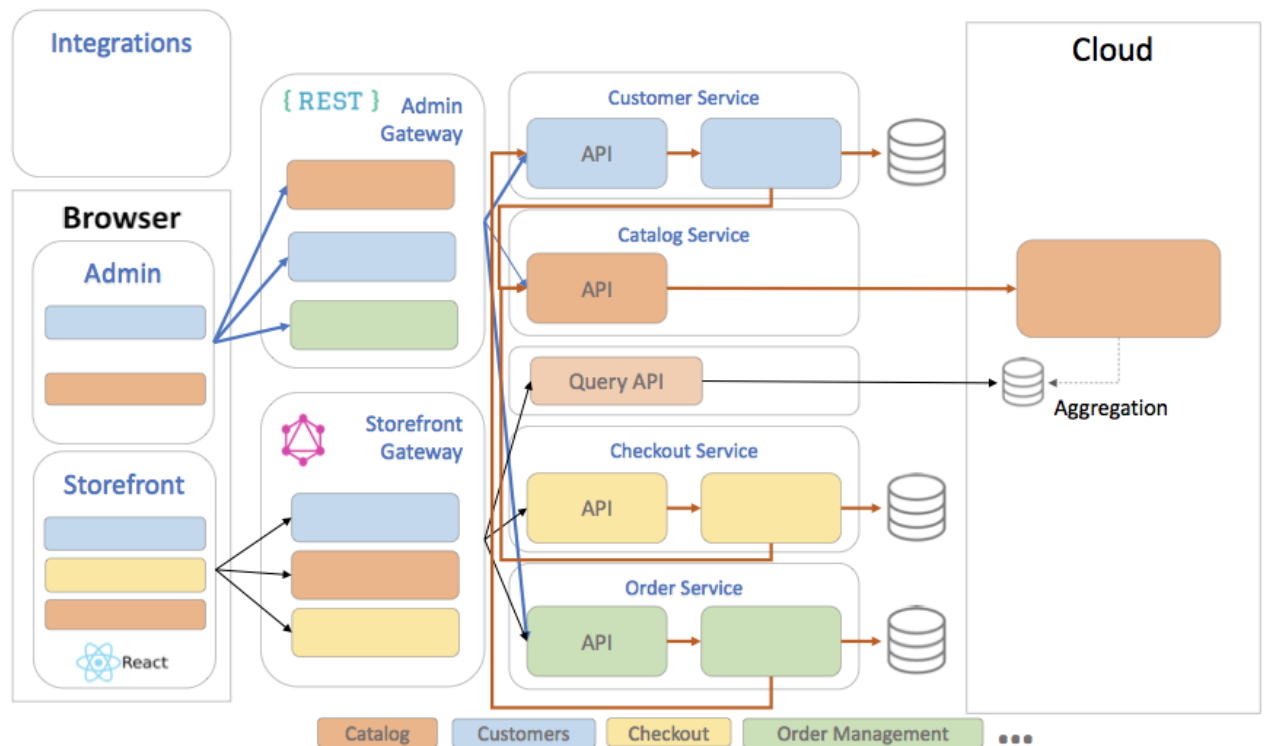


Рисунок 1.3. Архітектурна діаграма Magento [6].

У основі застосунків що використовують WordPress та WooCommerce полягає універсальна CMS WordPress (архітектура MVC) та плагін WooCommerce.

WordPress та WooCommerce найбільш розповсюджені серед малого та середнього бізнесу.

Застосунок базується на технології PHP. Основним серед недоліків є мала швидкодія. Не має можливості створювати аукціони.

Spotify розповсюджене серед середнього та малого бізнесу; SpotifyPlus розповсюджене серед крупного бізнесу. Цей застосунок базується на технології NodeJS.

Spotify є гнучким рішенням з високою швидкістю [7]. Також є можливість інтеграції зі сторонніми ІС за допомогою REST API.

Grandnode від nopCommerce розповсюджене серед середнього та малого бізнесу. Базується на технології Asp.Net Core та є кросплатформним. Відрізняється швидкою бистродією[15].

KeystoneJS розповсюджене серед малого бізнесу. Відрізняється легковажністю та швидкою розверткою. Також з'являється Open Source. Платформа базується на технології NodeJS. Keystone дає можливість будувати швидший та більш масштабований застосунок, ніж будь-яка інша CMS або фреймворк для застосунків. Достатньо тільки описати свою схему та отримати потужний API GraphQL та красивий інтерфейс керування вмістом і даними. Розробники фреймворка KeystoneJS позиціонують його як «The superpowered CMS for developers» [8] (Надпотужна CMS для розробників) «Жодного шаблону чи завантаження – лише елегантні API, які допоможуть вам передати важливий код, не жертвуючи гнучкістю чи потужністю створеного на замовлення сервера».

Порівняльна оцінка за результатами аналізу приведена у таблиці 1.1.

Таблиця 1.1 – Порівняльна оцінка за результатами аналізу.

Критерії	Magento	WordPress та WooCommerce	Snopify	Grandnode	Keystone JS
технологія та СУБД	PHP	PHP	NodeJS	ASP.NET Core	NodeJS
рівень бізнесу	Крупний	малий, середній	малий, середній, крупний	малий, середній	малий, середній
ліцензія продукту	Community, Enterprise	GNU GPL	комерційна	GNU, яка є Open Source	MIT, яка є Open Source
особливості архітектури	много-шарова	плагін	RESTful API	документна БД MongoDB	RESTful API
розповсюдженість	висока	Висока	Висока	Мала	середня
масштабованість	висока	Середня	Висока	Висока	висока
швидкодія	висока	Мала	висока, асинхронна	Висока	висока, асинхронна
трудомісткість процесу розробки	висока	Висока	Висока	Висока	низка

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

13

1.3. Формалізація задачі та постановка часткових задач досліджень.

За результатами аналізу сучасного стану процесів електронної комерції та аналізу відомих рішень щодо реалізації процесів електронної комерції прийнято рішення використовувати у розробці проєктного застосунку CMS KeystoneJS тому що

- 1) CMS KeystoneJS є основаним на Open Source
- 2) KeystoneJS ідеально підходить для малого бізнесу та стартапів
- 3) Швидко розробляється, швидко розгортається та швидко функціонує
- 4) Має легковажність та масштабованість
- 5) Можлива інтеграція з іншими ІС за допомогою GraphQL API

Задача з розробки застосунку для реалізації процесів електронної комерції може бути розбита на такі часткові задачі:

1. Розробка архітектури програмного застосунку для реалізації процесів електронної комерції.

1.1. Визначення вимог до програмного застосунку.

1.2. Визначення структурних складових застосунку.

1.3. Визначення діючих лиць (акторів).

1.4. Визначення сценаріїв, згідно яких діють актори

1.5. Визначення сутностей об'єктів, які зустрічаються в електронній комерції.

2. Програмна розробка застосунку для реалізації процесів електронної комерції.

2.1. Розробка схеми моделі даних виходячи з сутностей об'єктів та вимог.

2.2. Розробка точок GraphQL API для управління об'єктами.

2.3. Розробка мобільного застосунку.

2.4. Тестування застосунку.

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

14

Висновки за 1 розділом.

Для визначення найкращої технології для подальшої розробки застосунку для реалізації процесів електронної комерції було проаналізовано найбільш розповсюджені технології на сьогоднішній день. Аналіз показує, що для програмного застосунку для реалізації електронної комерції найкраще підходить фреймворк KeystoneJS на основі технології NodeJS. По-перше, фреймворк KeystoneJS є проектом Open Source. По-друге проект орієнтований на малий бізнес та стартапи. По-третє, є простим в процесі розробки. По-четверте, є можливість інтеграції з іншими ІС за допомогою GraphQL API.

Основна задача була розбита на часткові задачі для подальшого рішення.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

РОЗДІЛ 2.

АРХІТЕКТУРА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ДЛЯ РЕАЛІЗАЦІЇ ПРОЦЕСІВ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

2.1 Розробка архітектури програмного застосунку для реалізації процесів електронної комерції

Найкращі у світі серверні програми електронної комерції розроблені для високої масштабованості, оскільки вони завжди відповідають за створення, підтримку та доставку інформації для frontend користувача [9, 30].

Однак серверна частина програми електронної комерції може бути як однорідною, так і неоднорідною і складатися з 2 частин: бази даних (де зберігаються дані або інформація) і сервера (який з'єднує дані з браузером)[25,26].

2.1.1. Інженерія вимог.

Вимоги до програмного застосунку з реалізації процесів електронної комерції містять:

- конфіденційність, безпека і захист даних;
- відмовостійкість;
- асинхронність доступу користувачів до системи;
- висока продуктивність при звертанні до системи.

Відповідно до вимог з безпеки системи, в системі потрібно виділити рівні доступу користувачів, що мають право доступу до певних функцій або даних, додати способи перевірки доступу [18,19].

Система повинна виконувати такі функції:

- реєструвати користувача з адміністративними правами під час першого запуску програми;
- редагувати користувача з адміністративними правами;
- реєструвати користувача з правами продавця;
- редагувати користувача з правами продавця;

					ІАЛЦ.467200.003 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

- видаляти користувача з правами продавця;
- реєструвати користувача з правами покупця;
- редагувати користувача з правами покупця;
- видаляти користувача з правами покупця;
- додавати типи товарів з ієрархічною структурою;
- редагувати типи товарів;
- редагувати рівні ієрархії типів товарів;
- видаляти типи товарів;
- додавати картки товарів із зазначенням типу товару;
- редагувати картки товарів;
- видаляти картки товарів;
- додавати зображення до картки товарів;
- редагувати зображення до картки товару;
- видаляти зображення до картки товару;
- створювати знижку для товару;
- редагувати знижку для товару;
- видаляти знижку для товару;
- застосовувати знижку для товару до замовлення;
- створювати користувача при реєстрації через браузер або мобільний додаток;
- відображати список товарів, їх кількість та ціну;
- при переході до картки товару система повинна відображати детальну інформацію про товар та всі фотографії та знижки;
- вибирати товар та додавати до кошика;
- прибирати товар із кошика;
- змінювати кількість обраного товару;
- відображати сумарну інформацію та сумарну вартість на замовлення;
- покупець повинен уміти переглядати товари;
- покупець повинен уміти додавати товари до замовлення;
- покупець повинен вміти скасовувати замовлення;

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

17

- видавати помилку у разі спроби перевищення рівня доступу;
- покупець повинен уміти оформляти замовлення із зазначенням адреси доставки;
- покупець повинен вміти здійснювати оплату;
- менеджер повинен переглядати замовлення;
- менеджер повинен виконувати замовлення, при цьому статус замовлення повинен змінюватися на виконання;
- ключові поля в картках товару, замовлення, користувача, знижки повинні бути обов'язково заповнені;
- у разі відсутності заповнення ключових полів користувачам має видаватися повідомлення про помилку;
- система повинна бути асинхронною та витримувати підвищені навантаження.

2.1.2. Архітектурне проектування програмного застосунку.

Процес архітектурного проектування програмного застосунку містить:

- 1) Визначення структурних складових застосунку. Це перш за все інфраструктура та додаткове програмне середовище. Далі потрібно визначити складових компонентів-модулей.
- 2) Визначення діючих лиць (акторів), до яких можна віднести користувачів з різним рівням доступу (адміністратора, менеджерів-продавців та покупців);
- 3) Визначення сценаріїв, згідно яких діють актори. В тому числі сценарій першого запуску та реєстрації в системі;
- 4) Визначення сутностей об'єктів, які зустрічаються в електронній комерції;
- 5) Розробка схеми моделі даних виходячи з сутностей об'єктів та вимог;
- 6) Розробка точок GraphQL API для управління об'єктами;
- 7) Розробка мобільного застосунку;

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

У CMS KeystoneJS передбачені функції, які зображено на рисунку 2.1

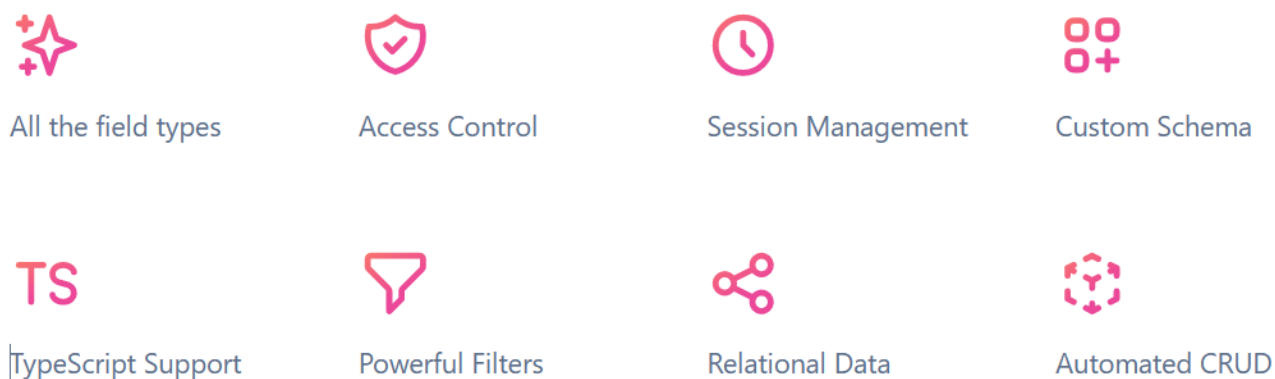


Рисунок 2.1. Функції CMS KeystoneJS.

Функції, які реалізовано в CMS KeystoneJS:

- контроль доступу;
- управління сесіями;
- автоматичний CRUD;
- зв'язані дані;
- вільна схема моделі користувача;
- широкі можливості фільтрів;
- підтримка всіх типів даних та форматування цих даних;
- можливість простої інтеграції з frontend - компонентом.

Система буде побудована за трирівневим підходом. Перший – це рівень презентації, орієнтований на користувача, другий – бізнес (або логічний) рівень, третій – рівень даних [10].

Презентаційний шар містить інтерфейсну частину програми, яка повинна надати користувачам можливість переглядати каталог продуктів, можливість фільтрувати їх за категоріями, шукати їх, додавати їх у свій кошик і завершати замовлення з оплатою та визначенням доставки.

Крім того, цей рівень повинен дозволяти користувачеві переглядати статус та історію замовлень.

Логічний шар – це наступний рівень, котрий містить усі внутрішні

процеси системи: обробку продуктів, їх запаси, обробку замовлень, платежі та обробку та планування доставки.

Крім того, вирішальне значення у цьому випадку мають авторизація та аутентифікація користувачів за допомогою контролю доступу системи. В основному, мета логічного рівня полягає в обробці даних з презентаційного рівня або рівня даних та передача даних далі в обох напрямках.

Рівень даних - це останній рівень, який відповідає за зберігання даних. По суті, це база даних нашої системи електронної комерції. Мета рівня зберігання даних — підтримувати доступність і узгодженість даних, а також забезпечити дотримання принципів атомарності, узгодженості, ізольованості, довговічності, що гарантують надійну роботу транзакцій бази даних.

Систему електронної комерції, що створюється, повинна містити усі три шари.

Розглянемо елементи системи, що створюють кожен окремий шар, а також зв'язки між рівнями.

Елементи презентаційного шару. Рівень інтерфейсу буде складатися з трьох основних елементів: основного додатка, платформи Next.JS зі службою CDN та веб-сервера Nginx.

Це досить просте рішення: головна програма, вбудована в React має всі переваги SEO та швидші початкові завантаження з сервера.

Роль CDN тут полягає в тому, щоб допомогти з обслуговуванням статичних файлів, переважно зображень. Nginx працює як зворотний проксі-сервер і балансувальник навантаження. Весь зв'язок з логічним рівнем відбувається через GraphQL.

Елементи логічного шару містять лише два елементи: CMS Keystone.js, що працює як основний процесор, а також командний блок GraphQL, які діють як розширення і працюють як обгортка для можливостей текстового пошуку для базового рівня.

Архітектура програмного застосунку приведена на рис. 2.2.

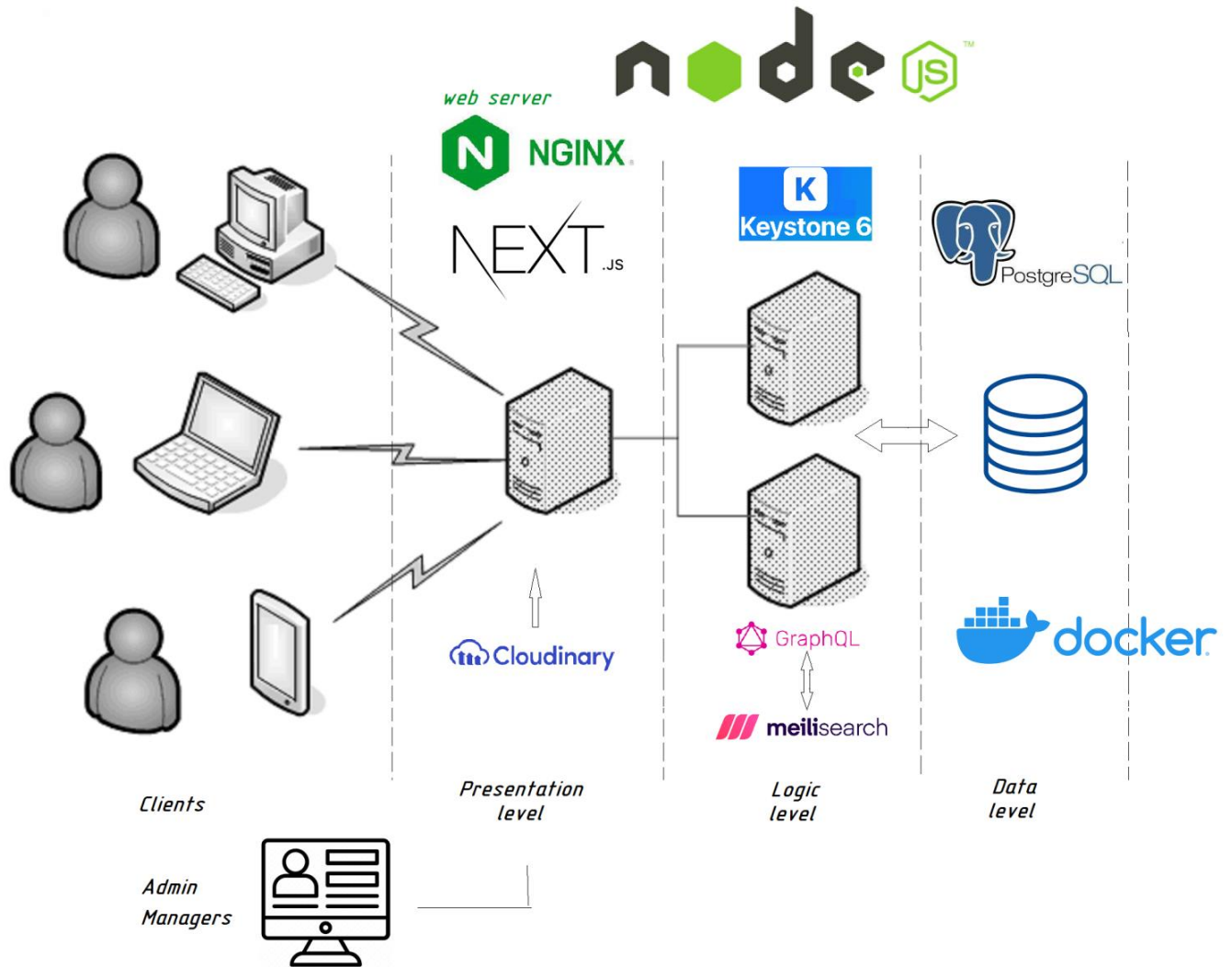


Рисунок 2.2. Загальна архітектура програмного застосунку.

Розглянемо архітектуру більш детально.

Keystone керує трьома основними сутностями для всієї системи: User (користувач), Item (продукт) і Booking (замовлення).

Решта сутностей можуть бути прямо чи опосередковано похідними від цих базових. По-перше, Користувач зберігає основну інформацію про клієнтів і персонал нашої системи електронної комерції, в основному інформацію про авторизацію та призначені ролі.

Також Користувач має відношення один до однієї сутності кошика і може мати кілька сутностей замовлень і кілька сутностей адреси. Кожна сутність замовлення має один підключений платіж і відправлення та містить кілька сутностей продукту.

Змн.	Арк.	№ докум.	Підпис	Дата

Продукт має відношення з однією сутністю PriceList (каталог чи прайс-лист). Для простоти ми припускаємо, що не будемо використовувати настройку для кількох складів. Також Продукт має кілька сутностей ItemImage (Зображень продукта).

Також продукт має зв'язок з кількома сутностями ItemType (Тип чи Категорія), і кожна з них може бути пов'язана з батьківським деревом, що формує ієрархію категорій.

Структура базових сутностей зображена на рис. 2.3.

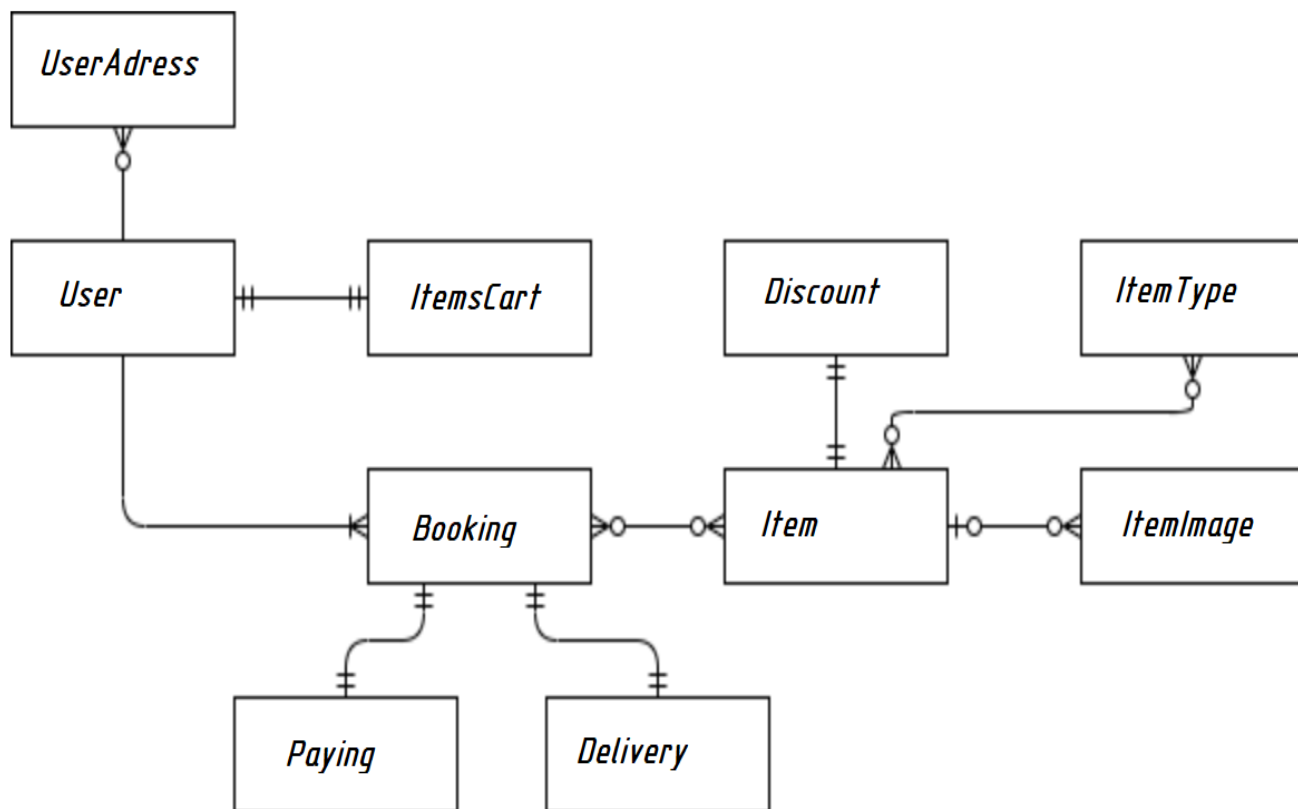


Рисунок 2.3. Структура базових сутностей

У межах структури базових сутностей, роль Keystone полягає в управлінні вмістом базових сторінок для основної інтерфейсної програми. Але важливіше те, що ця роль також полягає в організації планувальника періодичних процесів та зовнішніми інтеграціями за допомогою API.

Прикладом планувальника можуть бути повідомлення електронною поштою, пов'язані з етапами обробки замовлень та інформуванням клієнтів про це або індексацією даних про товари в пошукових системах після змін.

Приклади зовнішніх інтеграцій містять підключення до зовнішньої доставки обробки API (наприклад, API компанії Нова пошта).

З метою розробки програмного застосунку для реалізації процесів електронної комерції необхідно організувати інфраструктуру та додаткове програмне середовище, в якому буде функціонувати застосунок електронної комерції. Отже з початку встановлюється необхідна програмна інфраструктура. До неї відноситься:

1) Node.JS. Інструментарій може бути встановлений на будь-які платформи: Unix-подібні ОС, MacOS чи Windows ОС. В даному проєкті використовується Node.JS v16.14.2 для Windows x64. Встановлюється стандартно (дистрибутив завантажується з офіційного сайту розробника. Після запуску потрібно ознайомитися з ліцензією, вибрати місце розташування та інсталювати пакет).

2) Docker. Це програмне забезпечення дозволяє використовувати віртуальні контейнери з додатковими функціями. Даний проєкт має основний контейнер з образом СУБД проєкту (Postgres 14), образом інструментарію пошуку у backend-частини проєкту та інструментарію адміністрування бази даних. Завдяки Docker маємо ізольованість програми на рівні процесу та безпечну роботу з небезпечним кодом (технологія ізоляції контейнерів дозволяє запускати будь-який код без шкоди для ОС); зручне приховування процесів (для кожного контейнера можна використовувати різні методи обробки даних, приховуючи процеси фону); швидкісне розгортання (допоміжні компоненти можна не встановлювати, а використовувати вже готові образи); просте масштабування (будь-який проєкт можна розширити, впровадивши нові контейнери); зручний запуск (програму, що знаходиться всередині контейнера, можна запустити на будь-якому docker-хості); оптимізація файлової системи (образ складається з шарів, які дозволяють ефективно використовувати файлову систему) [11]. В даному проєкті використовується Docker Desktop 4.8.2 для Windows (рис. 2.4).

Змн.	Арк.	№ докум.	Підпис	Дата

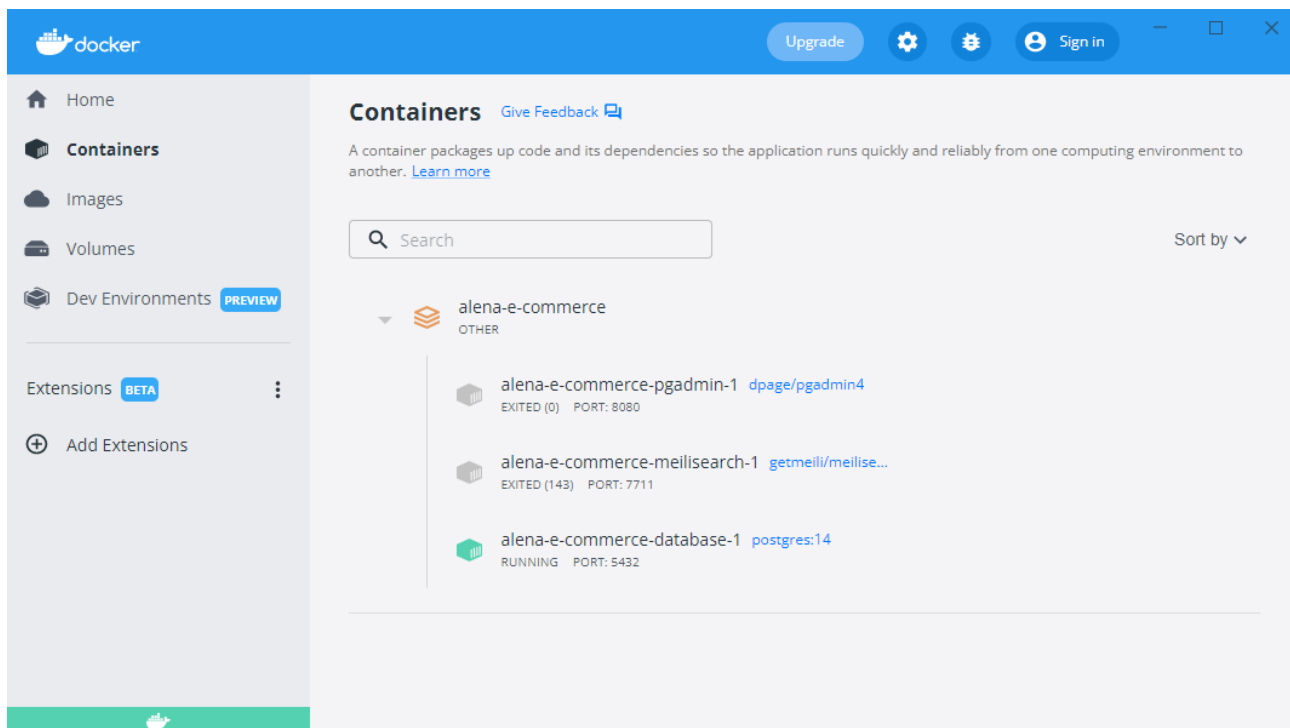


Рисунок 2.4. Контейнер Docker як складова інфраструктури проєкту.

3) Yarn. Це менеджер пакетів, який одночасно виконує роль менеджера проєкту. Завдяки Yarn легко встановлювати необхідні компоненти інфраструктури та створювати інсталяцію всього проєкту.

4) Репозиторій застосунку що створено на платформі GitHub (<https://github.com/drA666/alena-e-commerce>).

5) Keystone інструментарій –CMS для background компоненти

6) Next React інструментарій для frontend компоненти

7) PostgreSQL СУБД для бази даних проєкту

8) CDN Cloudinary – хмарний сервіс, який дозволяє зберігати, обробляти, передавати до користувача статичні зображення. В проєкті цей сервіс грає роль додаткового хостінгу для зображень

2.2. Програмна розробка програмного застосунку для реалізації процесів електронної комерції.

Програмний застосунок для реалізації процесів електронної комерції реалізовано на мові програмування typescript та javascript.

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

24

З метою розробки застосунку насамперед встановлено додаткове програмне середовище, а саме: Node.JS for Windows, Docker Desktop for Windows, Keystone-6 + Cloudinary, Next React, PostgreSQL, створено обліковий запис CDN Cloudinary

Далі описуються моделі базових сутностей на мові TypeScript.

Опис моделі заказа приведено на рис. 2.5

```
booking.schema.ts
1  import { list } from '@keystone-6/core';
2  import { json, relationship, select, timestamp } from '@keystone-6/core/fields';
3  import { BookingStatusOptions } from '../consts/booking-status-options.const';
4  import { filterCustomerAccess, filterCustomerAccessCreate } from '../shared';
5
6  export const Booking = list({
7    fields: {
8      user: relationship({ ref: 'User' }),
9      items: json(),
10     payment: relationship({ ref: 'Paying' }),
11     delivery: relationship({ ref: 'Delivery' }),
12     employee: relationship({ ref: 'User' }),
13     status: select({ type: 'enum', options: BookingStatusOptions }),
14     createdAt: timestamp({
15       defaultValue: { kind: 'now' },
16     }),
17     lastModification: timestamp({
18       defaultValue: { kind: 'now' },
19     },
20     db: {
21       updatedAt: true,
22     },
23   },
24   access: {
25     operation: {
26       query: ({ session }) => !!session,
27       create: ({ session }) => !!session,
28       update: ({ session }) => !!session,
29       delete: ({ session }) => !!session,
30     },
31     filter: {
32       query: ({ session }) => filterCustomerAccess(session),
33       update: ({ session }) => filterCustomerAccess(session),
34       delete: ({ session }) => filterCustomerAccess(session),
35     },
36     item: {
37       create: ({ session, inputData }) =>
38         filterCustomerAccessCreate(session, inputData),
39     },
40   },
41 });
```

Рисунок 2.5. Опис моделі заказа

В моделі можемо побачити, що сутність заказа зв'язано з такими сутностями, як платіж, доставка, менеджер, та безпосередньо з користувачем, який створив заказ. При цьому, сутність менеджера та покупця є по суці однаковими сутностями – сутністю користувача.

Модель сутності користувача зображена на рис. 2.6.

```
user.schema.ts x
1 import { list } from '@keystone-6/core';
2 import { text, password, select, relationship } from '@keystone-6/core/fields';
3 import { UsersRolesValues } from '../consts/usersroles.const';
4 import { UsersRoles } from '../enums/usersroles.enum';
5 import { filterCustomerAccess, filterCustomerAccessCreate } from '../shared';
6
7 export const User = list({
8   fields: {
9     name: text({ validation: { isRequired: true } }),
10    email: text({
11      validation: { isRequired: true },
12      isIndexed: 'unique',
13      isFilterable: true,
14    }),
15    password: password({ validation: { isRequired: true } }),
16    role: select({
17      type: 'enum',
18      options: UsersRolesValues,
19      defaultValue: UsersRoles.Customer,
20    }),
21    useraddress: relationship({ ref: 'UserAddress', many: true }),
22  },
23  access: {
24    operation: {
25      query: ({ session }) => !!session,
26      update: ({ session }) => !!session,
27      delete: ({ session }) => !!session,
28    },
29    filter: {
30      query: ({ session }) => filterCustomerAccess(session),
31      update: ({ session }) => filterCustomerAccess(session),
32      delete: ({ session }) => filterCustomerAccess(session),
33    },
34    item: {
35      update: ({ session, inputData }) =>
36        filterCustomerAccessCreate(session, inputData),
37    },
38  },
39  hooks: {
40    afterOperation: async ({ operation, item, context }) => {
41      if (operation !== 'create') {
42        return;
43      }
44      const userId = item?.id;
45      await context.prisma.itemscart.create({
46        data: {
47          user: { connect: { id: userId } },
48        },
49      });
50    },
51  },
52 });
```

Рисунок 2.6. Опис моделі користувача

В моделі можемо побачити, що сутність заводу зв'язано з сутністю адреса у відношенні один користувач – багато адрес.

Змн.	Арк.	№ докум.	Підпис	Дата

Модель сутності адреси можемо побачити на рис. 2.7.

```
useraddress.schema.ts
1  import { list } from '@keystone-6/core';
2  import { relationship, text, timestamp } from '@keystone-6/core/fields';
3  import { filterCustomerAccess, filterCustomerAccessCreate } from '../shared';
4
5  export const UserAddress = list({
6    fields: {
7      addressName: text(),
8      name: text({ validation: { isRequired: true } }),
9      streetAddress: text({ validation: { isRequired: true } }),
10     streetAddress2: text(),
11     city: text({ validation: { isRequired: true } }),
12     postCode: text({ validation: { isRequired: true } }),
13     country: text({ validation: { isRequired: true } }),
14     telNo: text({ validation: { isRequired: true } }),
15     user: relationship({
16       ref: 'User',
17       ui: {
18         hideCreate: true,
19       },
20     }),
21     createdAt: timestamp({
22       defaultValue: { kind: 'now' },
23     }),
24     lastModification: timestamp({
25       defaultValue: { kind: 'now' },
26       db: {
27         updatedAt: true,
28       },
29     }),
30   },
31   access: {
32     operation: {
33       query: ({ session }) => !!session,
34       create: ({ session }) => !!session,
35       update: ({ session }) => !!session,
36       delete: ({ session }) => !!session,
37     },
38     filter: {
39       query: ({ session }) => filterCustomerAccess(session),
40       update: ({ session }) => filterCustomerAccess(session),
41       delete: ({ session }) => filterCustomerAccess(session),
42     },
43     item: {
44       create: ({ session, inputData }) =>
45         filterCustomerAccessCreate(session, inputData),
46     },
47   },
48 });
```

length: 1517 lines: 48 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS

Рисунок 2.7. Опис моделі адреси.

Однією з найважливішою сутністю є сутність продукту (Item). Сутність продукту має опис, тип, може мати одне чи кілька зображень та декілька скидок. Сутність типу продукту має ієрархічний вигляд. Тобто кілька продуктів можуть мати загальний тип. Цей загальний тип може мати кілька детальних дочірніх типів і таке інше.

Модель сутності продукту зображена на рис. 2.8.

```
item.schemas.ts
1 | import { list } from '@keystone-6/core';
2 | import {
3 |   decimal,
4 |   integer,
5 |   relationship,
6 |   text,
7 |   timestamp,
8 | } from '@keystone-6/core/fields';
9 | import { document } from '@keystone-6/fields-document';
10 | import { UsersRoles } from '../enums/usersroles.enum';
11 |
12 | export const Item = list({
13 |   fields: {
14 |     name: text({ validation: { isRequired: true } }),
15 |     description: document(),
16 |     seoDesc: text({ ui: { displayMode: 'textarea' } }),
17 |     itemTypes: relationship({ ref: 'ItemType', many: true }),
18 |     itemImages: relationship({ ref: 'ItemImage', many: true }),
19 |     price: decimal(),
20 |     stars: integer(),
21 |     discount: relationship({ ref: 'Discount' }),
22 |     createdAt: timestamp({
23 |       defaultValue: { kind: 'now' },
24 |     }),
25 |     lastModification: timestamp({
26 |       defaultValue: { kind: 'now' },
27 |     }, {
28 |       db: {
29 |         updatedAt: true,
30 |       },
31 |     }
32 |   },
33 | });
```

Рисунок 2.8. Опис моделі продукту.

Опис сутності зображення продукту, яка зв'язана з сутністю продукту зображено на рис. 2.9

```
item-image.schemas.ts
1 | import { list } from '@keystone-6/core';
2 | import { text, timestamp } from '@keystone-6/core/fields';
3 | import { cloudinaryImage } from '@keystone-6/cloudinary';
4 | import { UsersRoles } from '../enums/usersroles.enum';
5 |
6 | export const ItemImage = list({
7 |   fields: {
8 |     alt: text({ validation: { isRequired: true } }),
9 |     image: cloudinaryImage({
10 |       cloudinary: {
11 |         cloudName: process.env.CLOUDINARY_CLOUD_NAME ?? 'alenarasuk',
12 |         apiKey: process.env.CLOUDINARY_API_KEY ?? '639437786523791',
13 |         apiSecret: process.env.CLOUDINARY_API_SECRET ?? 'DHtx8_I7tlyASuCn0nfiWhuMons',
14 |         folder: process.env.CLOUDINARY_API_FOLDER ?? '',
15 |       },
16 |     }),
17 |     createdAt: timestamp({
18 |       defaultValue: { kind: 'now' },
19 |     }),
20 |     lastModification: timestamp({
21 |       defaultValue: { kind: 'now' },
22 |     }, {
23 |       db: {
24 |         updatedAt: true,
25 |       },
26 |     }
27 |   },
28 | });
```

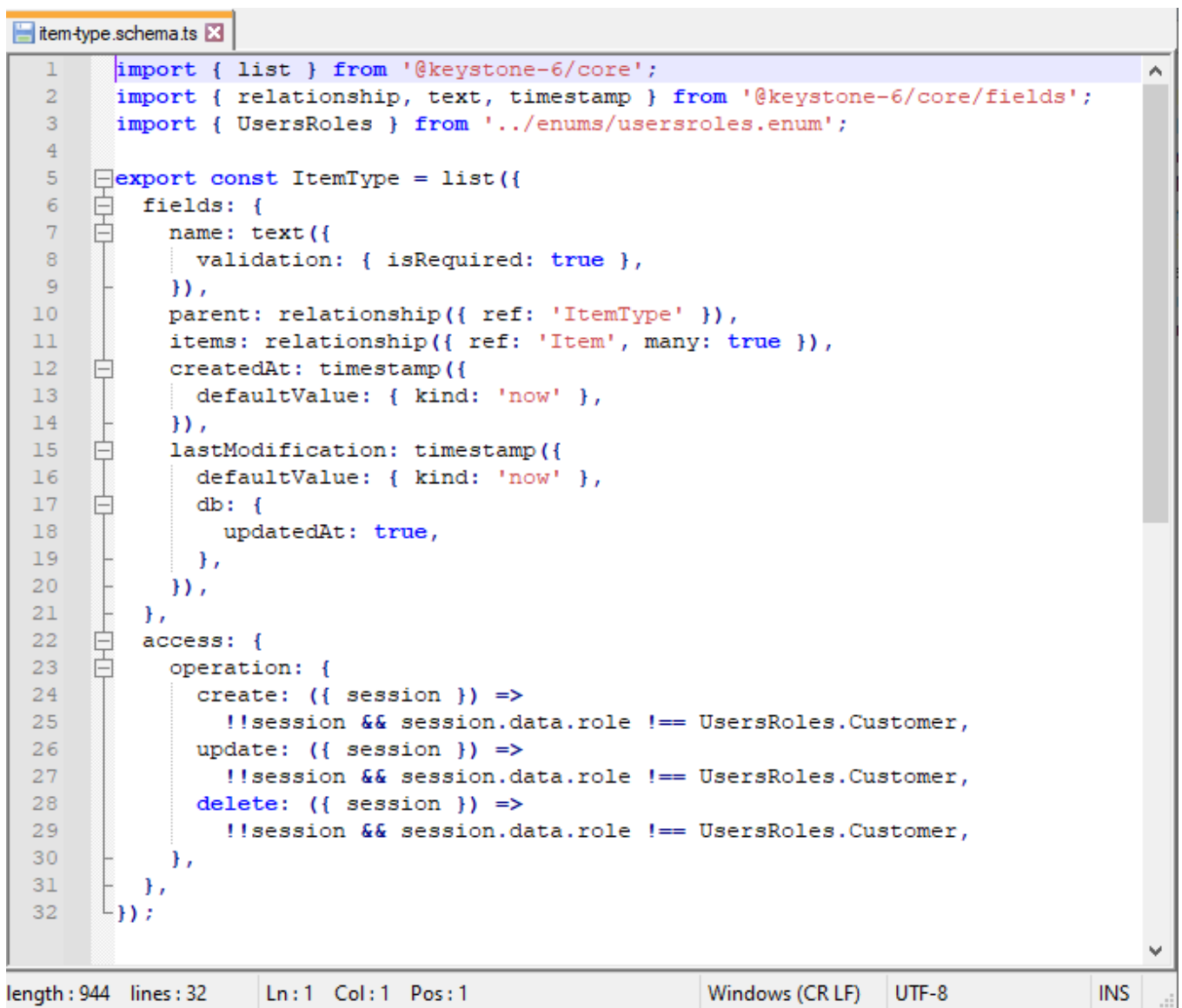
Рисунок 2.9. Опис моделі зображення продукту.

Відзначимо, що зображення для продукту використовують службу CDN Cloudinary, що дозволяє зберігати та кешувати зображення, таким чином зменшити трафік та зменшити витрачання дискового простору сервера. Для функціонування Cloudinary необхідно створити обліковий запис та підключити API до проєкту (рис. 2.10).

Для підключення API вказується

- назва облікового запису;
- ключ API;
- секретний ключ API.

Опис сутності типу продукту зображено на рис. 13.



```
item-type.schema.ts
1 import { list } from '@keystone-6/core';
2 import { relationship, text, timestamp } from '@keystone-6/core/fields';
3 import { UsersRoles } from '../enums/usersroles.enum';
4
5 export const ItemType = list({
6   fields: {
7     name: text({
8       validation: { isRequired: true },
9     }),
10    parent: relationship({ ref: 'ItemType' }),
11    items: relationship({ ref: 'Item', many: true }),
12    createdAt: timestamp({
13      defaultValue: { kind: 'now' },
14    }),
15    lastModification: timestamp({
16      defaultValue: { kind: 'now' },
17    },
18    db: {
19      updatedAt: true,
20    },
21  }),
22  access: {
23    operation: {
24      create: ({ session }) =>
25        !!session && session.data.role !== UsersRoles.Customer,
26      update: ({ session }) =>
27        !!session && session.data.role !== UsersRoles.Customer,
28      delete: ({ session }) =>
29        !!session && session.data.role !== UsersRoles.Customer,
30    },
31  },
32 });
```

length: 944 lines: 32 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS

Рисунок 2.10. Опис моделі типу продукту.

Опис моделі кошику приведено на рис. 2.11.

```
items-cart.schemas.ts
1  import { list } from '@keystone-6/core';
2  import { decimal, relationship, timestamp } from '@keystone-6/core/fields';
3  import { filterCustomerAccess, filterCustomerAccessCreate } from '../shared';
4
5  export const ItemsCart = list({
6    fields: {
7      user: relationship({
8        ref: 'User',
9        ui: {
10         hideCreate: true,
11       },
12     }),
13     items: relationship({
14       ref: 'CartItem',
15       many: true,
16       db: {
17         foreignKey: true,
18       },
19     }),
20     lastModified: timestamp({
21       defaultValue: { kind: 'now' },
22       db: {
23         updatedAt: true,
24       },
25     }),
26     sum: decimal({ validation: { isRequired: true }, defaultValue: '0' }),
27   },
28   access: {
29     operation: {
30       query: ({ session }) => !!session,
31       update: ({ session }) => !!session,
32       delete: ({ session }) => !!session,
33     },
34     filter: {
35       query: ({ session }) => filterCustomerAccess(session),
36       update: ({ session }) => filterCustomerAccess(session),
37       delete: ({ session }) => filterCustomerAccess(session),
38     },
39   },
40   graphql: {
41     omit: ['create'],
42   },
43   hooks: {
44     resolveInput: async ({ resolvedData, context }) => {
45       if (resolvedData?.items?.connect?.length) {
46         const items = await context.query.ItemsCartItem.findMany({
47           where: {
48             id: {
49               in: resolvedData?.items?.connect?.map(
50                 (el: { id: string }) => el.id
51               ),
52             },
53           },
54           query: 'amount item { price }',
55         });
56         const sum = items.reduce((prev: number, current) => {
57           const currentSum = current.item.price * current.amount;
58           return prev + currentSum;
59         }, 0);
60         resolvedData.sum = sum;
61       }
62       return resolvedData;
63     },
64     beforeOperation: async ({ resolvedData, item, context }) => {
65       if (resolvedData?.items) {
66         const dbItems = await context.query.ItemsCart.findOne({
67           where: { id: item ? (item.id as string) : '' },
68           query: 'items { id }',
69         });
70         if (dbItems.items.length) {
71           await context.query.ItemsCartItem.deleteMany({
72             where: dbItems.items.map((el: { id: string }) => ({
73               id: el.id,
74             })),
75           });
76         }
77       }
78     },
79   },
80 });
```

Рисунок 2.11. Опис моделі кошику.

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.
30

Процеси користувача з кошиком:

Додавання в кошик:

- 1) перевірка запасів;
- 2) створення об'єктів CartItem;
- 3) оновити кошик

Видалити товар з кошика:

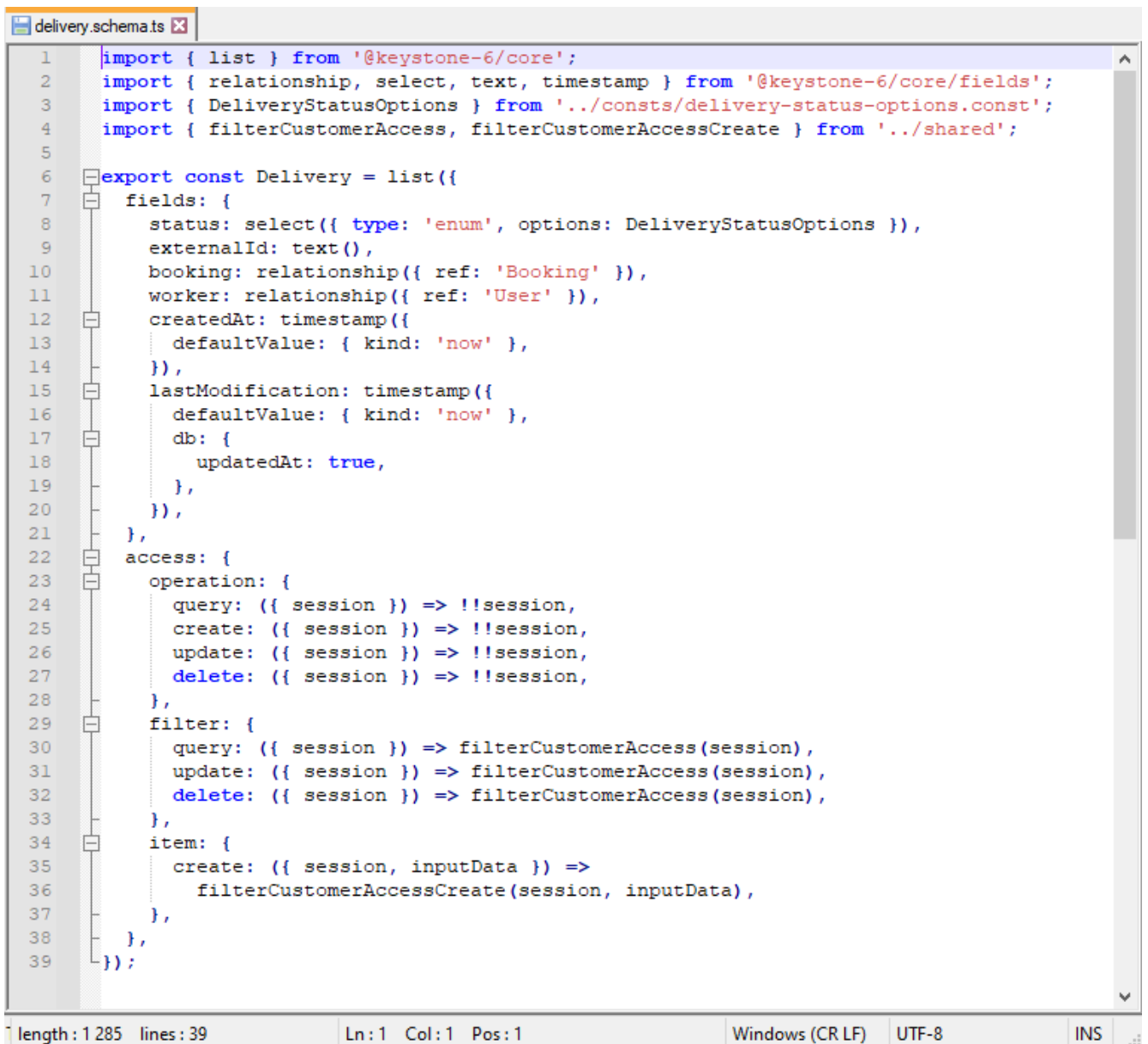
- 1) видалення об'єкта CartItem;
- 2) оновлення кошика;

Зміна кількості в кошику:

- 1) перевірка запасів;
- 2) оновлення сутності CartItem;
- 3) оновлення

кошика

Опис моделі доставки приведено на рис. 2.12.



```
1 import { list } from '@keystone-6/core';
2 import { relationship, select, text, timestamp } from '@keystone-6/core/fields';
3 import { DeliveryStatusOptions } from '../consts/delivery-status-options.const';
4 import { filterCustomerAccess, filterCustomerAccessCreate } from '../shared';
5
6 export const Delivery = list({
7   fields: {
8     status: select({ type: 'enum', options: DeliveryStatusOptions }),
9     externalId: text(),
10    booking: relationship({ ref: 'Booking' }),
11    worker: relationship({ ref: 'User' }),
12    createdAt: timestamp({
13      defaultValue: { kind: 'now' },
14    }),
15    lastModification: timestamp({
16      defaultValue: { kind: 'now' },
17      db: {
18        updatedAt: true,
19      },
20    }),
21  },
22  access: {
23    operation: {
24      query: ({ session }) => !!session,
25      create: ({ session }) => !!session,
26      update: ({ session }) => !!session,
27      delete: ({ session }) => !!session,
28    },
29    filter: {
30      query: ({ session }) => filterCustomerAccess(session),
31      update: ({ session }) => filterCustomerAccess(session),
32      delete: ({ session }) => filterCustomerAccess(session),
33    },
34    item: {
35      create: ({ session, inputData }) =>
36        filterCustomerAccessCreate(session, inputData),
37    },
38  },
39 });
```

length: 1 285 lines: 39 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS

Рисунок 2.12. Опис моделі доставки.

Опис моделі платежу приведено на рис. 2.13.

```
paying.schema.ts x
1  import { list } from '@keystone-6/core';
2  import {
3    decimal,
4    relationship,
5    select,
6    text,
7    timestamp,
8  } from '@keystone-6/core/fields';
9  import { PayingStatusOptions } from '../consts/paying-status-options.const';
10 import { filterCustomerAccess, filterCustomerAccessCreate } from '../shared';
11
12 export const Paying = list({
13   fields: {
14     booking: relationship({ ref: 'Booking' }),
15     sumAll: decimal(),
16     currency: text(),
17     externalId: text(),
18     status: select({ type: 'enum', options: PayingStatusOptions }),
19     createdAt: timestamp({
20       defaultValue: { kind: 'now' },
21     }),
22     lastModification: timestamp({
23       defaultValue: { kind: 'now' },
24       db: {
25         updatedAt: true,
26       },
27     }),
28   },
29   access: {
30     operation: {
31       query: ({ session }) => !!session,
32       create: ({ session }) => !!session,
33       update: ({ session }) => !!session,
34       delete: ({ session }) => !!session,
35     },
36     filter: {
37       query: ({ session }) => filterCustomerAccess(session),
38       update: ({ session }) => filterCustomerAccess(session),
39       delete: ({ session }) => filterCustomerAccess(session),
40     },
41     item: {
42       create: ({ session, inputData }) =>
43         filterCustomerAccessCreate(session, inputData),
44     },
45   },
46 });
```

Рисунок 2.13. Опис моделі платежу.

Сутність платежу зв'язана з сутністю замовлення. Також сутність платежу має такі поля як сума, валюта, статус платежу, дата створення, дата модифікації.

Остання модель - це модель знижки.

Змн.	Арк.	№ докум.	Підпис	Дата

Опис моделі знижки приведено на рис. 2.14.

```
discount.schema.ts x
1  import { list } from '@keystone-6/core';
2  import { integer, timestamp } from '@keystone-6/core/fields';
3  import { UsersRoles } from '../enums/usersroles.enum';
4
5  export const Discount = list({
6    fields: {
7      discount: integer(),
8      nextDelivery: timestamp(),
9      amountInNextDelivery: integer(),
10     createdAt: timestamp({
11       defaultValue: { kind: 'now' },
12     }),
13     lastModification: timestamp({
14       defaultValue: { kind: 'now' },
15       db: {
16         updatedAt: true,
17       },
18     }),
19   },
20   access: {
21     operation: {
22       create: ({ session }) =>
23         !!session && session.data.role !== UsersRoles.Customer,
24       update: ({ session }) =>
25         !!session && session.data.role !== UsersRoles.Customer,
26       delete: ({ session }) =>
27         !!session && session.data.role !== UsersRoles.Customer,
28     },
29   },
30 });

length: 857 lines: 30 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8
```

Рисунок 2.14. Опис моделі знижки.

Наступний крок – скомпонувати застосунок, налаштувати міграцію для спрощення подальшого встановлення.

Далі налаштовується обмеження доступу користувачів до різних сутностей на основі ролі користувача в системі. По суті, клієнти можуть змінювати лише дані, пов’язані з ними: деталізувати профіль користувача, редагувати та оформлювати замовлення, таке інше, але адміністратори мають набагато більше привілеїв. Все це має впливати на систему на трьох рівнях: спочатку обмежити доступ до інтерфейсу адміністратора, потім фільтрувати доступ на основі типу операції до кожної моделі сутностей і, нарешті, на рівні деяких полів у схемах.

Процес встановлення бази даних приведено на рис.2.15.

```
C:\Users\Ras\Program\alena-e-commerce\ecom-backend>yarn dev
yarn run v1.22.18
$ keystone dev
Starting Keystone
Dev Server Starting on http://localhost:5000
GraphQL API Starting on http://localhost:5000/api/graphql
Generating GraphQL and Prisma schemas
- Drift detected: Your database schema is not in sync with your migration history.

The following is a summary of the differences between the expected database schema given your migrations files, and the actual schema of the database.

It should be understood as the set of changes to get from the expected schema to the actual schema.

[+] Added enums
- BookingStatusType
- DeliveryStatusType
- PayingStatusType
- UserRoleType

[+] Added tables
- Booking
- Delivery
- Discount
- Item
- ItemImage
- ItemType
- ItemsCart
- Paying
- User
- UserAddress
- _ItemType_items
- _Item_itemImages
- _Item_itemTypes
- _ItemsCart_items
- _User_useraddress

[*] Changed the `Booking` table
[+] Added index on columns (delivery)
[+] Added index on columns (employee)
[+] Added index on columns (payment)
[+] Added index on columns (user)
[+] Added foreign key on columns (delivery)
[+] Added foreign key on columns (employee)
```

Рисунок 2.15. Процес встановлення бази даних

До проєкту згенеровано GraphQL API згідно схеми з моделями даним. Приклад роботи. GraphQL API забезпечує автоматичне додавання, редагування, читання, та видалення об'єктів сутностей у базі даних (функції CRUD).

Завдяки GraphQL API інтегруються між собою backend та frontend компоненти системи.

За допомогою GraphQL API до проєкту також можна підключити зовнішні системи, а саме системи бухгалтерського та складського обліку, автоматизувати заповнення контенту та актуалізувати дані по наявності і вартості товарів.

API можна побачити на рис. 2.16.

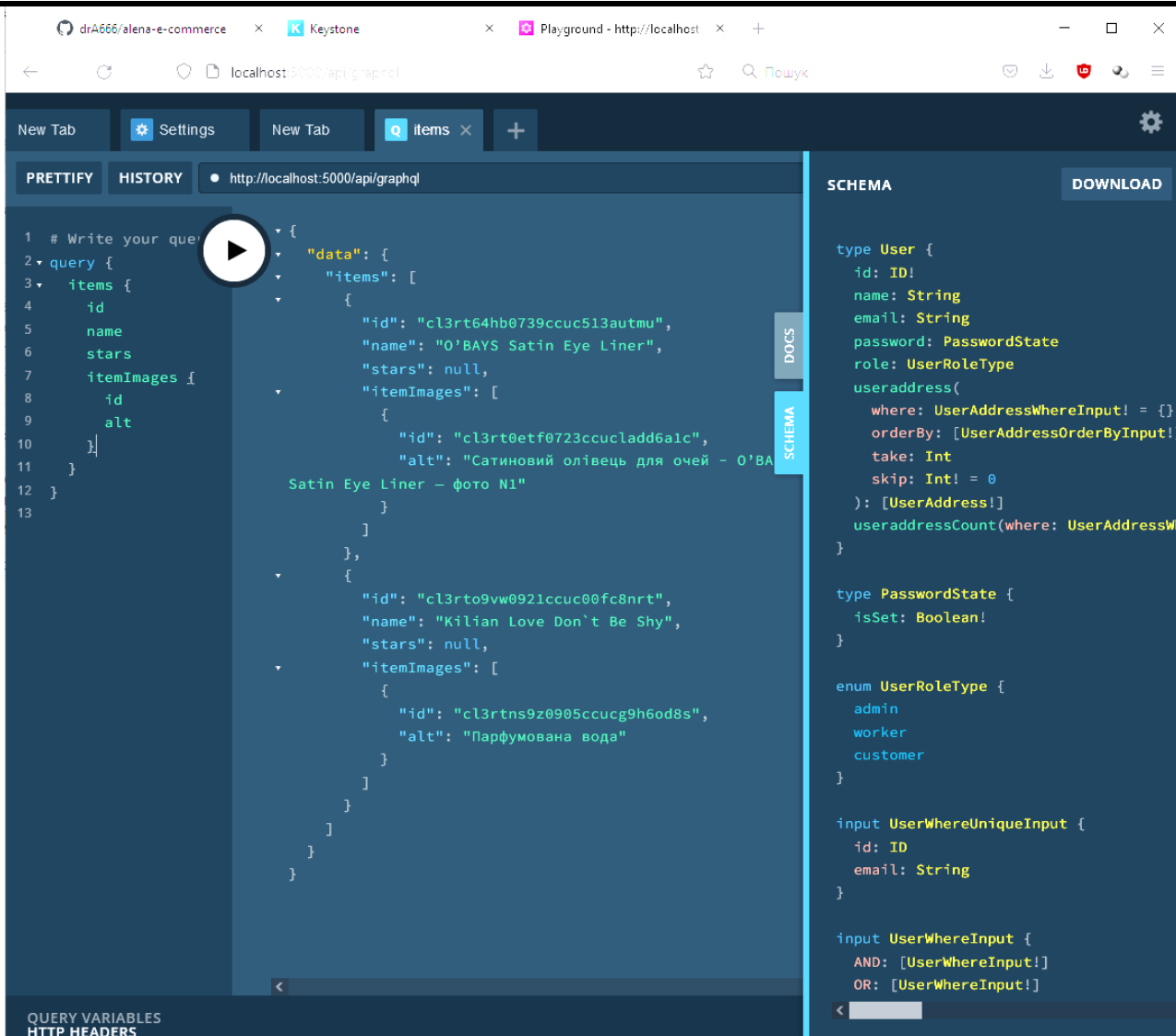


Рисунок 2.16. Приклад роботи GraphQL API

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

35

Висновки за 2 розділом.

В процесі роботи над проектом були сформульовані вимоги до програмного застосунка для процесів електронної комерції.

В ході роботи була спроектована архітектура застосунку:

- Визначено структурні складові застосунку;
- Визначено діючі лиця (актори);
- Визначено сценарії, згідно яких діють актори;
- Визначено сутності об'єктів, які зустрічаються в електронній комерції;
- Розроблені схеми моделі даних виходячи з сутностей об'єктів та вимог;
- Розроблені точки GraphQL API для управління об'єктами;
- Розроблено мобільний застосунок.

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

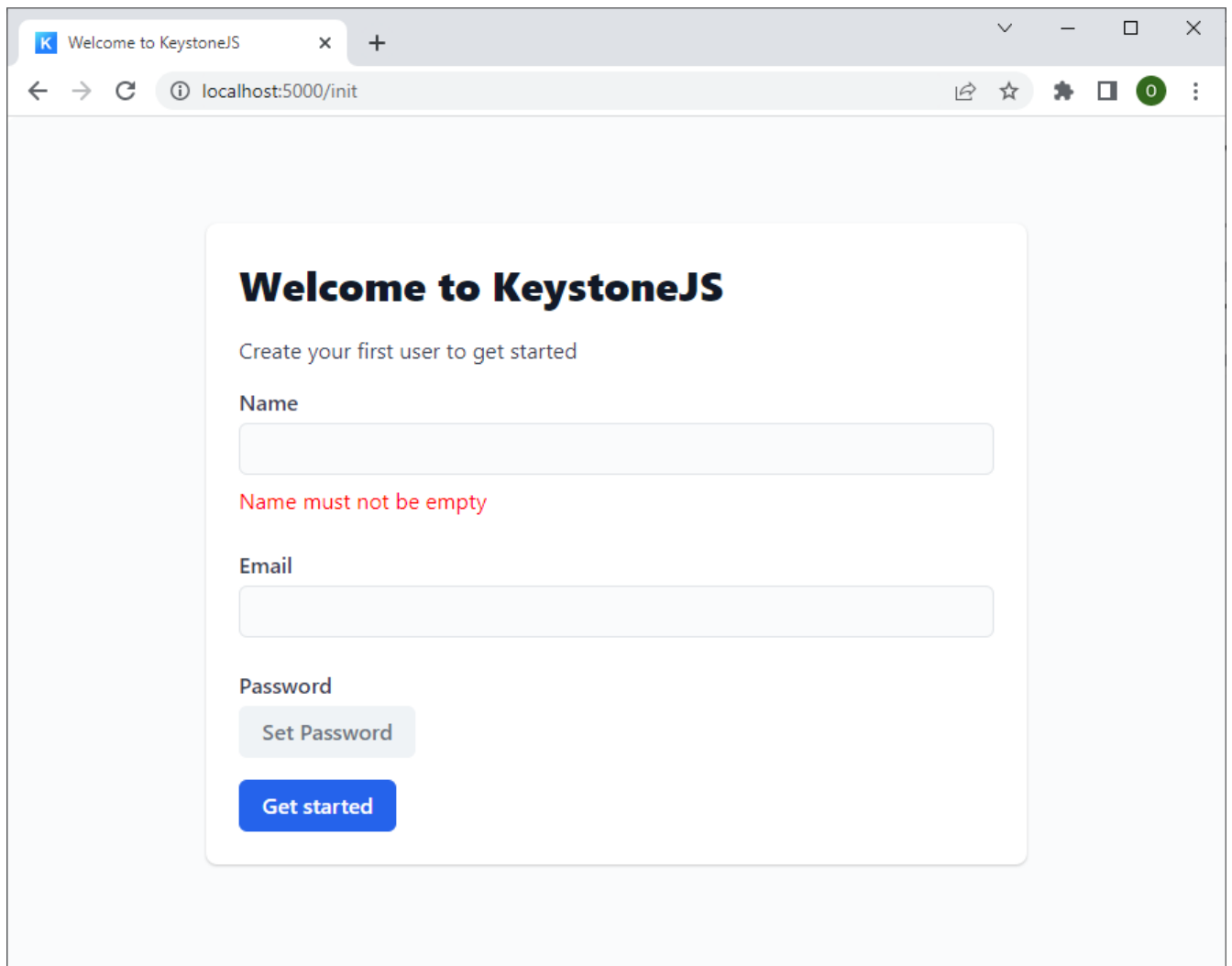
36

РОЗДІЛ 3.

ПРИКЛАДНІ АСПЕКТИ ЗАСТОСУВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ ДЛЯ РЕАЛІЗАЦІЇ ПРОЦЕСІВ ЕЛЕКТРОННОЇ КОМЕРЦІЇ.

3.1. Демонстрація практичних можливостей розробленого програмного застосунку для реалізації процесів електронної комерції.

При запуску початкової сторінки frontend необхідно створити першого користувача – адміністратора (рис. 3.1). Поля Ім'я, Поштова скринька та Пароль є обов'язковими до заповнення.



The screenshot shows a web browser window with the following elements:

- Browser tab: Welcome to KeystoneJS
- Address bar: localhost:5000/init
- Page title: Welcome to KeystoneJS
- Text: Create your first user to get started
- Form fields:
 - Name: (with error message: Name must not be empty)
 - Email:
 - Password: (with a 'Set Password' button)
- Buttons: 'Get started' (blue), 'Set Password' (grey)

Рисунок 3.1. Початкова сторінка backend

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

37

При вході адміністратора відкривається робочий стіл backend з усіма сутностями, які були створені раніше (рис. 3.2).

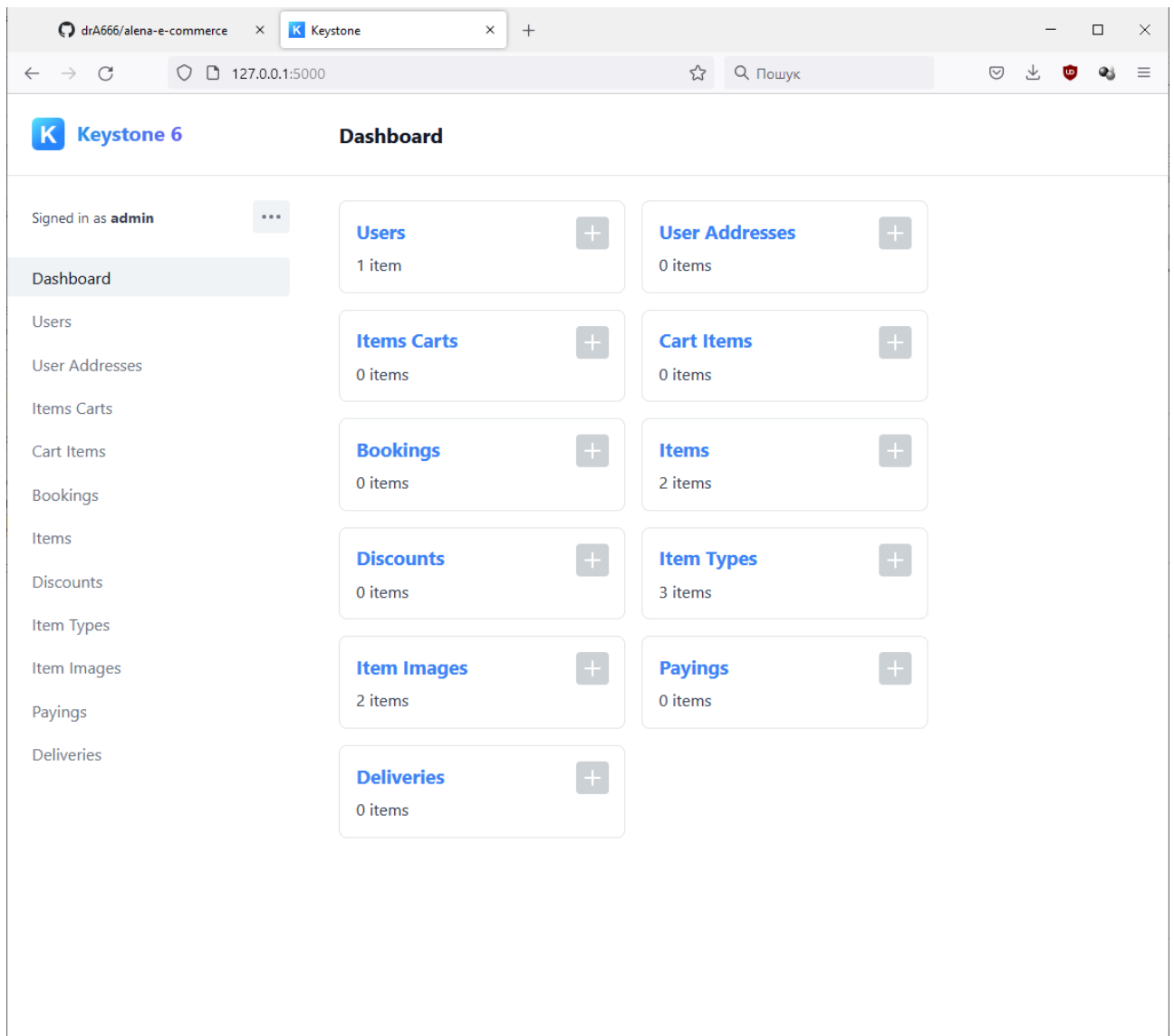


Рисунок 3.2. Робочий стіл backend

Як можна побачити на рисунку, адміністратором було добавлено кілька товарів, типів товару, зображень товару.

Адміністратор може добавляти користувачів, добавляти товари.

Менеджери можуть добавляти товари.

Користувачи можуть створювати замовлення.

Зв'язок між frontend та backend відбувається за допомогою GraphQL API. Точка API прив'язана до backend за адресою <http://localhost:5000/api/graphql>. Frontend виконує запити об'єктів сутностей згідно зі схемою моделей сутностей.

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

38

Приклад створення типу продуктів приведено на рис. 3.3.

The screenshot shows the Keystone 6 admin interface. The browser address bar displays the URL: 127.0.0.1:5000/item-types/cl3rpzrpz0309ccuc270gaunx. The page title is 'Keystone 6 Item Types > Макіяж'. The user is signed in as 'admin'. The sidebar on the left contains navigation links: Dashboard, Users, User Addresses, Items Carts, Cart Items, Bookings, Items, Discounts, Item Types (highlighted), Item Images, Payings, and Deliveries. The main form for 'Макіяж' includes the following fields: Name (Макіяж), Item ID (cl3rpzrpz0309ccuc270g), Parent (Косметика і парфумерія), Items (Select...), Created At (29.05.2022 22:53:59.019), and Last Modification (29.05.2022 23:27:19.122). Below the form are buttons for 'Save changes', 'No changes', and 'Delete'. A green notification banner at the bottom right says 'Saved successfully'.

Рисунок 3.3. Приклад введення типу продуктів.

3.2. Програмна документація на програмний за стосунок.

В даному розділі приведена інструкція для користувачів застосунку.

Для розгортання спроектованої системи необхідно встановити додаткове програмне середовище. По-перше, з офіційного сайту необхідно завантажити фреймворк Node.JS та інстальювати його. По-друге, з офіційного сайту необхідно завантажити та інстальювати застосунок Docker.

Наступного кроку необхідно інстальювати менеджер пакетів Yarn. Менеджер інстальюється автоматично за командою.

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

39

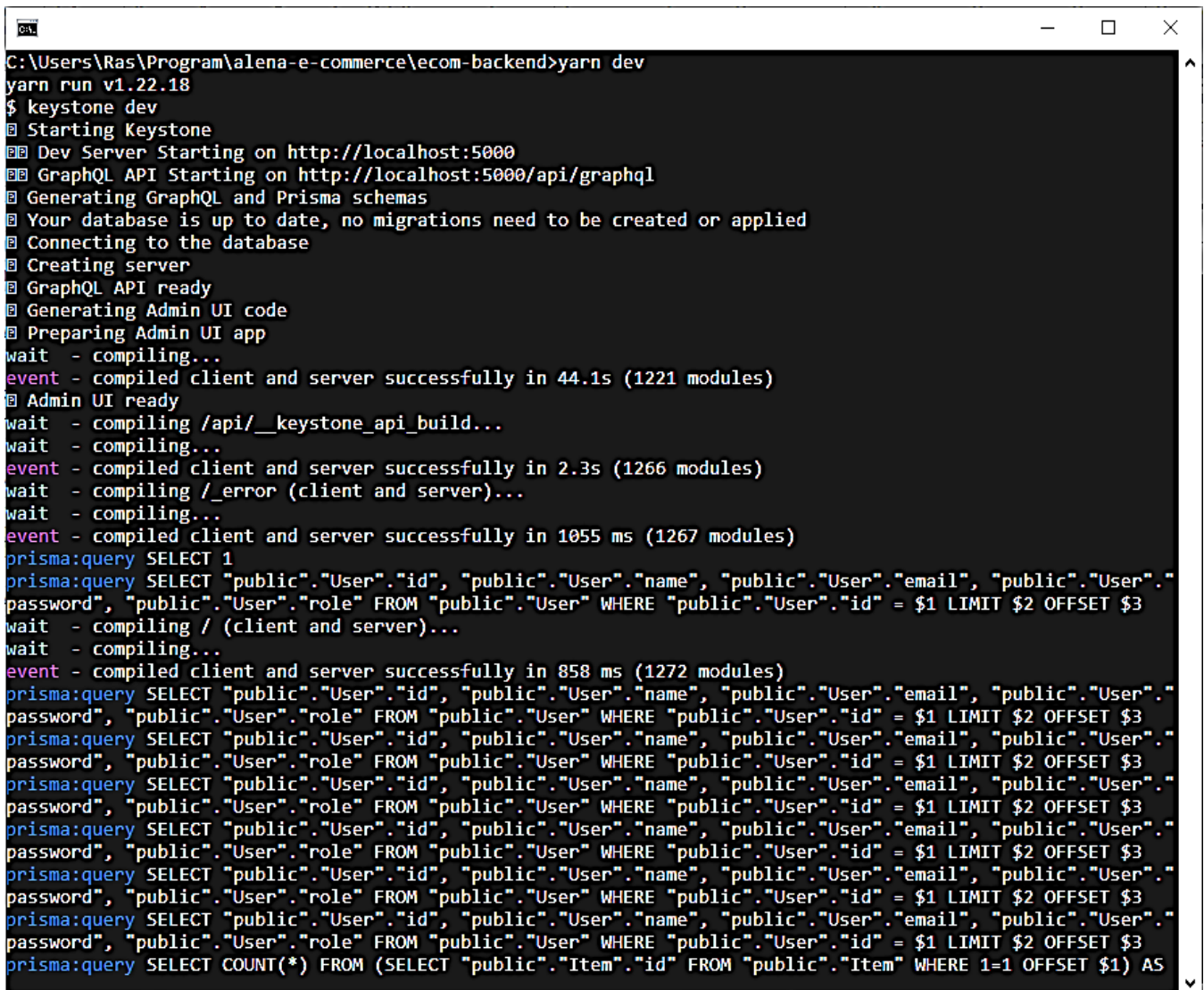
Для встановлення Yarn необхідно в командному рядку виконати команду “npm install –g Yarn” один раз на початку роботи.

Далі необхідно завантажити репозиторій спроектованої системи з платформи GitHub: <https://github.com/drA666/alena-e-commerce>, та розпакувати його до робочого каталогу.

Наступні команди виконуються в командному рядку NodeJS.

- startBD.sh (Linux) startBD.cmd (Windows) – для запуску СУБД
- startFrontend.sh startFrontend.cmd – для запуску frontend компонента
- startBackend.sh startBackend.cmd – для запуску backend компонента

Про успішний запуск свідчить записи лога в командному рядку (рис. 3.4)



```
C:\Users\Ras\Program\alena-e-commerce\ecom-backend>yarn dev
yarn run v1.22.18
$ keystone dev
Starting Keystone
Dev Server Starting on http://localhost:5000
GraphQL API Starting on http://localhost:5000/api/graphql
Generating GraphQL and Prisma schemas
Your database is up to date, no migrations need to be created or applied
Connecting to the database
Creating server
GraphQL API ready
Generating Admin UI code
Preparing Admin UI app
wait - compiling...
event - compiled client and server successfully in 44.1s (1221 modules)
Admin UI ready
wait - compiling /api/_keystone_api_build...
wait - compiling...
event - compiled client and server successfully in 2.3s (1266 modules)
wait - compiling /_error (client and server)...
wait - compiling...
event - compiled client and server successfully in 1055 ms (1267 modules)
prisma:query SELECT 1
prisma:query SELECT "public"."User"."id", "public"."User"."name", "public"."User"."email", "public"."User"."password", "public"."User"."role" FROM "public"."User" WHERE "public"."User"."id" = $1 LIMIT $2 OFFSET $3
wait - compiling / (client and server)...
wait - compiling...
event - compiled client and server successfully in 858 ms (1272 modules)
prisma:query SELECT "public"."User"."id", "public"."User"."name", "public"."User"."email", "public"."User"."password", "public"."User"."role" FROM "public"."User" WHERE "public"."User"."id" = $1 LIMIT $2 OFFSET $3
prisma:query SELECT "public"."User"."id", "public"."User"."name", "public"."User"."email", "public"."User"."password", "public"."User"."role" FROM "public"."User" WHERE "public"."User"."id" = $1 LIMIT $2 OFFSET $3
prisma:query SELECT "public"."User"."id", "public"."User"."name", "public"."User"."email", "public"."User"."password", "public"."User"."role" FROM "public"."User" WHERE "public"."User"."id" = $1 LIMIT $2 OFFSET $3
prisma:query SELECT "public"."User"."id", "public"."User"."name", "public"."User"."email", "public"."User"."password", "public"."User"."role" FROM "public"."User" WHERE "public"."User"."id" = $1 LIMIT $2 OFFSET $3
prisma:query SELECT "public"."User"."id", "public"."User"."name", "public"."User"."email", "public"."User"."password", "public"."User"."role" FROM "public"."User" WHERE "public"."User"."id" = $1 LIMIT $2 OFFSET $3
prisma:query SELECT "public"."User"."id", "public"."User"."name", "public"."User"."email", "public"."User"."password", "public"."User"."role" FROM "public"."User" WHERE "public"."User"."id" = $1 LIMIT $2 OFFSET $3
prisma:query SELECT COUNT(*) FROM (SELECT "public"."Item"."id" FROM "public"."Item" WHERE 1=1 OFFSET $1) AS
```

Рис. 3.4. Запис лога в командному рядку

Створення нового користувача відбувається як при вході в адміністративну панель backend, в середині адміністративної панелі backend, так і в компоненті frontend. На рис. 3.5 показана картка зі створення користувача у компоненті backend. В картці є детальна інформація про користувача: ім'я, електронна скринька, пароль користувача, роль користувача, та адреса користувача. Обов'язковими полями є всі поля окрім адреси.

The screenshot shows a web browser window with the URL '127.0.0.1:5000'. The page title is 'Keystone 6'. The main content area is titled 'Create User'. It contains the following fields and controls:

- Name:** Text input field containing 'manager1'.
- Email:** Text input field containing 'manager1@mana.m'.
- Password:** Two text input fields, both masked with dots. Each has a visibility toggle (eye icon) and a close button (X).
- Role:** Dropdown menu with 'worker' selected.
- Useraddress:** Dropdown menu with 'Select...' selected.
- Create related User Address:** Button below the Useraddress dropdown.
- Create User:** Blue button at the bottom left.
- Cancel:** Text button at the bottom right.

Рисунок 3.5. Картка зі створення користувача у компоненті backend.

У межах контролю доступу та безпеки є певна політика складності пароля: мінімум 8 символів, повинні бути як цифри так і літери верхнього та нижнього регістрів.

Якщо пароль має недостатню складність, то користувачеві буде видано повідомлення з даною інформацією (рис. 3.6).

Name
Петренко Петро Петрович

Email
petro@ppp.pp

Password
.....
Password must be at least 8 characters long

Role
customer

Рисунок 3.6. Повідомлення користувачу про вимогу до паролю
Картка для заповнення адреси користувача показана на рис. 3.7.

Address Name
address 1

Name
manager1

Street Address
Warshavska

Street Address2

City
Kyiv

Post Code
00000

Country
Country must not be empty

Tel No

Create User Address Cancel

Рисунок 3.7. Картка адреси користувача

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

42

Картка товару зображена на рис. 3.8. Обов'язковими до заповнення є поля назви товару, його опису та ціни товару.

The screenshot shows the 'Create Item' form in the Keystone 6 dashboard. The form is titled 'Create Item' and is part of the 'Dashboard'. The user is signed in as 'admin'. The form fields include: 'Name' (O'BAYS Satin Eye Liner), 'Description' (empty), 'Seo Desc' (Контурний олівець для очей), 'Item Types' (Макіяж), 'Item Images' (d3rt0etf0723ccucladd6a1c), and 'Price' (100.0000). A green success message at the bottom right reads 'Created Successfully' with a checkmark icon. The left sidebar shows navigation options like Users, Items, Bookings, Discounts, Item Types, Item Images, Payings, and Deliveries.

Рисунок 3.8. Картка товару

Користувач може вказати один чи кілька типів товару. Також користувач може завантажити одне чи декілька зображень до картки товару.

Система є оптимізованою для пошукових систем. Назва та опис товару мають підвищену вагу при індексації контенту. Тому користувач повинен приділити підвищену увагу при заповненні назви та опису товару. Існують певні рекомендації при заповненні назви та опису товару:

- 1) не має бути граматичних помилок при заповненні назви та опису
- 2) ключові слова мають бути як в назві так і в опису товару

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

43

Картка зображення товару показана на рис. 3.9. Поле альтернативного запису Alt є обов'язковим. З метою оптимізації пошуковими системами бажано, щоб поле Alt містило ключові слова продукту.

Адміністратор або менеджер можуть прикріпляти зображення.

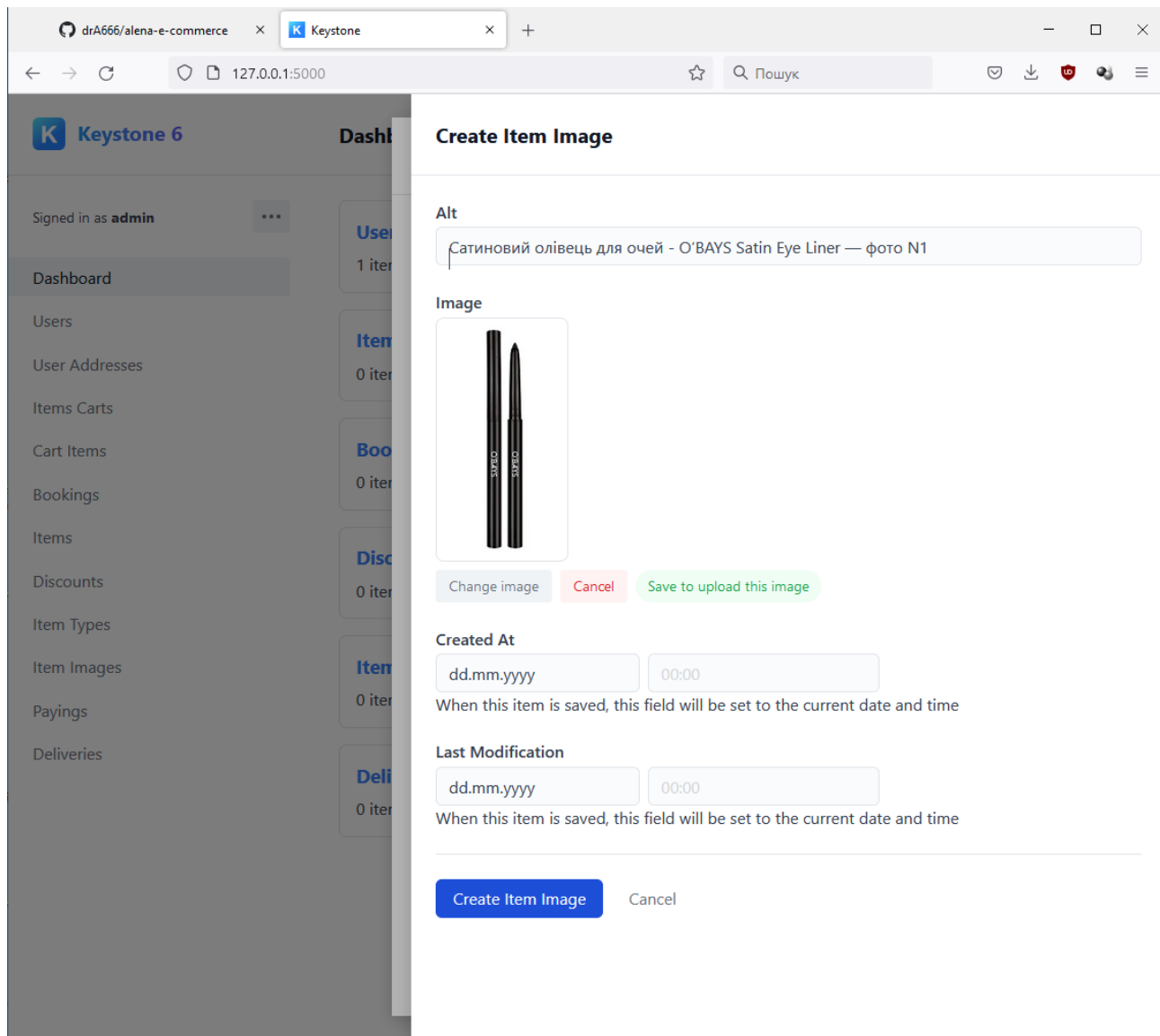


Рисунок 3.9. Картка зображення товару.

Товар має тип. Типи товарів мають ієрархічну структуру. Типи товарів дозволяють об'єднувати товари в групи. У компоненті frontend типам товару відповідають окремі пункти меню.

Типи товарів вносять додаткову вагу у оптимізацію сторінок товару для пошукових систем. Тому бажано, щоб назвами типів товарів з'являлися ключові слова.

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

44

В картці типу товарів можуть бути прив'язані товари.

Картка типу товарів показана на рис. 3.10. Тип товару мають можливість створювати адміністратор або менеджер-продавець.

The screenshot displays the 'Item Types' configuration page in the Keystone 6 system. The page is titled 'Item Types > Парфумерія'. On the left, a sidebar lists various system components, with 'Item Types' currently selected. The main area contains several input fields and buttons: 'Name' is set to 'Парфумерія'; 'Item ID' is 'c13ronzzb0235ccu'; 'Parent' is 'Косметика і парфумерія'; 'Items' is set to 'Select...'; 'Created At' shows two timestamps: '29.05.2022' and '22:16:50.260'; 'Last Modification' shows '29.05.2022' and '23:27:05.983'. A note states: 'When this item is saved, this field will be set to the current date and time'. At the bottom, there are three buttons: 'Save changes' (blue), 'No changes' (grey), and 'Delete' (red).

Рис. 3.10. Картка типу товарів

У картці типу товарів поле ім'я типу товарів є обов'язковим.

Адміністратор або менеджер можуть додати нову знижку. Для знижки є обов'язковими поля відсотку знижки, граничної вартості замовлення, терміну дії знижки.

Користувач не має змоги редагувати знижку.

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

45

Картка знижки зображена на рис. 3.11.

The screenshot shows a web browser window with the URL 127.0.0.1:5000/discounts. The page title is 'Keystone 6' and the main heading is 'Create Discount'. The form contains the following fields:

- Discount:** Input field with the value '50'.
- Next Delivery:** Date and time input fields. The date is '20.05.2022' and the time is '00:00'.
- Amount In Next Delivery:** Input field with the value '1000'.
- Created At:** Date and time input fields. The date is 'dd.mm.yyyy' and the time is '00:00'. Below the field is the text: 'When this item is saved, this field will be set to the current date and time'.
- Last Modification:** Date and time input fields. The date is 'dd.mm.yyyy' and the time is '00:00'. Below the field is the text: 'When this item is saved, this field will be set to the current date and time'.

At the bottom of the form, there are two buttons: 'Create Discount' (in blue) and 'Cancel'.

Рис. 3.11. Картка знижки на товари

Користувач-покупець при першому вході повинен зареєструватися. При реєстрації обов'язковими до заповнення є поля Ім'я, Email, Password.

До поля пароль діють певні правила зі складності: пароль повинен бути не менш за вісім символів, складатися з цифр, як мінімум однієї великої та як мінімум однієї малої літери. Якщо пароль не відповідає рівню складності, то користувачеві буде видане відповідне повідомлення.

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

46

Початкова сторінка покупця приведена на рис. 3.12



Головна / Увійти

Увійти

Shoot! Authentication failed.

Email

Password



Зареєструватись

Ім'я

Email

Password



Рис. 3.12. Початкова сторінка покупця

Після авторизації покупцеві відкривається список товарів. Список розбитий за сторінками. На кожній сторінці розташовано шість одиниць товару. Позиція товару складається з назви, фотографії, скороченого опису та ціни. Є можливість навігації між сторінками списку. У покупця є можливість подивитися замовлення, які були зроблені раніше. Є можливість перейти в корзину поточного замовлення. Також є можливість вийти зі свого облікового запису, якщо перейти за відповідним посиланням. Користувач також має можливість детально подивитись на кожний товар якщо перейти за відповідним посиланням.

Список товарів для покупця приведено на рис. 3.13

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467200.003 ПЗ

Арк.

47

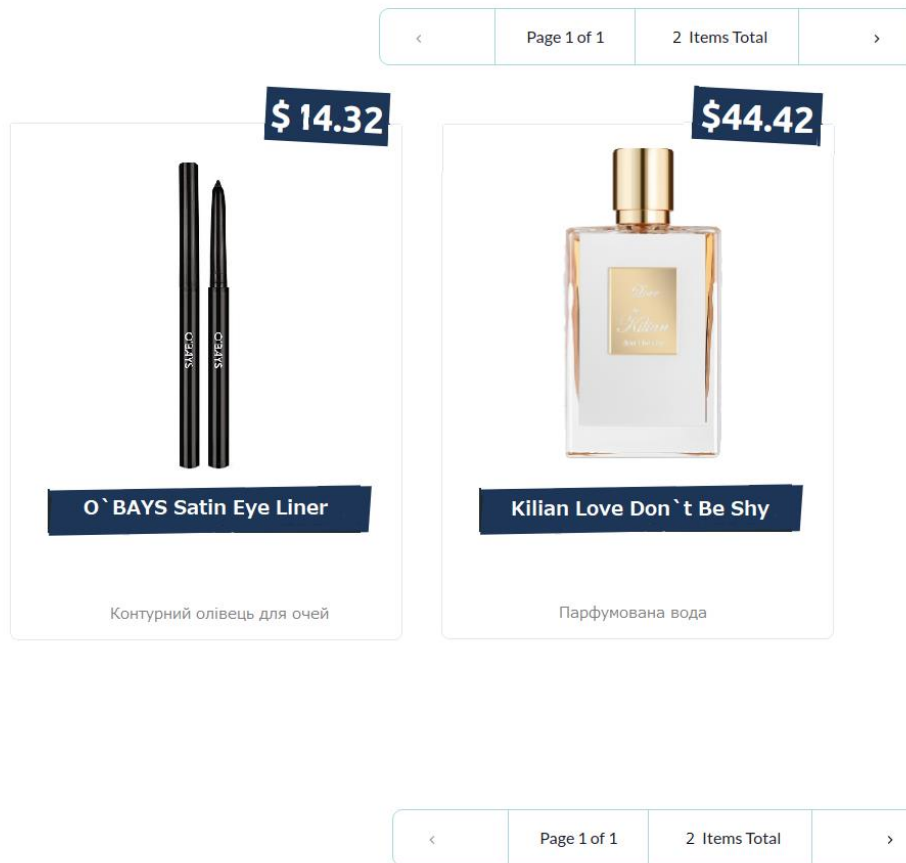


Рис. 3.13. Список товарів для покупця

Детальний опис товару приведено на рис. 3.14

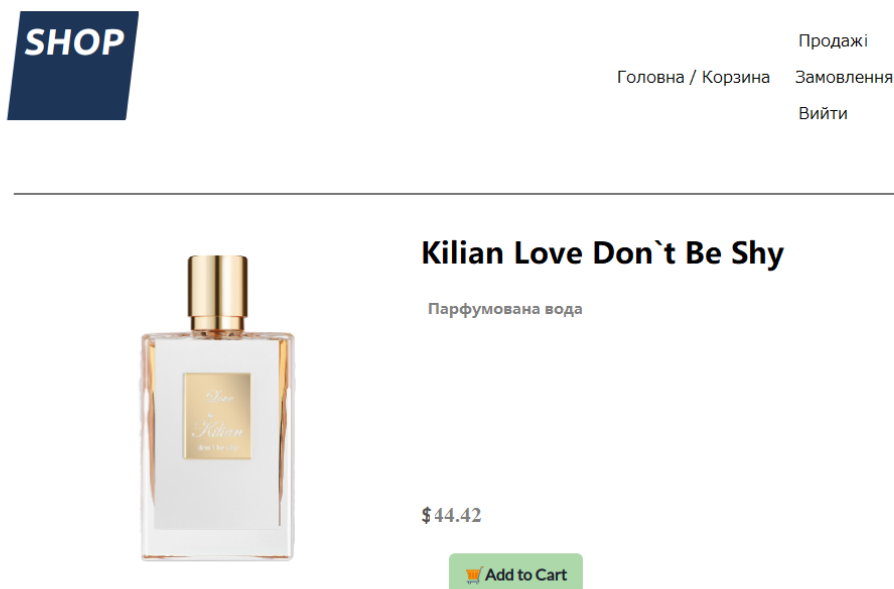


Рис. 3.14. Детальний опис товару

Змн.	Арк.	№ докум.	Підпис	Дата


Кошик користувача приведений на рис. 3.15

SHOP

Закази

ORDER ID	CHARGE ID
60683301143cb988cfa6a494	pi_11c5gcGwSvHwY0aOEcg8Snz
606881fcb4ef9d001c10ff04	pi_11cAtDGwSvHwY0aO9GsLMx

Корзина

 **Kilian Love Don't Be Shy**
\$88.84
(2 x \$44.42 each)

Card number MM / YY

Checkout

\$88.84 Close

Рис. 3.15. Кошик користувача

Висновки за 3 розділом.

В даному розділі було продемонстровано практичні можливості розробленого програмного застосунку для реалізації процесів електронної комерції.

Також в розділі приведена програмна документація на програмний застосунок, у якій описана інструкція для користувачів застосунку.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

ВИСНОВКИ.

В дипломному проєкті розроблено програмний застосунок для реалізації електронної комерції.

Аналіз сучасного стану процесів електронної комерції та відомих рішень показав, що існуючі системи орієнтовані середній та великий бізнес та потребують значних ресурсів для впровадження. Безперечним є той факт, що кількість користувачів, які беруть участь в електронній комерції, постійно зростає. Тому актуальність теми проєкту залишається на високому рівні. Готові рішення, наприклад, для великого бізнесу не завжди підходять для малого бізнесу та навпаки.

Проектування програмного застосунку здійснене на основі проведеного аналізу та згідно з вимогами до системи. В ході проектування визначено структурні складові застосунку; визначено діючі лиця та сценарії за якими вони діють; визначено сутності об'єктів. Цільовим призначенням проєкту є стартапи малого бізнесу; проєкт може бути розгорнутий як в Unix-подібних системах, так і в системі Windows чи в системі MacOS, тобто є кросплатформним; додаткове програмне забезпечення, яке складає інфраструктуру проєкту надає програмному застосунку безпеку, надійність, відмовостійкість, масштабованість та швидкість.

Розробка програмного застосунку здійснена згідно розробленої архітектури. Описані моделі даних, скомпонований застосунок. Протестоване систему. В проєкті впроваджено контроль доступу на основі розділу користувачів згідно ролей та обмеження доступу для ролей користувачів, які не є адміністраторами. Застосунок обробляє запити асинхронно, що відрізняє цей проєкт, наприклад, від PHP (дякуючи асинхронності в проєкт можна додати елементи аукціону).

У якості прикладу застосування розробленої програмної інформаційної системи у проєкті приведений інтернет-магазин косметики. До проєкта додається графічний матеріал в межах єдиної системи програмної документації.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

Таким чином, усі завдання проекту виконані у повному обсязі, а мета досліджень, яка полягає у автоматизації професійної діяльності торговельних підприємств із впровадженням технологій глобальних інформаційних систем та процесів досягнуто.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

СПИСОК ВИКОРИСТАНИХ ПЕРШОДЖЕРЕЛ – ЛІТЕРАТУРИ

1. Синявська О.О. Електронна торгівля в Україні: тенденції та перспективи розвитку. *Вісник ХНУ імені В.Н. Каразіна. Серія «Міжнародні відносини. Економіка. Країнознавство. Туризм»*. 2019. Вип. 9. С. 126–132.
2. Startsev M.V. Electronic commerce as the way of the intensification of business processes. *Соціально-економічні явища та процеси*. 2011. № 5-6 (027-028). С. 212-215.
3. Zheng Qin. Introduction to E-commerce. Beijing: Tsinghua University Press, 2009. 517 с.
4. Model–view–controller [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Model-view-controller> (дата звернення 12.05.2022)
5. eCommerce Usage Distribution in the Top 1 Million Sites [Електронний ресурс] – Режим доступу до ресурсу: <https://trends.builtwith.com/shop> (дата звернення 14.05.2022)
6. Adobe Commerce 2.4 Developer Guide. Architecture Guide. Magento architectural diagrams [Електронний ресурс] – Режим доступу до ресурсу: https://devdocs.magento.com/guides/v2.4/architecture/archi_perspectives/arch_diagrams.html (дата звернення 10.05.2022)
7. Shopify. For developers. Theme architecture. [Електронний ресурс] – Режим доступу до ресурсу: <https://shopify.dev/themes/architecture> (дата звернення 12.05.2022)
8. Keystone 6. [Електронний ресурс] – Режим доступу до ресурсу: <https://keystonejs.com/> (дата звернення 11.05.2022)
9. Hiren Dhaduk. How to Build a Node.js Ecommerce App? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.simform.com/blog/build-ecommerce-web-application-node-js/> (дата звернення 14.05.2022)
10. Maciej Krawczyk. Build Your Own E-Commerce Keystone.js-Based System — Requirements and Architecture [Електронний ресурс] – Режим доступу до ресурсу: <https://betterprogramming.pub/build-your-own-e-commerce-keystone->

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

- [js-based-system-requirements-and-architecture-3d639241b963](#) (дата звернення 11.05.2022)
11. Docker overview [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.docker.com> (дата звернення 11.05.2022)
12. Yarn package overview [Електронний ресурс] – Режим доступу до ресурсу: <https://yarnpkg.com> (дата звернення 11.05.2022)
13. Maciej Krawczyk. Build Your Own E-Commerce Keystone.js-Based System — Environment Setup and Base Models [Електронний ресурс] – Режим доступу до ресурсу: <https://betterprogramming.pub/build-your-own-e-commerce-keystone-js-based-system-environment-setup-and-base-models-2c02a3e3a70b> (дата звернення 18.05.2022)
14. Maciej Krawczyk. Building Your Own E-Commerce Keystone.js-Based System — Access Control [Електронний ресурс] – Режим доступу до ресурсу: <https://betterprogramming.pub/building-your-own-e-commerce-keystone-js-based-system-access-control-1a366ed7e064> (дата звернення 17.05.2022)
15. Спиринцев В.В. Web-технології та Web-дизайн: HTML, CSS [Текст]: навч. посіб. / В.В. Спиринцев, В.В. Гнатушенко, О.С. Волковський //.-Дніпро: «Ліра», 2017.-163с.
16. Дронов В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / В.А. Дронов - СПб.: БХВ-Петербург, 2011. - 416 с.: ил. - (Профессиональное программирование).
17. Росс В. С. Создание сайтов: HTML, CSS, PHP, MySQL/ В.С. Росс [Учебное пособие, ч. 1]- МГДД(Ю)Т, М.:2010 - 107 с.
18. Лаврищева К.М. Програмна інженерія. – К.: 2008. – 319 с.
19. Трофименко О. Г. Веб-технології та веб-дизайн : навч. посібник / О. Г. Трофименко, О. Б. Козін, О. В. Задерейко, О. Є. Плачінда. – Одеса : Фенікс, 2019. – 284 с.
20. Ташков П. А. Т25 Веб-мастеринг на 100 %: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка /П.А.Ташков. - СПб.: Питер, 2010. - 512 с.: ил. - (Серия «На 100%»).

21. Пасічник О.Г. Основи веб-дизайну /О.Г.Пасічник, О.В.Пасічник, І.В.Стеценко: [Навч.посіб.].-К.:Вид. група ВНУ.- 2009 .- 336 с.: іл.
22. Лаврищева Е.М. Основы инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева, В.Ю. Суслов. – 2-е изд., перераб. и доп. – К.: Академкнига, 2007. – 672 с.
23. Lebedenko S., Lebedenko M. Estimation of web-site effectiveness on B2B market / S. Lebedenko, M. Lebedenko // Journal L'Association 1901 "SEPIKE", # 7, Deutschland, France, USA, 2015
24. What Does That Server Really Serve? [Електронний ресурс]. – Режим доступу: - <http://bostonreview.net/richard-stallman-free-software-DRM>
25. Адаптивний веб-дизайн. [Електронний ресурс]. – Режим доступу: - https://ru.wikipedia.org/wiki/Адаптивный_веб-дизайн
26. Савельева Н. Системы управления контентом // Открытые системы. — 2004. — No 4.
27. Економіка підприємства: Навч. посібник / [П.В. Круш, В.І. Подвігіна, Б.М.Сердюк]; За заг. ред. П.В. Круша – К. :Ельга–Н, КНТ, 2007. – 780с.
28. Мельник О.В. – Електронна комерція: онлайн підручник. [Електронний ресурс]. – Режим доступу: - <http://intkonf.org/melnik-ov-elektronna-komertsiya-yak-skladova-chastinaelektronnogo-biznesu/>
29. Nanekaran, Y. An Introduction To Electronic Commerce. International Journal of Scientific & Technology Research, 2 (4), 190-193. 2013.
30. Вплив інтернет-технологій на підприємницьку діяльність суб'єктів господарювання України : збірник тез за матеріалами міжнар. наук.-практ. конф., 25 листопада 2010 р. / М-во освіти і науки України, Національний технічний університет України „КПІ”. — К.:Політех. 2010. — 202 с.

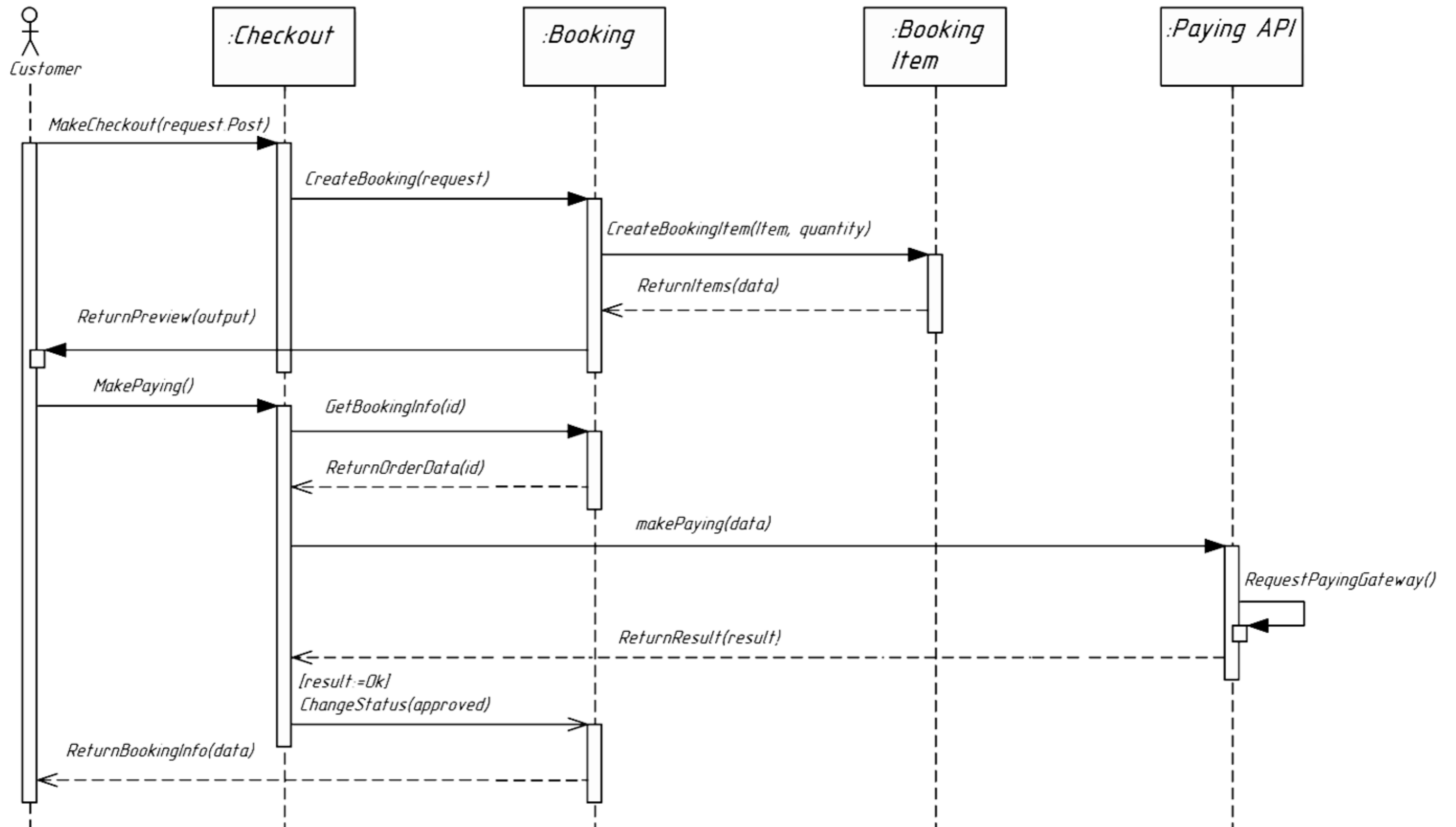
ДОДАТОК А

до дипломного проекту
на тему: «**Програмний застосунок для реалізації процесів
електронної комерції**»

Р-схема алгоритму

ІАЛЦ.467200.004 Д1

Аркушів 1



					ІАЛЦ.467200.004 Д1		
					Програмний застосунок для реалізації процесів електронної комерції		
					P-схема алгоритму		
					Літера	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата			
Розробив		Растюк А.В.					
Перевірів		Писарчук О.О.					
Реценз.					Аркуш 1	Аркушів 1	
Н.контроль		Симоненко В.П.			Дипломна робота		
Затв.							
					НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІП-83		

ДОДАТОК Б

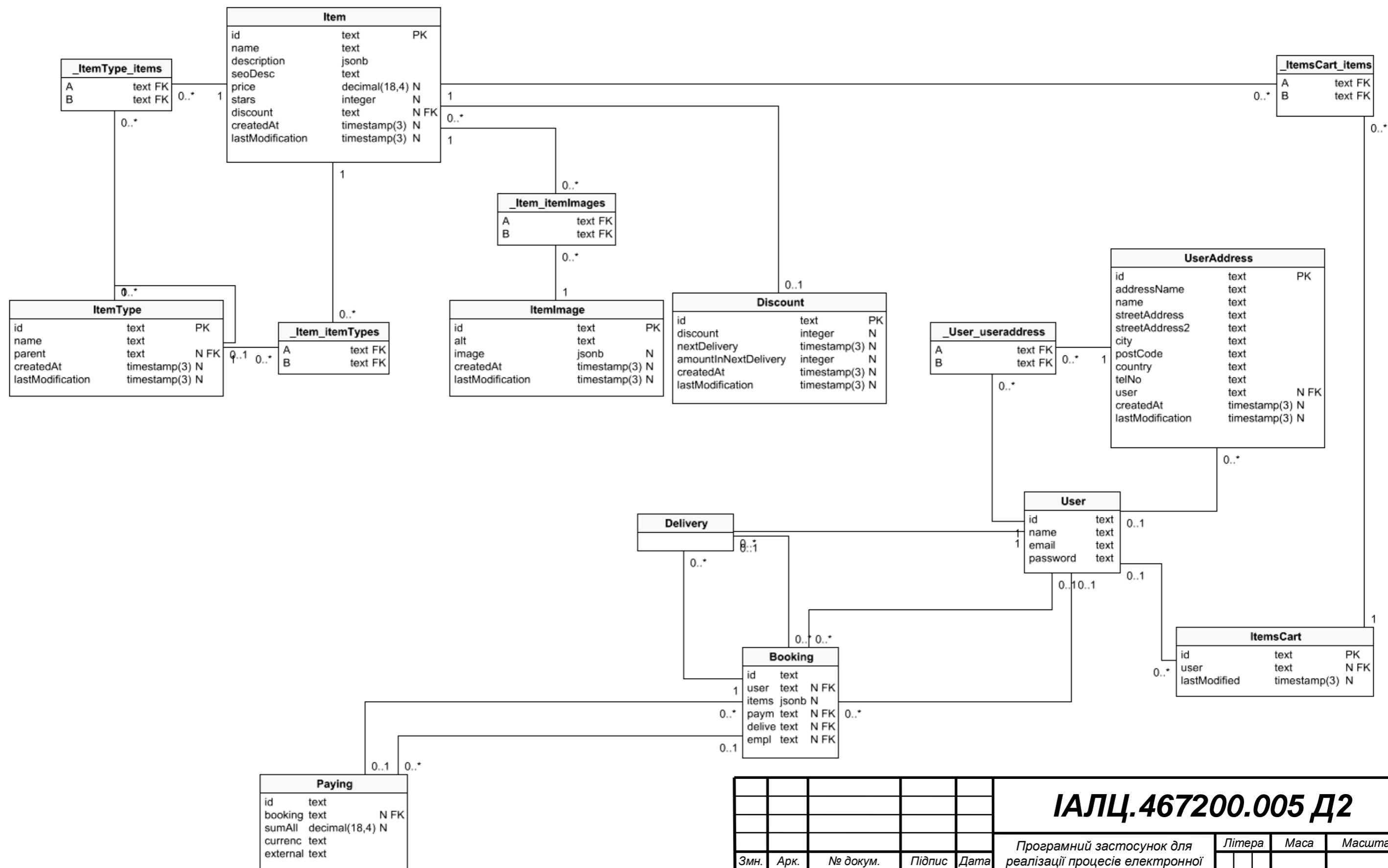
до дипломного проекту
на тему: «Програмний застосунок для реалізації процесів
електронної комерції»

**Функціональна схема
(діаграма класів)**

ІАЛЦ.467200.005 Д2

Аркушів 1

Київ – 2022 р.



					ІАЛЦ.467200.005 Д2			
					Програмний застосунок для реалізації процесів електронної комерції			
					Функціональна схема			
					Літера		Маса	Масштаб
					Аркуш 1		Аркушів 1	
					Дипломна робота			
					НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІП-83			

Змн.	Арк.	№ докум.	Підпис	Дата
Розробив		Расюк А.В.		
Перевірює		Писарчук О.О.		
Реценз.				
Н.контроль		Симоненко В.П.		
Затв.				

ДОДАТОК В

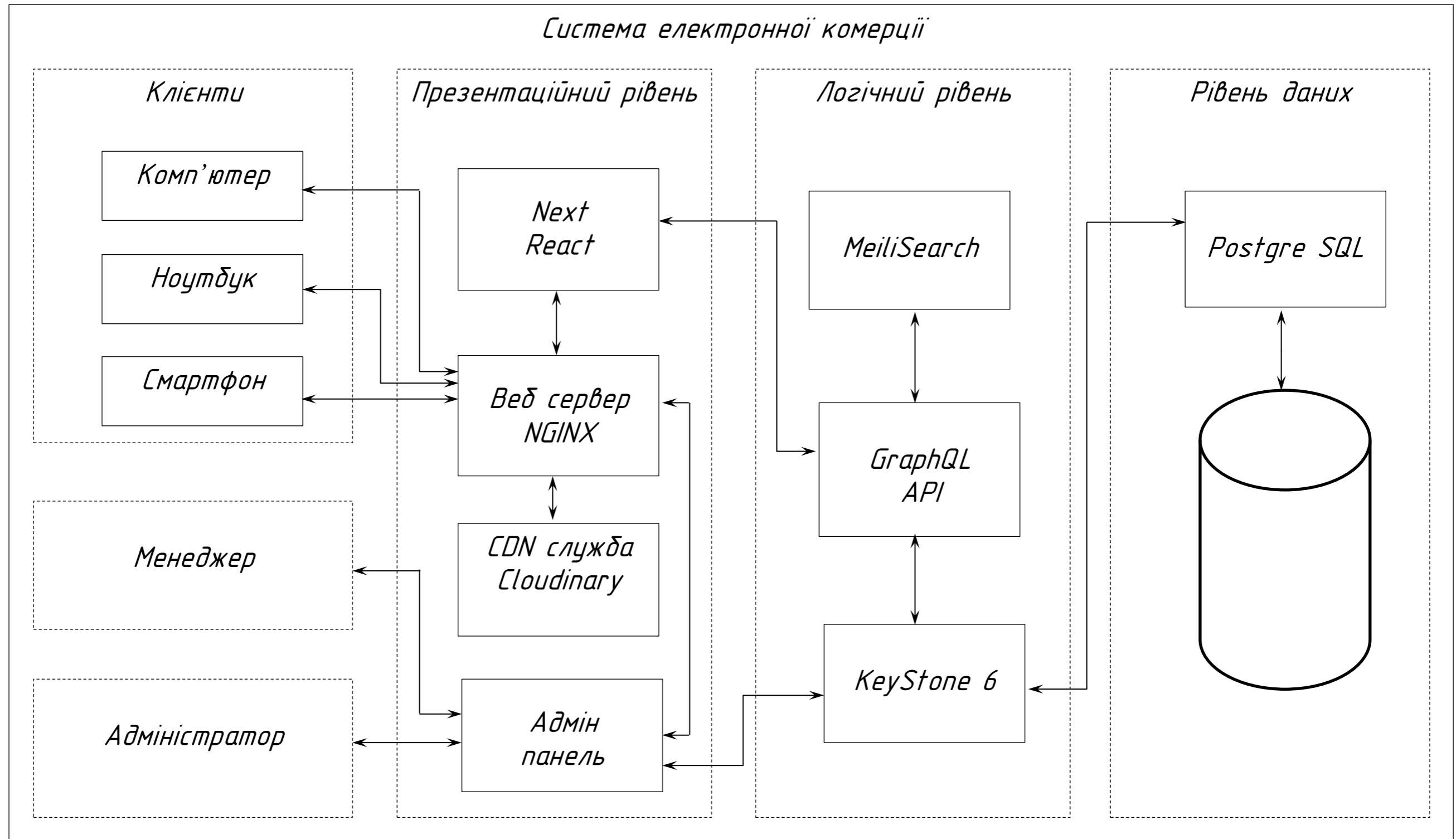
до дипломного проекту
на тему: «Програмний застосунок для реалізації процесів
електронної комерції»

Структурна схема

ІАЛЦ.467200.006 ДЗ

Аркушів 1

Київ – 2022 р.



					ІАЛЦ.467200.006 ДЗ		
					Програмний застосунок для реалізації процесів електронної комерції		
					Структурна схема		
					Літера	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата			
Розробив		Расюк А.В.					
Перевірів		Писарчук О.О.					
Реценз.					Аркуш 1	Аркушів 1	
Н.контроль		Симоненко В.П.			НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІП-83		
Затв.					Дипломна робота		

ДОДАТОК Г

до дипломного проекту
на тему: «**Програмний застосунок для реалізації процесів
електронної комерції**»

Опис програми

ІАЛЦ.467200.007 Д4

Аркушів 3

Київ – 2022 р.

АНОТАЦІЯ

Система «Alena-e-commerce» призначена для створення кросплатформних інтернет-магазинів на базі JavaScript–оточенні Node.JS.

Основним призначенням застосунку є автоматизація процесів електронної комерції. Ця система орієнтована на стартапи малого бізнесу.

Система «Alena-e-commerce» має GraphQL API, за допомогою чого може інтегруватися зі сторонніми системами.

ANNOTATION

The Alena-e-commerce system is designed to create cross-platform online stores based on JavaScript-environment Node.JS.

The main purpose of the application is to automate e-commerce processes. This system is aimed at small business startups.

Alena-e-commerce system has GraphQL API, which allows it to integrate with third-party systems.

ЗМІСТ

Загальні відомості	2
Функціональне призначення	2
Опис логічної структури.....	2
Використані технічні засоби	2
Виклик й завантаження.....	3
Вхідні дані	3
Вихідні дані	3

ІАЛЦ.467200.007 Д4								
					Програмний застосунок для реалізації процесів електронної комерції Опис програми			
Змн.	Арк.	№ докум.	Підпис	Дата		Літ.	Арк.	Акрушіє
Розробив		Расюк А.В.					3	1
Перевірів		Писарчук О.О.				НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІП-83		
Реценз.								
Н.контроль		Симоненко В.П.						
Затв.								

ЗАГАЛЬНІ ВІДОМОСТІ

Alena-e-commerce — це система електронної комерції з відкритим кодом, призначена для створення інтернет-магазинів. Ця програма розроблена з використанням мови JavaScript та TypeScript.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Система Alena-e-commerce призначена для автоматизації професійної діяльності торгівельних підприємств із впровадженням технологій глобальних інформаційних систем та процесів.

Всі користувачі системи діляться на три ролі: адміністратор, менеджер-продавець та покупець. Адміністратор та менеджер керують контентом: заповнюють товари та обробляють замовлення.

Покупці ознайомлюються з описом товарів та створюють замовлення.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Система Alena-e-commerce складається з системи управління базою даних, де зберігаються дані або інформація і серверної частини, що з'єднує дані з браузером.

Для управління контентом призначена адміністративна панель.

Почуєць взаємодіє з системою за допомогою браузера.

Також складовою частиною системи є точка інтерфейсу GraphQL API. За допомогою API з системою Alena-e-commerce взаємодіють сторонні системи.

ВИКОРИСТАНІ ТЕХНІЧНІ ЗАСОБИ

Інфраструктура програми складається з

- 1) Програмної платформи Node.JS;
- 2) Docker – віртуальної станції;

					ІАЛЦ.467200.007 Д4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

3) менеджера пакетів Yarn

Ці компоненти встановлюються окремо перед використанням системи Alena-e-commerce

ВИКЛИК Й ЗАВАНТАЖЕННЯ

Репозиторій системи розташований за адресою <https://github.com/drA666/alena-e-commerce>. Для старту системи служать скрипт-файли:

- startBD.sh (Linux) startBD.cmd (Windows) – для запуску СУБД
- startBackend.sh startBackend.cmd – для запуску backend компонента
- startFrontend.sh startFrontend.cmd – для запуску frontend компонента

ВХІДНІ ДАНІ

На вході системи є опис продуктів, зображення до продуктів, дані користувачів, замовлення покупців.

ВИХІДНІ ДАНІ

На виході системи є виконані замовлення, та точки GraphQL API для підключення сторонніх систем.

					ІАЛЦ.467200.007 Д4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3