

КУРСОВА РОБОТА  
КР. ІІ – 27.00.00.000 ІЗ  
Група ІІ-22-1  
Хімії Денис  
2025

Івано-Франківський національний технічний університет нафти і газу

(повне найменування вищого навчального закладу)

Кафедра інженерії програмного забезпечення

(повна назва кафедри)

## КУРСОВА РОБОТА

з дисципліни «Технологія компонентного програмування для веб»

на тему: «Веб-сайт організації роботи книжкового магазину»

Студента 3 курсу групи **ІІІ-22-1**

Спеціальності **121** –

**Інженерія програмного забезпечення**

**Хімія Д.Ю.**

(прізвище та ініціали)

Керівник: **Ваврик Т.О.**

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала: \_\_\_\_\_

Кількість балів: \_\_\_\_ Оцінка ECTS: \_\_\_\_

Члени комісії

_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)

Івано-Франківський національний технічний університет нафти і газу

Кафедра: **Інженерія програмного забезпечення**

Дисципліна: **Об'єктно-орієнтоване програмування**

Спеціальність: **Інженерія програмного забезпечення**

Курс        **1**        Група        **ІІІ-22-1**        Семестр        **2**

## ЗАВДАННЯ НА КУРСОВУ РОБОТУ

Студенту                                **Хімію Денису Юрійовичу**

1. Тема роботи: **Веб-сайт організації роботи книжкового магазину**
2. Термін здачі студентом закінченої роботи: **5 червня 2025 року**
3. Вхідні дані до роботи: **згідно з обраним варіантом**
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити): **Анотація, вступ, теоретичний аналіз матеріалу, обґрунтування та аналіз предметної області, постановка завдання, створення уніфікованих React-компонентів, створення вебзастосунку на їх основі, генерація тестових даних, демонстрація роботи застосунку, висновок, список використаних джерел.**
5. Дата видачі завдання: **12 березня 2025 року**

## КАЛЕНДАРНИЙ ПЛАН NV

Номер і назва етапів курсової роботи	Термін виконання етапів роботи	Примітка
1. Вибір варіанту завдання. Збір теоретичних матеріалів	21.03.25	Виконано
2. Постановка завдання. Аналіз вимог до розроблюваної ПС	28.03.25	Виконано
3. Проєктування та конструювання основних компонентів застосунку	25.04.25	Виконано
4. Реалізація та збірка вебзастосунку	09.05.25	Виконано
5. Генерація тестових даних. Тестування ПЗ	23.05.25	Виконано
6. Оформлення пояснювальної записки	30.05.25	Виконано
7. Захист курсового проєкту	05.06.2025	

Студент \_\_\_\_\_  
(підпис)

\_\_\_\_\_Хімій Д.Ю.\_\_\_\_\_  
(розшифровка підпису)

Керівник \_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(розшифровка підпису)

«\_12\_»\_\_\_березня\_\_\_2025р.\_\_\_\_

# Анотація

Курсова робота присвячена розробці вебзастосунку для інтернет-книгарні. Метою проекту є створення зручного, інтуїтивно зрозумілого та адаптивного інтерфейсу для користувачів, що дозволяє здійснювати пошук книжок (із використання фільтрів), перегляд інформації про окремі видання, додавання товарів до кошика та оформлення замовлення.

У межах реалізації створено головну сторінку, сторінки пошуку, перегляду детальної інформації про книги, кошика, а також сторінки входу та реєстрації. Для адміністраторів передбачено можливість редагування даних про книги. Окрему увагу приділено інформаційним сторінкам, таким як контактна інформація, умови доставки та оплати.

Інтерфейс містить спільні компоненти, зокрема картки з відображенням даних про книги, форми для вводу даних та їх поля, інформаційні сторінки (створені за єдиним шаблоном), меню навігації – Header та нижній колонтитул – Footer, які забезпечують зручну навігацію сайтом і підтримують цілісність користувацького досвіду. Потворне використання компонентів для спрощення процесу розробки. Використання CSS-стилів мінімізовано за допомогою бібліотеки React Bootstrap.

Функціональність вебзастосунку включає:

- авторизацію та автентифікацію користувачів;
- збереження вмісту кошика за допомогою локального сховища;
- засоби адміністрування (редагування каталогу книг);
- адаптивний дизайн із використанням React Bootstrap;

					КР. ІП – 27.00.00.000 ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Хімії Д.Ю.			Пояснювальна записка	Літ.		Арк.	Аркушів
Перевір.		Лютак І.З.						5	46
Реценз.						ІФНТУНГ, ІП-22-1			
Н. Контр.									
Затверд.									

– маршрутизацію сторінок за допомогою React Router.

Розробка вебзастосунку здійснювалася у середовищі Visual Studio Code з використанням середовища виконання Node.js та бібліотеки React.

**Ключові слова:** вебзастосунок, React, інтернет-книгарня, інтернет-магазин, книги, Bootstrap, Single Page Application, адаптивний дизайн.

					КР. ІП – 27.00.00.000 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	Пояснювальна записка		
Розроб.		Хімії Д.Ю.					
Перевір.		Лютак І.З.					
Реценз.							
Н. Контр.							
Затверд.							
					Літ.	Арк.	Аркушів
						6	46
					ІФНТУНГ, ІП-22-1		

# Зміст

Зміст.....	7
Вступ.....	9
1. Теоретична частина.....	11
1.1 Компонентне програмування .....	11
1.2 Принципи компонентного програмування .....	11
1.2.1 Відокремлення інтерфейсу від реалізації.....	11
1.2.2 Бінарна сумісність між клієнтом і сервером.....	12
1.2.3 Незалежність від мови програмування .....	13
1.2.4 Прозорість розташування .....	13
1.2.5 Керування одночасним доступом.....	14
1.2.6 Підтримка контролю версій.....	15
1.2.7 Безпека на основі компонентів.....	15
1.3 Single Page Application .....	16
1.4 Переваги SPA .....	17
1.5 Недоліки SPA .....	18
1.6 React.....	20
1.6.1 Особливості React.....	20

					КР. ІІ – 27.00.00.000 ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Хімії Д.Ю.			Пояснювальна записка	Літ.		Арк.	Аркушів
Перевір.		Лютак І.З.						7	46
Реценз.						ІФНТУНГ, ІІ-22-1			
Н. Контр.									
Затверд.									

2.	Практична частина .....	23
2.1	Актуальність теми .....	23
2.2	Аналіз вимог .....	24
2.2.1	Визначення користувачів системи .....	24
2.2.2	Цілі користувачів .....	25
2.2.3	Основні сценарії використання (use cases) .....	25
2.3	Розробка компонентів.....	26
2.3.1	Проектування компонентів відповідно до цілей користувачів ....	27
2.3.2	Структура сторінок та компонентів .....	28
2.4	Демонстрація проекту .....	32
2.4.1	Головна сторінка .....	33
2.4.2	Інформаційні сторінки .....	36
2.4.3	Авторизація .....	37
2.4.4	Каталог товарів .....	38
2.4.5	Сторінка опису книги .....	40
2.4.6	Сторінка редагування книги .....	41
2.4.7	Корзина .....	42
	Висновок .....	44
	Список використаних джерел .....	46

					КР. ІІ – 27.00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	<div>Пояснювальна записка</div> <div>ІФНТУНГ, ІІ-22-1</div>			
Розроб.		Хімії Д.Ю.						
Перевір.		Лютак І.З.						
Реценз.								
Н. Контр.								
Затверд.								
					Літ.	Арк.	Аркушів	
						8	46	



# Вступ

Сучасне суспільство дедалі більше користується цифровими технологіями у повсякденному житті, зокрема й для придбання товарів. Книги – не виняток, адже онлайн-книгарні дозволяють користувачам швидко знаходити потрібні видання, ознайомлюватися з їх описом та зручно оформлювати замовлення. У зв'язку з активним розвитком електронної комерції, створення вебзастосунків для продажу книжкової продукції є актуальним і затребуваним завданням.

Актуальність теми полягає в необхідності розробки зручних, функціональних і адаптивних вебінтерфейсів для інтернет-книгарень. Такі вебзастосунки повинні забезпечувати легкий доступ до каталогу книжок, зручний механізм додавання товарів до кошика, можливість авторизації користувачів, а також просту систему адміністрування книжкового контенту.

Об'єктом дослідження є процес створення клієнтської частини вебзастосунку для онлайн-магазину. Предметом дослідження виступає реалізація функціоналу інтернет-книгарні з використанням бібліотеки React.

**Мета курсової роботи** – розробити вебзастосунок інтернет-книгарні з адаптивним інтерфейсом та основними функціями для користувачів і адміністраторів.

## Завдання курсової роботи:

- Провести аналіз вимог до сучасного вебінтерфейсу інтернет-книгарні.
- Розробити архітектуру застосунку з урахуванням модульного підходу.
- Реалізувати такі сторінки:
  - головну сторінку з базовою інформацією про магазин,
  - сторінку **каталогу** з можливістю перегляду доступних книг,

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						9
Змін.	Арк.	№ докум.	Підпис	Дата		

- сторінку **опису окремої книги**,
  - сторінку **редагування даних про книгу** (для адміністратора),
  - сторінки **авторизації та автентифікації**,
  - сторінку **кошика** з підрахунком загальної вартості замовлення,
  - **інформаційні сторінки** (контакти, умови доставки тощо).
- Реалізувати контекстне управління кошиком і даними користувача за допомогою React Context API.
  - Забезпечити адаптивний дизайн інтерфейсу із використанням React Bootstrap.
  - Забезпечити уніфіковані компоненти.

У процесі роботи застосовано методи компонентно-орієнтованого програмування, особливо в контексті розробки графічної частини застосунку, а також принципи реактивного програмування. Основними технологіями виступили JavaScript, React, React Router, React Bootstrap. Розробка здійснювалася у середовищі Visual Studio Code з використанням Node.js.

При розробці використовувалася офіційна документація React, довідкові матеріали з React Bootstrap, а також аналіз функціоналу реальних інтернет-книгарень.

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						10
Змін.	Арк.	№ докум.	Підпис	Дата		

# 1.Теоретична частина

## 1.1 Компонентне програмування

Component-based software engineering, або компонентне програмування (далі КП) – це стиль розробки ПЗ, що спрямований на створення програмної системи з компонентів, слабо пов'язаних між собою та призначених для багаторазового використання.

[1]

Такий підхід забезпечує швидші цикли розробки, скорочує час виходу на ринок і підвищує якість програмного забезпечення. КП має кілька переваг. По-перше, воно дає змогу розробляти програмні системи модульно, що полегшує їх управління та супровід. Завдяки модульним компонентам розробники можуть зосереджуватися на окремих функціях, що забезпечує ефективніший процес розробки. Крім того, КП сприяє повторному використанню коду, зменшуючи надмірність і підвищуючи загальну ефективність розробки. Використовуючи перевірені й протестовані компоненти, розробники можуть заощадити час і зусилля на створення програмного забезпечення з нуля. До того ж, КП сприяє кращій співпраці між командами розробки, оскільки вони можуть зосередитися на конкретних компонентах, мінімізуючи конфлікти під час інтеграції коду.

[2]

## 1.2 Принципи компонентного програмування

### 1.2.1 Відокремлення інтерфейсу від реалізації

Фундаментальний принцип компонентного програмування полягає в тому, що базовою одиницею в застосунку є бінарно-сумісний (binary-compatible)

					КР. ІІ – 27.00.00.000 ПЗ	Арк.
						11
Змін.	Арк.	№ докум.	Підпис	Дата		

інтерфейс. Інтерфейс забезпечує абстрактне визначення сервісу між клієнтом і об'єктом. Цей принцип протиставляється об'єктно-орієнтованому підходу, в якому в центрі перебуває саме об'єкт, а не його інтерфейс. Інтерфейс забезпечує взаємодію між клієнтом та сервісом. Кожен сервіс реалізує інтерфейс у свій, незалежний спосіб. Інтерфейс реалізується бінарним компонентом, який є «чорною скринькою» і повністю інкапсулює свою внутрішню логіку.

Щоб використати компонент, клієнт має знати лише визначення інтерфейсу та мати доступ до бінарного компонента, який цей інтерфейс реалізує. Такий рівень опосередкування між клієнтом і об'єктом дозволяє замінити одну реалізацію інтерфейсу іншою без змін у клієнтському коді. Клієнт не потребує перекомпіляції для використання нової версії. Інколи навіть не потрібно перезапускати застосунок. Якщо інтерфейс залишається незмінним, об'єкти, що його реалізують, можуть еволюціонувати, і нові версії можна впроваджувати плавно й легко. Для реалізації функціональності, описаної інтерфейсом, у середині компонента використовують традиційні об'єктно-орієнтовані методи, але отримані ієрархії класів зазвичай простіші та легші в супроводі.

[3]

## 1.2.2 Бінарна сумісність між клієнтом і сервером

Ще одним ключовим принципом компонентного програмування є бінарна сумісність між клієнтом і сервером. У традиційному об'єктно-орієнтованому програмуванні всі учасники – як клієнти, так і сервери – повинні бути частиною єдиного монолітного застосунку.

Натомість компонентне програмування ґрунтується на пакуванні коду в компоненти – тобто бінарні будівельні блоки. Зміни в коді компонента

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						12
Змін.	Арк.	№ докум.	Підпис	Дата		

залишаються локалізованими всередині цього бінарного блоку; немає потреби перекомпілювати та повторно розгортати клієнтські застосунки. Однак можливість заміни та підключення нових бінарних версій сервера передбачає бінарну сумісність між клієнтом і сервером, тобто клієнтський код під час виконання повинен взаємодіяти з точками входу компонента, які точно відповідають його очікуванням щодо розміщення в пам'яті.

Ця бінарна сумісність є основою контракту між компонентом і клієнтом. Поки нова версія компонента дотримується цього контракту, клієнтський код залишатиметься незмінним і працездатним.

[3]

### 1.2.3 Незалежність від мови програмування

На відміну від традиційного об'єктно-орієнтованого програмування, у компонентному програмуванні сервер розробляється незалежно від клієнта. Оскільки клієнт взаємодіє із сервером лише під час виконання, єдиним зв'язком між ними є бінарна сумісність.

Звідси випливає важливий наслідок: мова програмування, якою реалізовано клієнт і сервер, не повинна впливати на їхню здатність взаємодіяти під час виконання.

[3]

### 1.2.4 Прозорість розташування

Компонентний застосунок містить кілька бінарних компонентів. Ці компоненти можуть перебувати в одному процесі, у різних процесах на одній машині або на різних машинах у мережі. З появою вебсервісів компоненти тепер також можуть бути розміщені через Інтернет.

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						13
Змін.	Арк.	№ докум.	Підпис	Дата		

Базова компонентна технологія має забезпечувати клієнту прозорість розташування, що дозволяє клієнтському коду не залежати від фактичного розташування об'єктів, якими він користується. Прозорість розташування означає, що в коді клієнта немає нічого, що вказує на місце виконання об'єктів. Той самий клієнтський код має бути здатен обробляти всі варіанти розташування об'єкта:

- у тому самому процесі;
- у різних процесах на тій самій машині;
- на різних машинах у тій самій локальній мережі;
- або навіть на різних машинах через Інтернет.

[3]

### 1.2.5 Керування одночасним доступом

Розробник компонента не може заздалегідь знати всі можливі способи його використання, особливо – чи будуть до нього одночасно звертатися кілька потоків. Тож слід заздалегідь підготуватись до використання компонента в багатопотоковому середовищі.

Одним із варіантів є реалізація механізму синхронізації всередині самого компонента. Однак такий підхід має два недоліки:

- Може призвести до взаємоблокування (deadlock) – за наявності власного синхронізаційного замка в кожного компонента застосунку, може виникнути ситуація, коли два компоненти з різних потоків намагаються звернутися один до одного.
- Неефективне використання системних ресурсів – якщо всі компоненти в застосунку будуть доступні лише одному потоку.

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						14
Змін.	Арк.	№ докум.	Підпис	Дата		

Щоб уникнути цих проблем, базова компонентна технологія повинна надавати сервіс керування одночасним доступом – спосіб, за допомогою якого компоненти можуть брати участь у загальній для всього застосунку системі синхронізації, навіть якщо самі компоненти були розроблені окремо.

Крім того, базова технологія має дозволяти як компонентам, так і клієнтам реалізовувати власні рішення синхронізації для більш точного контролю і кращої продуктивності.

[3]

### 1.2.6 Підтримка контролю версій

Компонентне програмування повинно дозволяти клієнтам і компонентам розвиватися незалежно один від одного. Розробники компонентів мають мати змогу випускати нові версії (або просто виправлення) існуючих компонентів без впливу на вже існуючі клієнтські застосунки. Розробники клієнтських застосунків мають мати можливість випускати нові версії своїх програм і очікувати їхню сумісність зі старішими версіями компонентів.

Базова компонентна технологія повинна підтримувати контроль версій, що дозволяє компоненту розвиватися в різних напрямках і дозволяє одночасно розміщувати на одній машині або навіть в одному клієнтському процесі кілька версій одного й того самого компонента. Технологія також має виявляти несумісності якомога раніше і повідомляти про них клієнта.

[3]

### 1.2.7 Безпека на основі компонентів

У компонентно-орієнтованому програмуванні компоненти розробляються окремо від клієнтських застосунків, які їх використовують. Розробники компонентів не можуть знати, як саме клієнтський застосунок або кінцевий

					КР. ІІ – 27.00.00.000 ПЗ	Арк.
						15
Змін.	Арк.	№ докум.	Підпис	Дата		

користувач скористається їхньою роботою. Аналогічно, клієнтський застосунок не може знати, чи взаємодіє він із зловмисним компонентом, який зловживає наданими клієнтом правами доступу. Крім того, навіть якщо і клієнт, і компонент не мають злих намірів, кінцевий користувач застосунку все одно може спробувати зламати систему або випадково завдати шкоди.

Щоб зменшити ці ризики, компонентна технологія має надавати безпекову інфраструктуру для роботи з такими ситуаціями, без жорсткого зв'язку між компонентами та клієнтськими застосунками. Вимоги до безпеки, політики та події (наприклад, додавання нових користувачів) є одними з найбільш мінливих аспектів життєвого циклу застосунку, і політики безпеки відрізняються в різних застосунках і для різних клієнтів.

Продуктивна компонентна технологія повинна забезпечувати мінімум безпекових політик у самих компонентах і мінімум безпекової логіки в коді. Вона також має дозволяти системним адміністраторам налаштовувати і керувати політикою безпеки застосунку без необхідності змінювати код.

[3]

## 1.3 Single Page Application

Односторінковий додаток (Single Page Application) – це веб-додаток, який завантажує лише один HTML-документ, а далі змінює вміст сторінки динамічно через JavaScript (наприклад, за допомогою Fetch API), коли потрібно показати інший контент.

Це дозволяє користувачам працювати з сайтом без повного перезавантаження сторінок, що покращує швидкість і робить взаємодію більш плавною та динамічною. Водночас таке рішення має і недоліки: складніше оптимізувати SEO, складніше підтримувати стан додатку, реалізувати навігацію і ефективний моніторинг продуктивності.

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						16
Змін.	Арк.	№ докум.	Підпис	Дата		



[4]

## 1.4 Переваги SPA

SPA-рішення мають багато переваг як з точки зору клієнтського досвіду, так і з точки зору внутрішньої розробки. Покращена продуктивність додатків, узгодженість, скорочення часу розробки та зниження витрат на інфраструктуру дозволяють командам розробників працювати більш ефективно.

### Одноразове завантаження

Після першого завантаження сторінки сервер більше не надсилає HTML. Користувач отримує “оболонку” сторінки, а браузер відображає інтерфейс. Далі, при навігації, SPA запитує лише дані, а браузер оновлює потрібні частини сторінки без повного перезавантаження. Сервер не витрачає ресурси на рендеринг HTML під час кожного переходу, що знижує навантаження на сервер. Це дуже зручно для сайтів з великою кількістю переходів і шаблонів.

### Швидка розробка фронтенду

Разом із швидшим часом виконання, SPA також дозволяють розробникам швидше створювати фронтед сайтів. Це пов’язано з роз’єднаною архітектурою SPA, а саме відокремленням бекенд-сервісів від фронтед-відображення.

Багато критично важливих бізнес-функцій на бекенді змінюються досить рідко на відміну від їх відображення. Відокремлюючи логіку і дані бекенду від того, як вони презентуються, вони перетворюються на сервіси (див. розділ 1.2.1), розробники можуть створювати багато різних фронтед-рішень для показу і використання цих сервісів.

З такою роз’єднаною структурою розробники можуть створювати, розгортати і експериментувати з фронтедом повністю незалежно від технології, що

					КР. ІІ – 27.00.00.000 ПЗ	Арк.
						17
Змін.	Арк.	№ докум.	Підпис	Дата		

лежить в основі бекенду. Вони проєктують, як саме хочуть, щоб виглядав і відчувався користувацький досвід, а потім отримують контент, дані і функціональність через ці сервіси

## Покращений користувацький досвід

Чим більше функціональностей створюється у вигляді модульних сервісів (архітектура мікросервісів), які можна оновлювати незалежно, тим легше експериментувати з тим, як вони відображаються та використовуються. SPA-застосунки, можуть без проблем працювати з бекенд-сервісами, розробленими різними мовами.

Фреймворки SPA чудово підходять для тестування таких сервісів з метою створення привабливого, динамічного та навіть анімованого користувацького досвіду.

[5]

## 1.5 Недоліки SPA

Будь-яка технологія, включаючи має свої недоліки. У випадку зі SPA це перш за все проблеми пов'язані із сприйняттям SPA-застосунків браузерами, які повинні сприймати різні стани застосунку, як окремі вебсторінки, що може не завжди здійснюватися коректно.

## Проблеми з SEO

SPA традиційно стикаються з труднощами пошукової оптимізації (SEO). Оскільки більшість контенту завантажується динамічно за допомогою JavaScript, пошуковим системам може бути складно індексувати такий контент, що може негативно вплинути на рейтинг сайту. Втім, існують обхідні шляхи й техніки для покращення SEO в SPA.

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						18
Змін.	Арк.	№ докум.	Підпис	Дата		

## Час початкового завантаження

Хоча SPA забезпечують швидкий та чутливий інтерфейс після завантаження, сам цей процес може тривати довше, ніж у традиційних сайтів. Це пов'язано з тим, що вся програма з усіма ресурсами завантажується одразу. Це може бути проблемою для користувачів із повільним Інтернет-з'єднанням.

## Історія браузера і закладки

SPA використовують маршрутизацію на стороні клієнта, тому історія переглядів і закладки браузера можуть працювати не так, як очікується. Користувачам може бути складно ділитися посиланнями або використовувати історію переглядів.

## Складна розробка

Розробка SPA зазвичай складніша, ніж створення традиційного сайту. Потрібно управляти маршрутизацією на клієнті, станом додатка та асинхронним отриманням даних. Це ускладнює розробку й може спричинити триваліший процес налагодження.

## Проблеми безпеки

SPA можуть бути вразливими до деяких типів атак, наприклад, міжсайтового скриптингу (XSS – Cross-Site Scripting), якщо не вжити належних заходів безпеки. Розробникам слід особливо уважно ставитися до захисту таких додатків від потенційних загроз.

[6]

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						19
Змін.	Арк.	№ докум.	Підпис	Дата		

## 1.6 React

React (також React.js або ReactJS) – це фронтенд JavaScript-бібліотека з відкритим кодом, яка має на меті зробити створення інтерфейсів на основі компонентів більш «плавним». Її підтримують компанія Meta (раніше Facebook) та спільнота окремих розробників і компаній.

React можна використовувати для розробки односторінкових, мобільних або серверно-рендерених застосунків за допомогою таких фреймворків, як Next.js і Remix. Оскільки React відповідає лише за інтерфейс користувача та рендеринг компонентів у DOM, застосунки на React зазвичай покладаються на сторонні бібліотеки для маршрутизації та іншої клієнтської функціональності.

Таким чином React активно використовується для створення Single Page Application в рамках концепції компонентного програмування. Варто зазначити, що ключовою перевагою React є повторний рендеринг лише тих частин сторінки, які змінилися, а отже уникання непотрібного оновлення незмінних DOM-елементів.

[7]

### 1.6.1 Особливості React

#### Декларативність

React дотримується декларативної парадигми програмування. Розробники створюють views для кожного стану застосунку, а React оновлює та рендерить компоненти, коли змінюються дані. Це контрастує з імперативним стилем програмування.

					КР. ІІ – 27.00.00.000 ПЗ	Арк.
						20
Змін.	Арк.	№ докум.	Підпис	Дата		

## Компоненти

Код React складається з сутностей, які називаються компонентами (див. розділ 1.1). Ці компоненти модульні й можуть бути повторно використані. Застосунки на React зазвичай складаються з багатьох рівнів компонентів. Вони рендеряться в кореневий елемент DOM за допомогою бібліотеки React DOM. Під час рендерингу значення передаються між компонентами через props (властивості). Внутрішні значення компонента називаються його станом (state). Два основних способи оголошення компонентів у React – це функціональні та класові компоненти.

## Функціональні компоненти

Функціональні компоненти оголошуються за допомогою функції (звичайної або стрілочної), яка приймає один аргумент props і повертає JSX. Починаючи з React v16.8, функціональні компоненти можуть використовувати стан завдяки хуку (детальніше про React Hooks в наступному розділі) useState.

## React Hooks

16 лютого 2019 року був випущений React 16.8, який представив React Hooks – функції, що дозволяють "підключатися" до стану та життєвого циклу з функціональних компонентів. Хуки не працюють у класах – вони дозволяють користуватись можливостями React без класів.

### Правила хуків

1. Хуки викликаються лише на верхньому рівні функцій – не в циклах/умова.
2. Хуки викликаються тільки в React-функціях – не в звичайних JavaScript-функціях.

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						21
Змін.	Арк.	№ докум.	Підпис	Дата		

## Серверні компоненти (Server Components)

Це функціональні компоненти, які виконуються виключно на сервері. Вперше ідею представили в доповіді "Data Fetching with Server Components". На відміну від Server Side Rendering, вони не передають JavaScript-код клієнту. Вони не мають доступу до хуків, але можуть бути асинхронними, що дозволяє виконувати запити прямо в них.

## Класові компоненти

Оголошуються за допомогою класів ES6. Вони мають ту ж логіку, що й функціональні компоненти, але для роботи зі станом і життєвим циклом використовують методи життєвого циклу з базового класу `React.Component`.

Запровадження хуків у React 16.8 дозволило керувати станом та життєвим циклом у функціональних компонентах, зменшивши потребу в класових компонентах. Це відповідає загальній тенденції до функціонального програмування та модульного дизайну.

## Маршрутизація (Routing)

React не має вбудованої підтримки маршрутизації. Це бібліотека для побудови інтерфейсів, а не повноцінний фреймворк. Для маршрутизації використовують сторонні бібліотеки, які дозволяють створювати маршрути, керувати навігацією й обробляти зміни в URL.

## Віртуальний DOM

React використовує віртуальний DOM – кеш структури DOM у пам'яті. React обчислює різницю між попереднім і новим станом та ефективно оновлює DOM браузера. Цей процес називається "узгодженням" (reconciliation). Це дозволяє писати код так, ніби вся сторінка рендериться при кожній зміні, але React оновлює лише ті компоненти, які насправді змінилися, що значно підвищує продуктивність.

[7]

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						22
Змін.	Арк.	№ докум.	Підпис	Дата		

## 2. Практична частина

### 2.1 Актуальність теми

Актуальність теми онлайн книгарні в сучасних умовах зумовлена стрімким розвитком книжкового бізнесу в Україні та глобальними змінами в способах взаємодії споживачів із інформаційними продуктами. Зростаючий попит на цифрові сервіси, включно з інтернет-торгівлею, сприяє активному переходу традиційних книжкових книгарень в онлайн-простір. Це дозволяє розширити аудиторію покупців, підвищити доступність продукції та оптимізувати процеси замовлення і доставки.

Розвиток книжкового бізнесу в Україні на сьогодні відзначається зростаючим інтересом до онлайн-торгівлі, що зумовлено не лише загальними трендами цифровізації, а й значним зростанням як попиту так і пропозиції на україномовну літературу, а отже значним розширенням асортименту, який фізично важко тримати в традиційних магазинах. Інтернет-платформи дозволяють видавцям і ретейлерам більш гнучко реагувати на попит, пропонуючи користувачам широкий вибір книг із можливістю швидкого пошуку, порівняння та замовлення.

Вебсайт у форматі Single Page Application (SPA) є особливо ефективним інструментом для реалізації онлайн книгарні, як і будь-якого онлайн магазину, оскільки забезпечує швидку взаємодію користувача з інтерфейсом, оновлюючи контент сторінки без необхідності постійного перезавантаження. У цьому контексті бібліотека React виступає ідеальним технічним вибором, оскільки надає можливість швидкого та ефективного оновлення DOM-сторінки, що особливо важливо при реалізації таких функцій, як динамічний каталог, фільтри, кошик та інші інтерактивні елементи, характерні для онлайн-магазинів загалом та книгарень зокрема. Крім того компонентна архітектурі

					КР. ІІ – 27.00.00.000 ПЗ	Арк.
						23
Змін.	Арк.	№ докум.	Підпис	Дата		

React полегшує розробку, масштабування та підтримку складних інтерфейсів користувача.

Таким чином, впровадження сучасних технологій веб-розробки у сфері книжкового бізнесу сприяє підвищенню конкурентоспроможності українських книгарень, розширенню їхнього функціоналу та покращенню сервісу для кінцевих користувачів.

## 2.2 Аналіз вимог

Перед початком розробки веб-застосунку доцільно провести початкове моделювання, що дає змогу чітко визначити функціональні вимоги до системи. На цьому етапі здійснюється ідентифікація основних категорій користувачів, які взаємодіятимуть із системою. Для кожної категорії формують user stories, тобто короткі описи їхніх цілей, що відображають очікувану користь від взаємодії з вебсайтом. Виходячи з цих цілей, складаються випадки використання (use cases) формалізовані сценарії, які покроково описують взаємодію користувача з системою для досягнення конкретного результату.

### 2.2.1 Визначення користувачів системи

У межах моделювання функціональності вебзастосунку електронної книгарні доцільно виокремити дві ключові категорії користувачів:

- **Покупець** – особа, яка взаємодіє із сайтом з метою перегляду, пошуку та замовлення книжкової продукції, а також ознайомлення з інформаційними розділами ресурсу;
- **Адміністратор** – особа, наділена правами керування контентом, яка відповідає за наповнення каталогу, редагування даних про книжки та підтримку інформаційних блоків.

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						24
Змін.	Арк.	№ докум.	Підпис	Дата		



## 2.2.2 Цілі користувачів

З огляду на ролі кожного типу користувача, формулюються такі цілі:

- Загальні цілі:
  - Виконання входу до облікового запису;
  - Реєстрація нового користувача.
- Цілі покупця:
  - Пошук книжки в каталозі;
  - Оформлення замовлення на книжку;
  - Ознайомлення з додатковими інформаційними розділами.
- Цілі адміністратора:
  - Додавання нових книжок до каталогу;
  - Редагування наявної інформації про книжки.

## 2.2.3 Основні сценарії використання (use cases)

### 1. Вхід / Реєстрація

- Заповнення відповідної форми введення даних;
- Перевірка правильності введених облікових даних;
- У разі реєстрації – збереження нового користувача в системі;
- Після успішної авторизації – надання доступу до функціональних можливостей згідно з роллю.

### 2. Пошук книжки

- Перегляд добірок книжок у рекомендованих тематичних розділах (через слайдер/карусель);
- Введення пошукового запиту у відповідне поле;
- Вибір фільтрів за категоріями (наприклад, діапазон ціни, жанр, піджанр);
- Виведення результатів пошуку у вигляді карток у галереї;

					КР. ІІ – 27.00.00.000 ПЗ	Арк.
						25
Змін.	Арк.	№ докум.	Підпис	Дата		

- Перегляд детальної інформації про конкретну книжку на окремій сторінці.

### 3. Оформлення замовлення

- Додавання книжки до кошика з картки або сторінки книжки;
- Перехід до форми оформлення замовлення в межах кошика;
- Коригування замовлених книг (вилучення/зміна кількості);
- Підтвердження замовлення (та перевірка правильності введених даних)

### 4. Ознайомлення з додатковою інформацією

- Перегляд повідомлень та новин у слайдері оголошень;
- Перехід за посиланнями з слайдера на відповідні інформаційні сторінки;
- Ознайомлення з відомостями про діяльність магазину в розділі «Про нас»;
- Використання інформаційних посилань у футері.

### 5. Додавання / редагування книжки (для адміністратора)

- Заповнення полів введення текстових даних про книжку (назва, автор, опис тощо);
- Збереження змін і оновлення інформації в базі даних.

## 2.3 Розробка компонентів

У контексті реалізації вебзастосунку електронної книгарні, розробка інтерфейсу користувача здійснюється відповідно до принципів компонентного програмування з використанням бібліотеки React. Такий підхід передбачає декомпозицію інтерфейсу на низку самостійних, повторно використовуваних компонентів, кожен з яких відповідає за окрему функціональність або фрагмент представлення даних. Виходячи зі сформульованих сценаріїв використання, було здійснено попередній розподіл функціональності між

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						26
Змін.	Арк.	№ докум.	Підпис	Дата		

відповідними компонентами, визначено сторінки, на яких вони будуть інтегровані, а також окреслено контексти для централізованого зберігання та передачі стану. Контексти дозволяють уникнути надлишкової передачі даних через ієрархію компонентів, забезпечуючи зручну й ефективну організацію взаємодії між окремими частинами застосунку.

## 2.3.1 Проектування компонентів відповідно до цілей користувачів

### 1. Вхід / Реєстрація

- Форма входу (LogInForm)
- Форма реєстрації (SignUpForm)

### 2. Пошук книжки

- Поле для пошуку (Search)
- Панель фільтрів (FilterBar)
- Картка з короткими відомостями про книгу (BookCard)
- Галерея для загального перегляду каталогу книг (WideGallery)
- Однорядкова галерея-слайдер із добіркою книг (NarrowGallery)
- Інформація про книгу (BookData)

### 3. Оформлення замовлення

- Кнопки для додавання в кошик (PurchaseButton)
- Іконка корзини з інформацією про кількість книг (CartIcon)
- Книги в корзині з можливістю редагування (WideCard, BookCard)
- Підтвердження замовлення (OrderForm)

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						27
Змін.	Арк.	№ докум.	Підпис	Дата		

#### 4. Ознайомлення з додатковою інформацією

- Слайдер з повідомленнями та новинами (UserCarousel)
- Футер з інформаційними посиланнями (Footer)
- Головне меню із дублікатом цих посилань (Menu)
- Різні новинні/інформаційні сторінки в форматі статей (Article)

#### 5. Додавання / редагування книжки (для адміністратора)

- Форма для редагування даних про книгу (BookEditor)

### 2.3.2 Структура сторінок та компонентів

#### 1. Головна сторінка

##### Призначення:

Містить основну інформацію, необхідну клієнтам – слайдер із новинами, добірки книг (рекомендовані, популярні, нові), а також стандартні елементи навігації – головне меню та футер.

##### Використовувані контексти:

- **BooksContext** – забезпечує доступ до каталогу книжок, що відображаються в добірках;
- **UsersContext** – містить інформацію про зареєстрованих користувачів;
- **UserContext** – зберігає стан поточного користувача (авторизація, роль);
- **CartContext** – керує станом кошика користувача;
- **GenresContext** – надає інформацію про жанри та піджанри книг для формування добірок та меню.

##### Компоненти:

					КР. III – 27.00.00.000 ПЗ	Арк.
						28
Змін.	Арк.	№ докум.	Підпис	Дата		

- **UserCarousel** – слайдер, який демонструє повідомлення, новини або акції, що динамічно завантажуються із сервера, забезпечуючи актуальну інформацію;
- **NarrowGallery** – однорядкова галерея-слайдер, яка відображає вибіркові книги з каталогу;
- **OrderBookCard** (дочірній від NarrowGallery) – картка книги із кнопками для додавання до кошика або, у разі адміністратора, для редагування (компонент **AdminPurchaseButtons**);
- **Menu** – головне меню, що містить навігаційні посилання на різні сторінки, а також компоненти:
  - **CartIcon** – іконка корзини з відображенням поточної кількості товарів;
  - **UserMenu** – панель для входу, реєстрації або відображення імені користувача із можливістю виходу;
  - **GenreDropdown** – випадаюче меню з жанрами і піджанрами, вибір яких спрямовує користувача на сторінку каталогу із застосуванням відповідних фільтрів;
- **Footer** – футер, який дублює частину посилань головного меню, а також містить контакти та набір логотипів довіри (компонент **TrustLogo**).

## 2. Сторінки входу та реєстрації

### Призначення:

Реалізовані одним динамічним компонентом, який залежно від стану відображає форму входу або реєстрації.

### Використовувані контексти:

					КР. III – 27.00.00.000 ПЗ	Арк.
						29
Змін.	Арк.	№ докум.	Підпис	Дата		

- **UsersContext** – для перевірки існування користувача та додавання нових записів;
- **UserContext** – для збереження стану авторизації.

#### Компоненти:

- **LogInForm** – форма для введення облікових даних та авторизації;
- **SignUpForm** – форма реєстрації нового користувача із валідацією введених даних;
- Обидві форми базуються на загальному компоненті **CustomForm**, який забезпечує уніфіковану обробку вводу, валідацію та подачу даних.

### 3. Сторінка каталогу книг

#### Призначення:

Забезпечує перегляд, пошук і фільтрацію книжок із каталогу.

#### Використовувані контексти:

- **BooksContext** – каталог книжок;
- **UsersContext** та **UserContext** – для персоналізації;
- **CartContext** – для управління кошиком;
- **GenresContext** – інформація про жанри;
- **AuthorsContext**, **CoversContext**, **PublishersContext** – додаткові контексти, які надають інформацію про авторів, обкладинки та видавництва відповідно.

#### Компоненти:

- **Search** – поле введення пошукового запиту;

					КР. III – 27.00.00.000 ПЗ	Арк.
						30
Змін.	Арк.	№ докум.	Підпис	Дата		

- **FilterBar** – панель фільтрів, реалізована як OffCanvas, що складається з кількох підкомпонентів:
  - **FilterSection** – контейнер для груп фільтрів, який дозволяє згорнути або розгорнути вміст;
  - **ToggleList** – список категорій з перемикачами (наприклад, видавництва);
  - **PriceRange** – два поля введення для задання діапазону цін із контролем коректності введення;
  - **GenreList** – містить **GenreAccordion** для вибору жанрів, а також набір кнопок для видалення вибраних жанрів;
- **WideGallery** – галерея для відображення результатів пошуку і фільтрації з частковим завантаженням (підвантаженням) книг.

#### 4. Сторінка даних про книгу

##### Призначення:

Відображення детальної інформації про конкретну книгу.

##### Використовувані контексти:

- **UserContext** – стан поточного користувача;
- **BookContext** – інформація про книгу, що переглядається.

##### Компоненти:

- Набір заголовків і текстових блоків для відображення метаданих та опису книги;
- **AdminPurchaseButtons** – кнопки для додавання до кошика або редагування книги (для адміністратора);
- **NarrowGallery** – галерея рекомендацій з подібних книг.

					КР. III – 27.00.00.000 ПЗ	Арк.
						31
Змін.	Арк.	№ докум.	Підпис	Дата		

## 5. Сторінка редагування даних про книгу

### Призначення:

Дозволяє вводити або змінювати інформацію про книгу.

### Використовувані контексти:

- **BookContext** – забезпечує початкові дані для форми і збереження змін.

### Компоненти:

- **CustomForm** – спеціалізована форма для введення даних про книгу, що підтримує валідацію та ініціалізацію полів початковими значеннями у разі редагування існуючої книги.

## 6. Інформаційні та новинні сторінки

### Призначення:

Створені для відображення статей, новин або інформації про магазин.

### Компоненти:

- **Article** – універсальний компонент, який формує сторінки на основі набору пропсів із блоками різного типу (зображення, заголовки, текст), що дозволяє створювати однотипні статичні або динамічні матеріали.

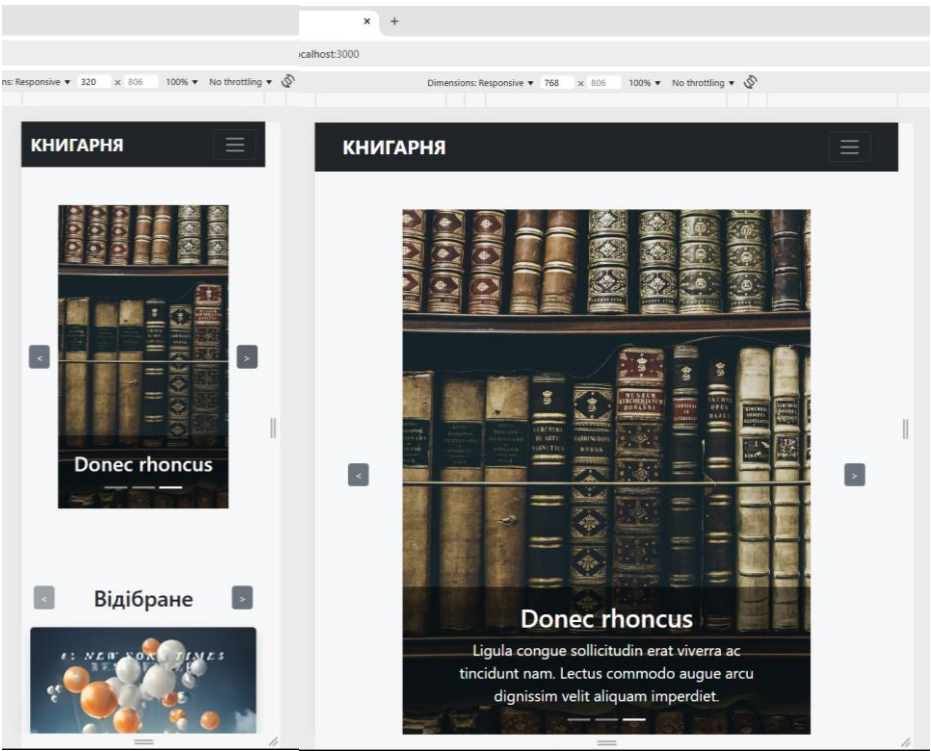
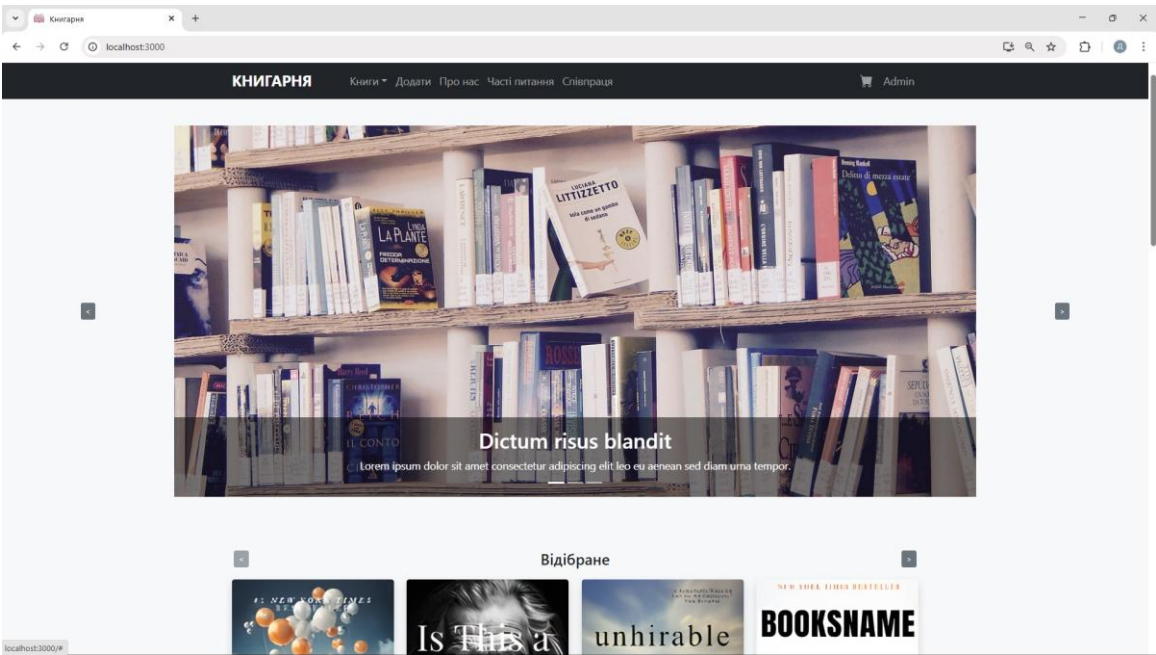
## 2.4 Демонстрація проекту

У цьому розділі представлено результати розробки вебзастосунку у вигляді скріншотів ключових сторінок системи. Наведені ілюстрації роботи інтерфейсу, що демонструють реалізацію різних сценаріїв використання відповідно до потреб користувачів. Окрему увагу приділено адаптивності дизайну, що забезпечує коректне та зручне відображення контенту на мобільних пристроях різних розмірів, сприяючи комфортній взаємодії незалежно від платформи користувача.

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						32
Змін.	Арк.	№ докум.	Підпис	Дата		

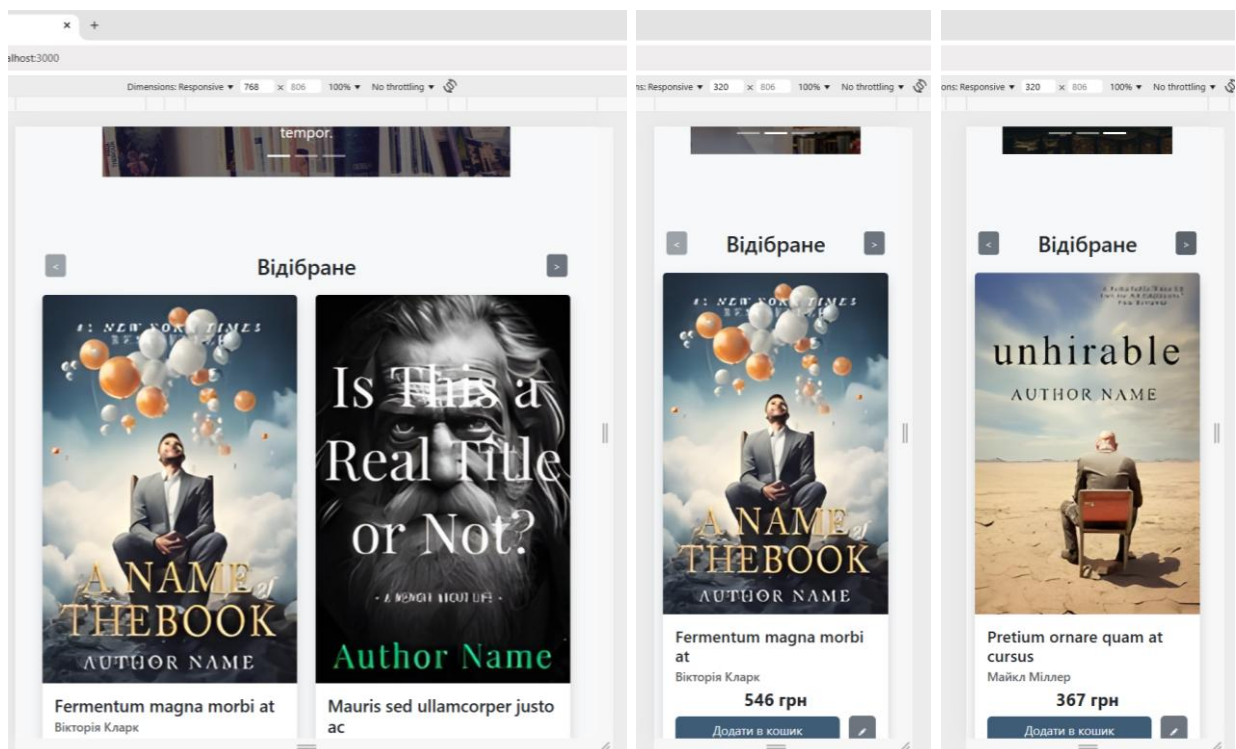
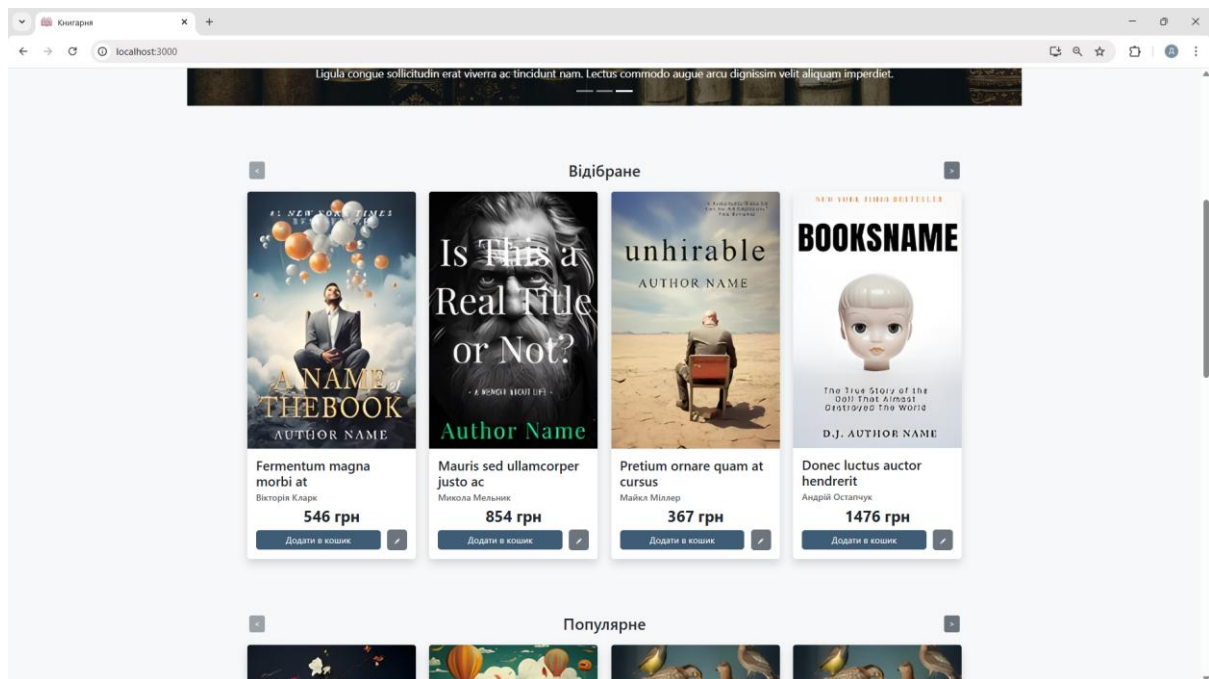


# 2.4.1 Головна сторінка



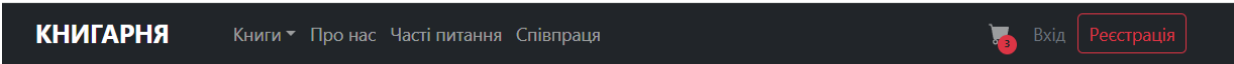
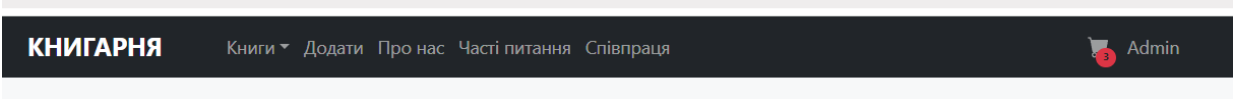
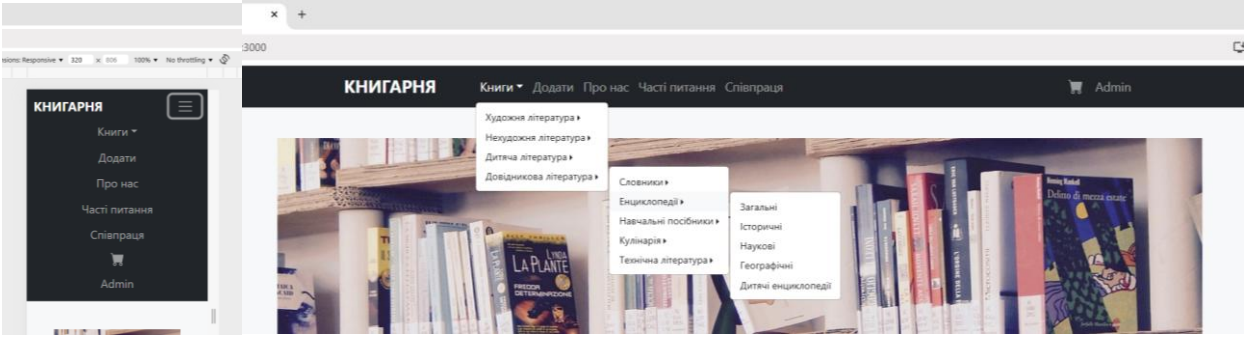
На головній сторінці ми сторінці можна побачити слайдер/карусель з новинами. Карусель повністю адаптивна та приховує довгий опис на особливо малих екранах.

					КР. III – 27.00.00.000 ПЗ	Арк.
						33
Змін.	Арк.	№ докум.	Підпис	Дата		

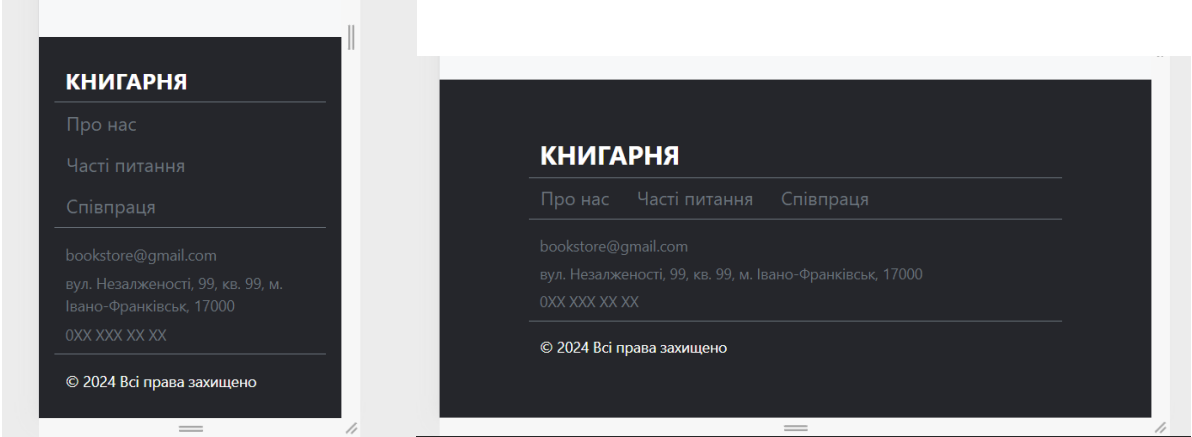
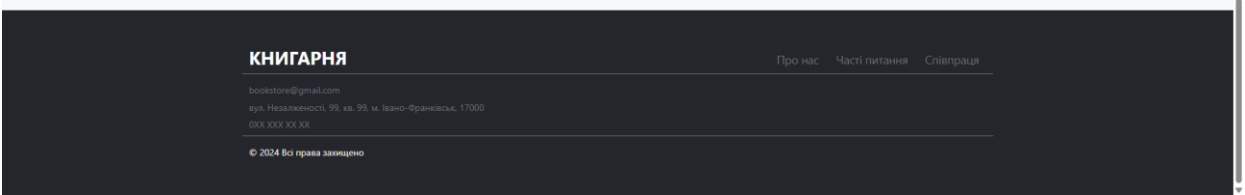


Нижче можна побачити різноманітні добірки книг подані у побінні до каруселі манері, елемент так само адаптивний, оскільки реалізований на основі Bootstrap Grid.

					КР. III – 27.00.00.000 ПЗ	Арк.
						34
Змін.	Арк.	№ докум.	Підпис	Дата		



Так само варто відзначити головне меню, що містить посилання, випадające меню з ієрархією жанрів книг, іконку корзини, справа від якої показується кількість книг в ній та набір кнопок для входу/реєстрації користувача. Меню повністю адаптивне завдяки Bootstrap.



Футер дублює частину даних із головного меню, а також містить контакти і TrustLogo. Варто зазначити, що адаптування цього компоненту було складнішим завданням через відсутність зручних Bootstrap-заготовок.

					КР. ІІ – 27.00.00.000 ПЗ	Арк.
						35
Змін.	Арк.	№ докум.	Підпис	Дата		



# 2.4.2 Інформаційні сторінки

Книги ▾ Про нас Часті питання Співпраця

### Mus donec rhoncus

Lorem ipsum dolor sit amet consectetur adipiscing elit. Leo eu aenean sed diam urna tempor pulvinar. Semper vel class aptent taciti sociosqu ad litora. Mus donec rhoncus eros lobortis nulla molestie mattis. Blandit quis suspendisse aliquet nisi sodales consequat magna. Ac tincidunt nam porta elementum a enim euismod.

Cras eleifend turpis fames primis vulputate ornare sagittis. Netus suscipit auctor curabitur facilisi cubilia curae hac. Sem placerat in id cursus mi pretium tellus. Egestas iaculis massa nisl malesuada lacinia integer nunc. Orci varius natoque penatibus et magnis dis parturient. Non purus est efficitur laoreet mauris pharetra vestibulum.

### Cras eleifend turpis

Sed diam urna tempor pulvinar vivamus fringilla lacus. Aptent taciti sociosqu ad litora torquent per conubia.

Eros lobortis nulla molestie mattis scelerisque maximus eget.

Fames primis vulputate ornare sagittis vehicula praesent dui. Curabitur facilisi cubilia curae hac habitasse platea dictumst.


### Finibus facilisis

Orci varius natoque penatibus et magnis dis parturient. Non purus est efficitur laoreet mauris pharetra vestibulum. Finibus facilisis dapibus etiam interdum tortor ligula congue. Dignissim velit aliquam imperdiet mollis nullam volutpat porttitor. Proin libero feugiat tristique accumsan maecenas potenti ultricies. Sit amet consectetur adipiscing elit quisque faucibus ex. Sed diam urna tempor pulvinar vivamus fringilla lacus.

### Porta elementum

Mus donec rhoncus eros lobortis nulla molestie mattis. Blandit quis suspendisse aliquet nisi sodales consequat magna. Ac tincidunt nam porta elementum a enim euismod. Cras eleifend turpis fames primis vulputate ornare sagittis.

Книги ▾ Про нас Часті питання Співпраця



### Primis vulputate


Lorem ipsum dolor sit amet consectetur adipiscing elit. Leo eu aenean sed diam urna tempor pulvinar. Semper vel class aptent taciti sociosqu ad litora. Mus donec rhoncus eros lobortis nulla molestie mattis. Blandit quis suspendisse aliquet nisi sodales consequat magna. Ac tincidunt nam porta elementum a enim euismod.

Cras eleifend turpis fames primis vulputate ornare sagittis. Netus suscipit auctor curabitur facilisi cubilia curae hac. Sem placerat in id cursus mi pretium tellus. Egestas iaculis massa nisl malesuada lacinia integer nunc. Orci varius natoque penatibus et magnis dis parturient. Non purus est efficitur laoreet mauris pharetra vestibulum.

Proin libero feugiat tristique accumsan maecenas potenti ultricies. Sit amet consectetur adipiscing elit.

Dimensions: Responsive 425 x 806 100% No throttling localhost:3000/third-news

КНИГАРНЯ




### Donec rhoncus

Sit amet consectetur adipiscing elit quisque faucibus ex. Sed diam urna tempor pulvinar vivamus fringilla lacus. Aptent taciti sociosqu ad litora torquent per conubia. Eros lobortis nulla molestie mattis scelerisque maximus eget.

Dimensions: Responsive 768 x 806 100% No throttling

Orci varius natoque penatibus et magnis dis parturient. Non purus est efficitur laoreet mauris pharetra vestibulum.

Porta elementum a enim euismod quam justo lectus. Fames primis vulputate ornare sagittis vehicula praesent dui. Curabitur facilisi cubilia curae hac habitasse platea dictumst. Id cursus mi pretium tellus dui convallis tempus. Nisl malesuada lacinia integer nunc posuere ut hendrerit.



### Non purus est efficitur laoreet

Orci varius natoque penatibus et magnis dis parturient. Non purus est efficitur

Інформаційні сторінки всі створені за допомогою одного компонента, який дозволяє динамічно генерувати їх із набору пропсів, що суттєво прискорює процес розробки. Адаптування цих сторінок не склало жодних труднощів

					КР. ІІІ – 27.00.00.000 ПЗ	Арк.
						36
Змін.	Арк.	№ докум.	Підпис	Дата		

## 2.4.3 Авторизація

**Вхід**

Електронна пошта або номер телефону

bookstoreadmin@gmail.com

Пароль

.....

Реєстрація

Підтвердити

**Вхід**

Електронна пошта або номер телефону

bookstoreadmin@gmail.com

Пароль

.....

Неправильний пароль

Реєстрація

Підтвердити

**Реєстрація**

Ім'я

.....

Електронна пошта

.....

Номер телефону

bookstoreadmin@gmail.com

Пароль

.....

Повторити пароль

.....

Дата народження

дд.мм.рррр

Вхід

Підтвердити

**Реєстрація**

Ім'я

Денис

Ім'я повинно містити лише латинські літери

Електронна пошта

bookstoreadmin@gmail.com

Номер телефону

+380662289614

Пароль

.....

Пароль повинен містити лише латинські літери та цифри

Повторити пароль

.....

Паролі не збігаються

Дата народження

13.06.2005

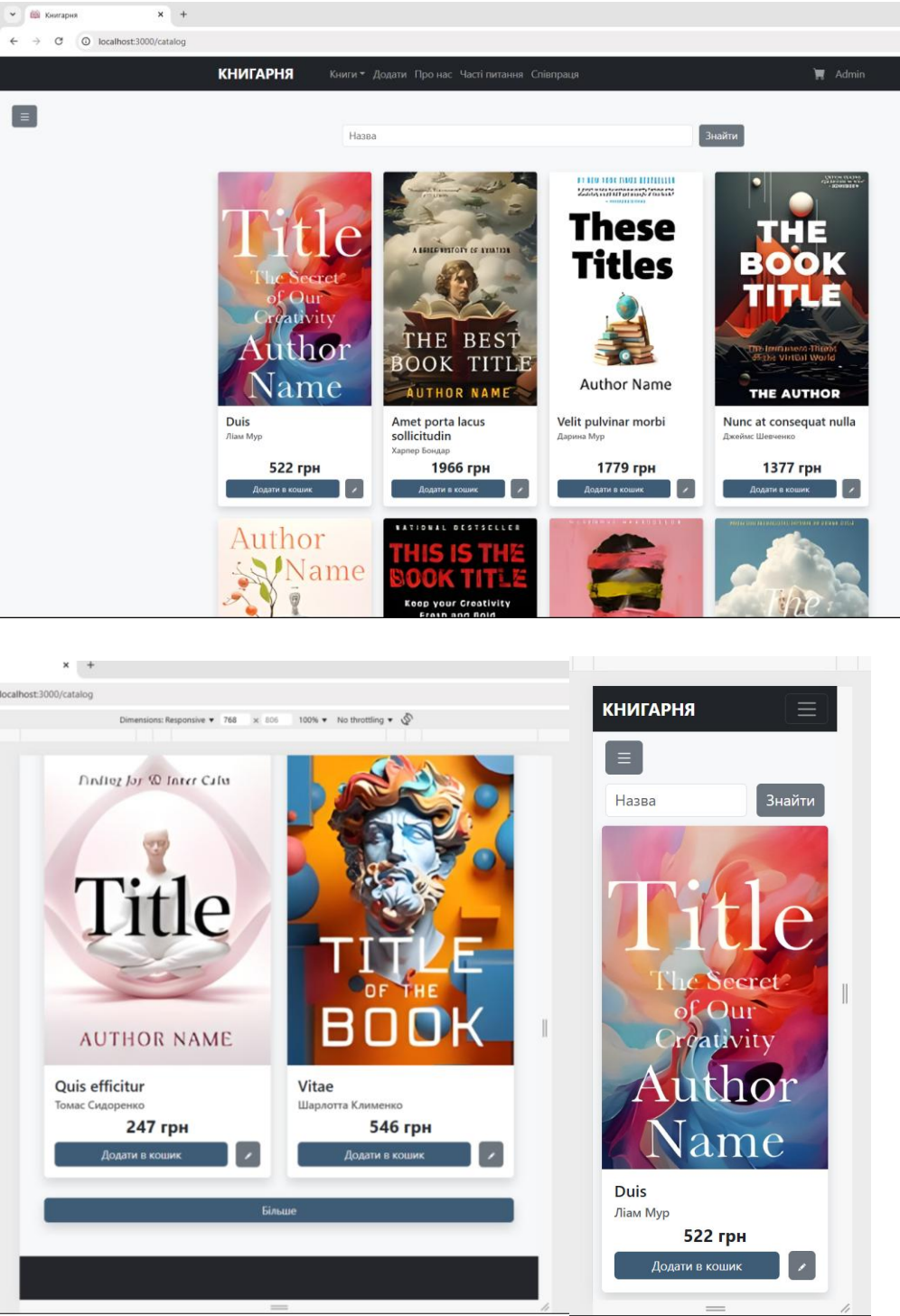
Вхід

Підтвердити

Форми для входу та реєстрації, створені за одним шаблоном, здійснюють обробку можливих помилок при введенні користувача та зручно адаптуються під різну ширину екрану.

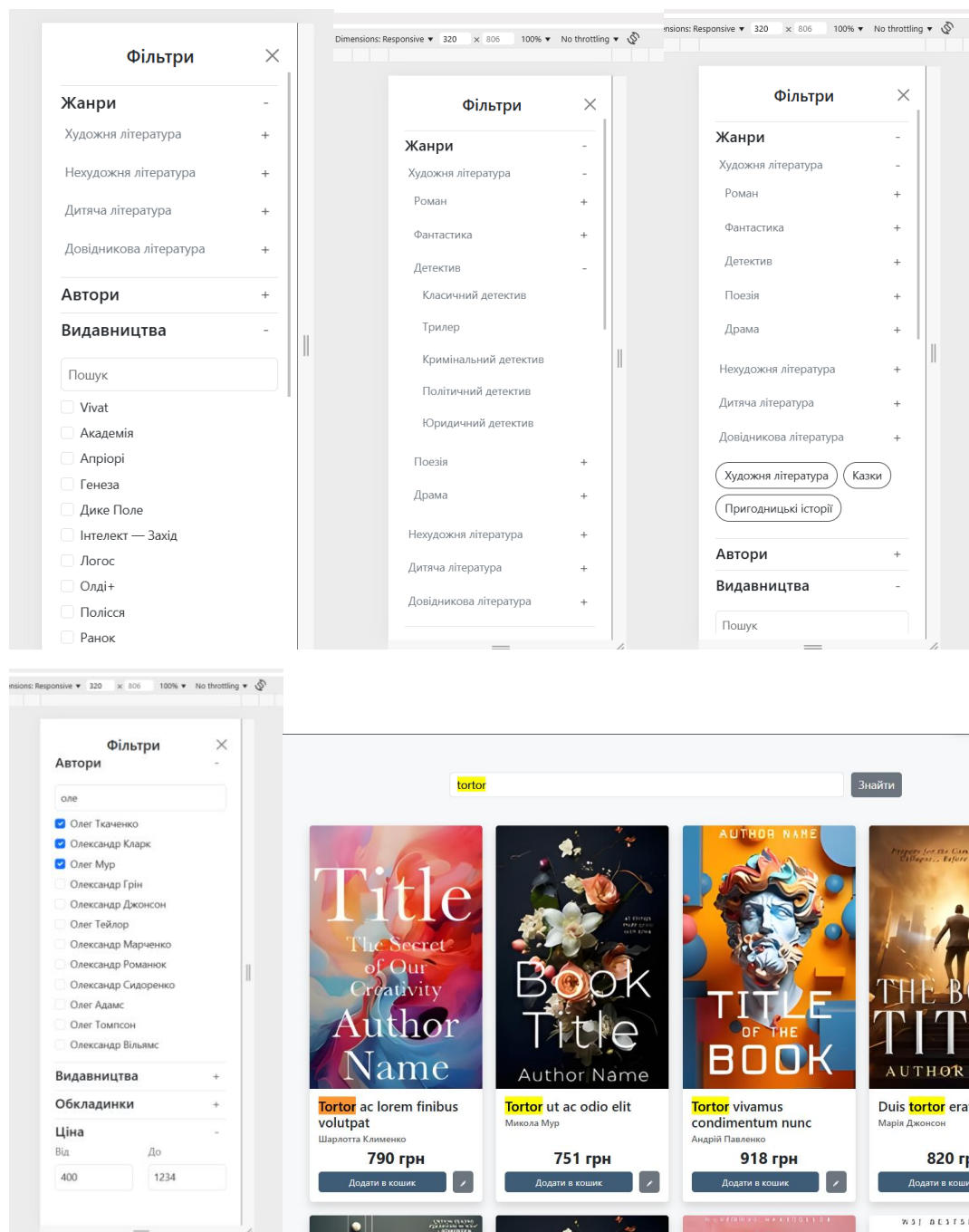
					КР. ІІ – 27.00.00.000 ПЗ	Арк.
						37
Змін.	Арк.	№ докум.	Підпис	Дата		

## 2.4.4 Каталог товарів



Сторінка каталогу товарів містить набір фільтрів, поле для пошуку та галерею з результатами (книгами доступними для придбання). Галерея повністю адаптивна, а також завантажує книги частинами, дозволяючи користувачу самому вибрати скільки він хоче переглянути.

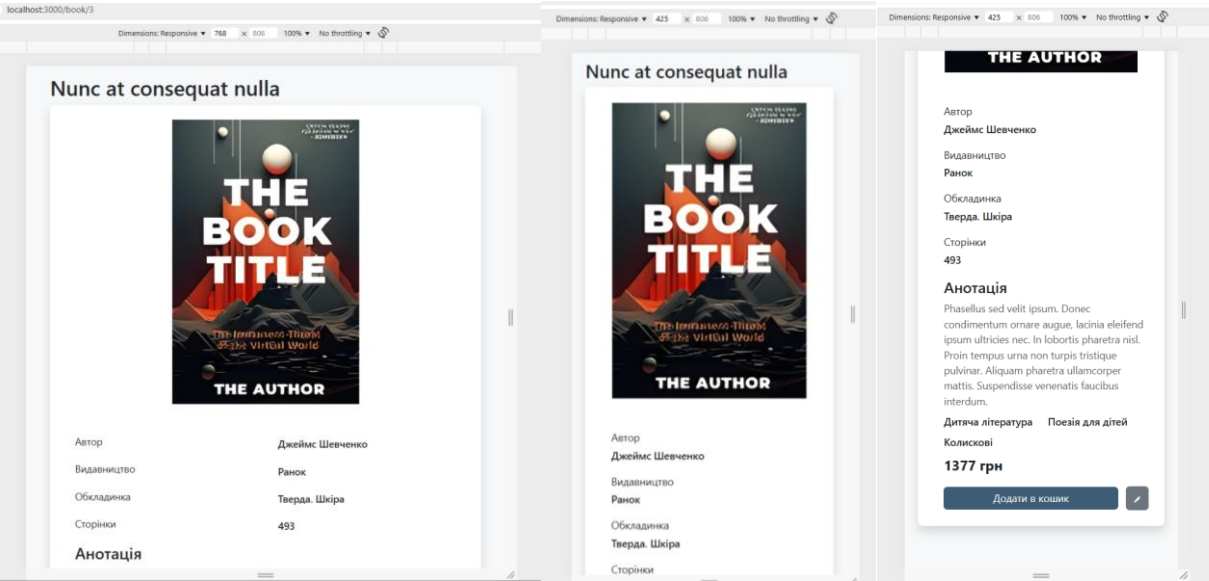
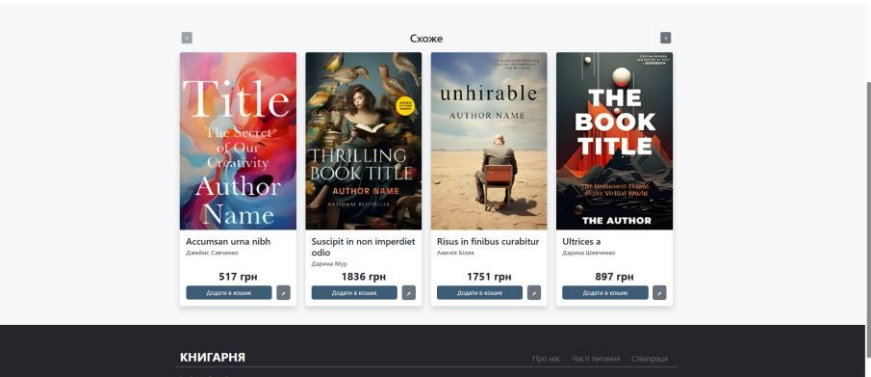
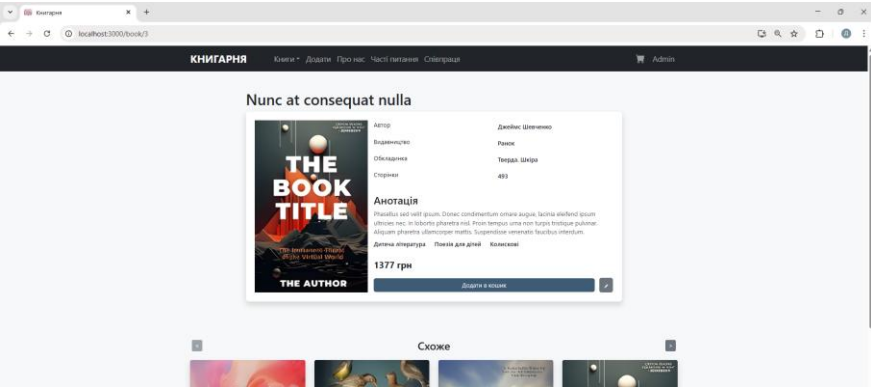
					КР. ІІ – 27.00.00.000 ПЗ	Арк.
						38
Змін.	Арк.	№ докум.	Підпис	Дата		



Поля з фільтрами дозволяє обирати з різних категорій (автори, видавництва тощо). Обирати діапазон цін, а також жанри, реалізація цього фільтру особливо цікава. Поля для пошуку так само працює (шукаючи слова у назві) і поєднується і фільтрами для ефективнішого пошуку.

					КР. III – 27.00.00.000 ПЗ	Арк.
						39
Змін.	Арк.	№ докум.	Підпис	Дата		

2.4.5 Сторінка опису книги

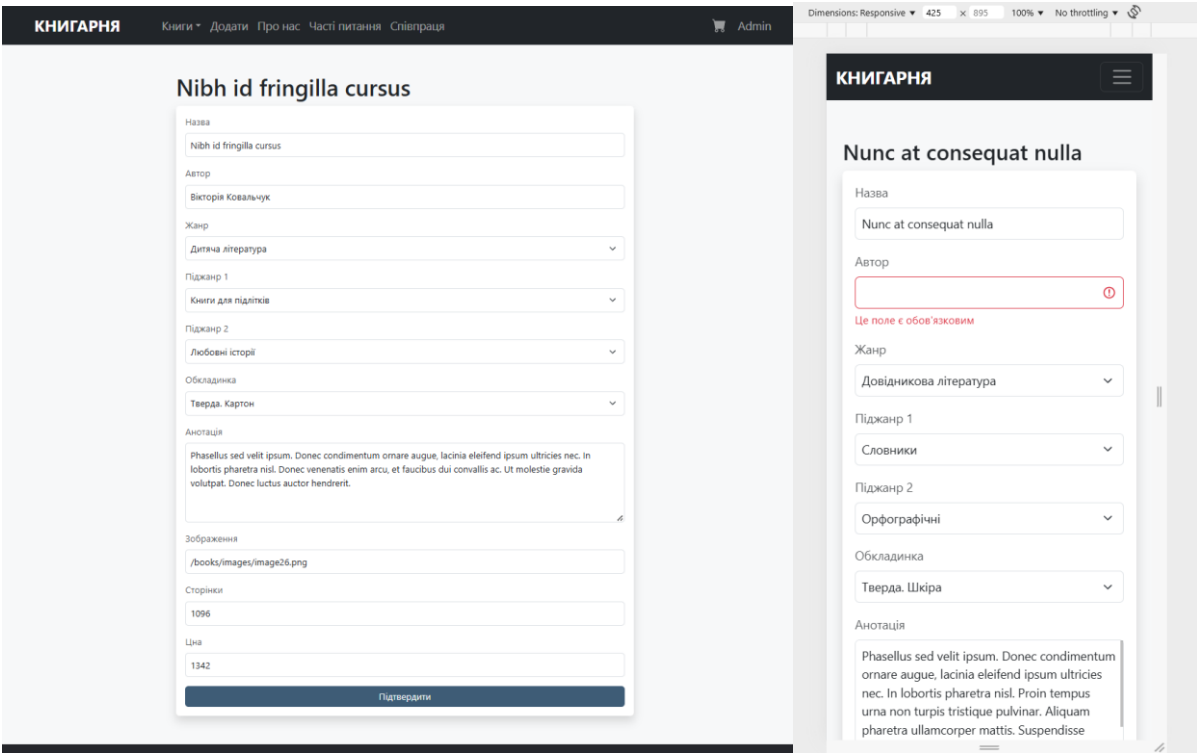


Сторінка опису книги містить детальний набір даних про книгу та NarrowGallery із подібними книгами. Сторінка активно використовує існуючі компоненти. Адаптація блоку з із безпосереднім описом вимагала значних зусиль.

					КР. ІІІ – 27.00.00.000 ПЗ	Арк.
						40
Змін.	Арк.	№ докум.	Підпис	Дата		



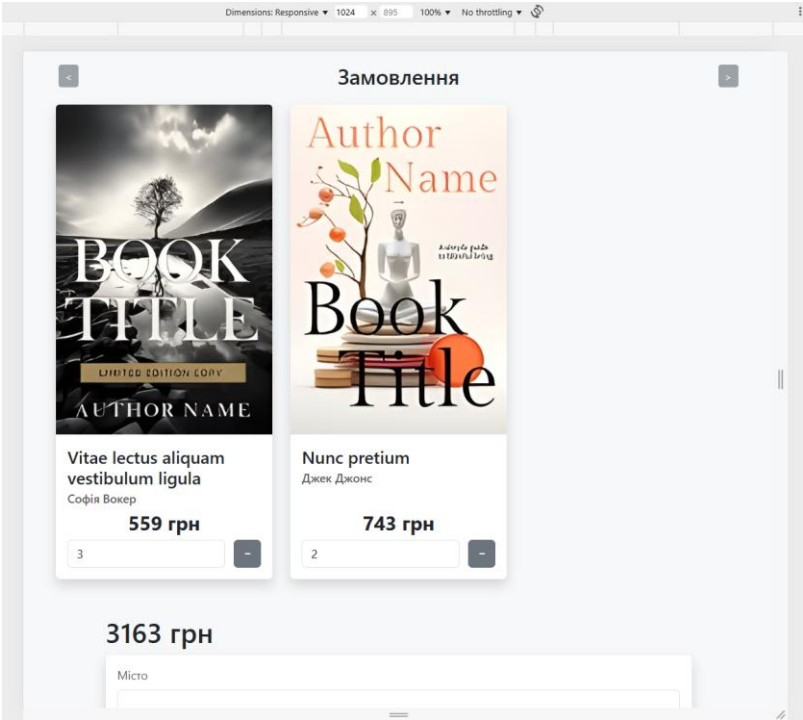
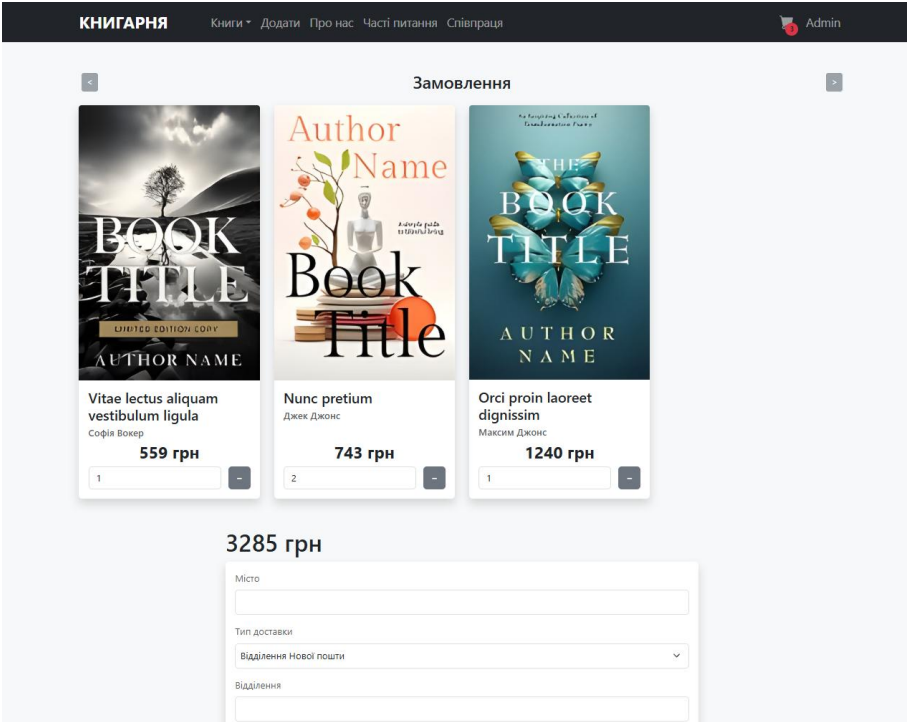
## 2.4.6 Сторінка редагування книги



Сторінка містить форму та забезпечує валідацію вводу. Єдиною проблемою було оновлення дочірніх жанрів при зміні користувачем батьківського. Загалом сторінка складається з існуючих компонентів.

					КР. III – 27.00.00.000 ПЗ	Арк.
						41
Змін.	Арк.	№ докум.	Підпис	Дата		

### 2.4.7 Корзина



Містить набір доданих користувачем книг у NarrowGallery із дещо зміненими картками книг. Книги можна вилучити з корзини або змінити їхню кількість, ціна замовлення автоматично перераховується.

					КР. III – 27.00.00.000 ПЗ	Арк.
						42
Змін.	Арк.	№ докум.	Підпис	Дата		

Софія Вокер

559 грн

743 грн

3163 грн

Місто

Івано-Франківськ

Тип доставки

Поштомат Нової пошти

Поштомат

7

Оплата

Передплата

Підтвердити

3163 грн

Місто

Це поле є обов'язковим

Тип доставки

Поштомат Нової пошти

Поштомат

7

Оплата

Передплата

Підтвердити

Містить поле для оформлення замовлення із валідацією. Доступні форми оплати залежать від вибраного типу доставки. Загалом, ця сторінка – черговий приклад ефективного повторного використання компонентів.

					КР. III – 27.00.00.000 ПЗ	Арк.
						43
Змін.	Арк.	№ докум.	Підпис	Дата		

# Висновок

В процесі виконання курсової роботи було створено функціональний вебзастосунок для онлайн-книгарні з використанням сучасних засобів компонентного програмування, зокрема бібліотеки React. Реалізовано ключові функції: перегляд каталогу книг, відображення інформації про кожну книгу, додавання до кошика, авторизація користувача, а також редагування інформації про книги адміністраторами. Для забезпечення зручного доступу до функцій застосунку створено навігаційне меню, інформаційні сторінки та уніфіковані компоненти заголовка і підвалу.

Серед переваг реалізованого проєкту можна відзначити:

- **Модульність** – структура застосунку дозволяє легко додавати нові сторінки та розширювати функціональність без порушення цілісності системи;
- **Повторне використання** – компонентний підхід дозволяє потворно використовувати компоненти в різних місцях додатку, суттєву прискорюючи процес розробки
- **Інтуїтивний інтерфейс** – використання React Bootstrap забезпечило адаптивну, зрозумілу та візуально привабливу верстку;
- **Реактивність** – усі дії користувача (додавання книг до кошика, авторизація тощо) миттєво відображаються в інтерфейсі завдяки реактивній архітектурі React.

Контекстне управління даними про користувача та вмістом кошика реалізовано за допомогою React Context API, що дало змогу централізовано управляти станом застосунку.

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						44
Змін.	Арк.	№ докум.	Підпис	Дата		

У розробці було використано: середовище Visual Studio Code, платформу Node.js, бібліотеку React, маршрутизацію через React Router та компоненти з React Bootstrap.

Отримані результати підтверджують доцільність використання React у розробці сучасних клієнтських вебзастосунків. Побудована структура та підходи до організації коду дозволяють використовувати даний проєкт як основу для подальшого розвитку.

У перспективі ПС може бути вдосконалена шляхом:

- інтеграції з повноцінною серверною частиною для зберігання замовлень та облікових записів;
- розширення фільтрації та пошуку книг;
- розширення набору адміністративних інструментів для управління роботою сервісу.

					КР. ІІ – 27.00.00.000 ІЗ	Арк.
						45
Змін.	Арк.	№ докум.	Підпис	Дата		

# Список використаних джерел

1. Component-based software engineering. Wikipedia : вебсайт. URL: [https://en.wikipedia.org/wiki/Component-based\\_software\\_engineering](https://en.wikipedia.org/wiki/Component-based_software_engineering) (дата звернення: 24.05.2025).
2. Component-Based Software Engineering Explained. WeSoftYou : вебсайт. URL: <https://wesoftyou.com/outsourcing/component-based-software-engineering-explained/> (дата звернення: 24.05.2025).
3. Principles of Component-Oriented Programming. O'Reilly : вебсайт. URL: <https://www.oreilly.com/library/view/programming-net-components/0596102070/ch01s03.html> (дата звернення: 25.05.2025).
4. SPA (Single-page application. Mozilla : вебсайт. URL: <https://developer.mozilla.org/en-US/docs/Glossary/SPA> (дата звернення: 26.05.2025).
5. Single Page Application. Bloomreach : вебсайт. URL: <https://www.bloomreach.com/en/blog/what-is-a-single-page-application> (дата звернення: 26.05.2025).
6. The Pros and Cons of Single-Page Applications. Medium : вебсайт. URL: [https://medium.com/@VAISHAK\\_CP/the-pros-and-cons-of-single-page-applications-spas-06d8a662a149](https://medium.com/@VAISHAK_CP/the-pros-and-cons-of-single-page-applications-spas-06d8a662a149) (дата звернення: 28.05.2025).
7. React (software). Wikipedia вебсайт. URL: [https://en.wikipedia.org/wiki/React\\_\(software\)#Function\\_components](https://en.wikipedia.org/wiki/React_(software)#Function_components) (дата звернення: 29.05.2025).
8. Learn. React : веб-сайт. URL: <https://react.dev/learn> (дата звернення: 19.04.2025).
9. Components. React Bootstrap : вебсайт. URL: <https://react-bootstrap.netlify.app/docs/components/accordion> (дата звернення: 08.04.2025).
10. Репозиторій проєкту. Github : вебсайт. URL <https://github.com/Denchyk-chyk/cw-ad-2024-code> (дата звернення: 27.05.2025).

					КР. ІІ – 27.00.00.000 ІІЗ	Арк.
						46
Змін.	Арк.	№ докум.	Підпис	Дата		