

ЗВІТ З ПЕРЕВІРКИ НА ПЛАГІАТ

ЦЕЙ ЗВІТ ЗАСВІДЧУЄ, ЩО ПРИКРПЛЕНА РОБОТА

Ларіоненко_ІПЗ-119К9

БУЛА ПЕРЕВІРЕНА СЕРВІСОМ ДЛЯ ЗАПОБІГАННЯ ПЛАГІАТУ MY.PLAG.COM.UA І

МАЄ:

СХОЖІСТЬ

6%

РИЗИК ПЛАГІАТУ

44%

ПЕРЕФРАЗУВАННЯ

1%

НЕПРАВИЛЬНІ ЦИТУВАННЯ

0%

Назва файлу: Ларіоненко_ІПЗ-119К9.docx

Файл перевірено: 2023-06-12

Звіт створено: 2023-06-12

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ПрАТ «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ (library.econom.zp.ua)
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Предметно-циклова комісія інформаційних технологій

ДО ЗАХИСТУ ДОПУЩЕНА

Голова ПЦК _____

спеціаліст в/к Сабанов

(library.econom.zp.ua) С.О.

ВИПУСКНА РОБОТА МОЛОДШОГО СПЕЦІАЛІСТА (library.econom.zp.ua)

**РОЗРОБКА (LIBRARY.ECONOM.ZP.UA) ВЕБ-ДОДАТКУ
"ІНТЕРНЕТ-МАГАЗИН З ПРОДАЖУ КНИЖОК"
ІЗ ЗАСТОСУВАННЯМ REACTJS**

Виконав
ст. гр. ПЗ-119К9 _____

Ларіоненко О.І.

Керівник
професор _____

Сабанов С.О.

Запоріжжя
2023

СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАТ «ПВНЗ «ЗІЕІТ»

Предметно-циклова комісія (library.econom.zp.ua) інформаційних технологій

ЗАТВЕРДЖУЮ

Голова ПЦК, спеціаліст в/к

Сабанов (library.econom.zp.ua)

С.О. _____

21 лютого 2023 р.

ЗАВДАННЯ

НА ВИПУСКНУ РОБОТУ МОЛОДШОГО СПЕЦІАЛІСТА

студенту гр. ППЗ-119К9

спеціальності: 121 – Інженерія програмного забезпечення

Ларіоненку Олегу Ігоровичу

(прізвище, ім'я, по батькові) (library.econom.zp.ua)

1. Тема: Розробка веб-додатку "Інтернет-магазин з продажу книжок" із застосуванням ReactJS

затверджена наказом по інституту: № 09.2-14 від 20 лютого 2023 року

(library.econom.zp.ua)

2. Термін здачі студентом закінченої роботи: 20 червня 2023 року

3. Перелік питань, що підлягають розробці:

1. Провести огляд літератури (library.econom.zp.ua) та інтернет-джерел, присвячених

тематиці випускної роботи.

2. Виконати огляд існуючих типів веб-додатків, загальних принципів їх створення, виділити особливості щодо інтернет-магазинів.

3. Розглянути інструментальні засоби створення інтернет-магазинів з метою вибору оптимальних для створення проекту.

4. *Визначитися з основними функціональними блоками проекту та структурою бази даних.*
5. *Реалізувати алгоритм роботи додатку.*
6. *Протестувати розробку та провести остаточне налагодження*
7. *Створити достатнє наповнення БД для нормального користування системою.*
8. *Продумати питання хостингу*
9. *Оформити результати роботи у вигляді пояснювальної записки, що відповідає стандартам підприємства щодо оформлення випускних робіт.*

4. Календарний графік

№ етапу	Зміст	Термін виконання	Готовність (%), підпис керівника	Підпис керівника про повну готовність етапу, дата
1	Збір практичного матеріалу за темою	16.01.23-17.02.23		
2	I атестація. I розділ випускної роботи	27.03.23-01.04.23		
3	II атестація. II розділ випускної роботи	01.05.23-06.05.23		
4	III атестація. III розділ випускної роботи, висновки та рекомендації, додатки, реферат.	29.05.23-03.06.23		
5	Перевірка випускної роботи на оригінальність	15.05.23-12.06.23		
6	Доопрацювання випускної роботи, підготовка презентації, отримання відгуку керівника та рецензії	05.06.23-10.06.23		
7	Попередній захист випускної роботи	12.06.23-18.06.23		
8	Подача випускної роботи на кафедру	за 3 дні до захисту		

a r y · e c o n o m · z p · u a)				
9	Захист випускної роботи	19.06.23- 24.06.23		

Дата видачі завдання: 22 лютого 2023 р.

Керівник випускної роботи _____ Сабанов С.О.

Завдання прийняв до виконання _____ (library.econom.zp.ua)
Ларіоненко О.І.

РЕФЕРАТ

Випускна робота молодшого спеціаліста містить (library.econom.zp.ua)

60 сторінок, 28 рисунків, 6 додатків, 25 бібліографічних посилань.

Об'єкт розробки – спеціалізований веб-сайт.

Метою роботи є розробка (library.econom.zp.ua) веб-додатку, що реалізує функціонал он-лайн магазину з продажу книжок

У випускній роботі розглядаються основні принципи функціонування та етапи побудови веб-додатків, що використовуються в якості торгівельних майданчиків, а також спеціальні інструментальні засоби створення подібних ресурсів.

Детально викладено процес розробки інтернет-магазину з продажу книжок, агреговано інформаційно-довідкові матеріали зі створення фулстек веб-додатку для продажу друкованих та електронних видань.

Додаток розроблено на базі клієнт-серверної архітектури, з використанням бібліотеки React.

BACKEND, FRONTEND, HTML, NODE.JS, REACT JS, MOBX, AXIOS,
WEB-САЙТ, EXPRESS, POSTGRESQL, SEQUELIZE, БАЗА ДАНИХ,
КЛІЄНТ-СЕРВЕР, СУБД

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	9
РОЗДІЛ 1. ТИПИ ВЕБ_ДОДАТКІВ ТА ЗАГАЛЬНІ ПРИНЦИПИ ЇХ СТВОРЕННЯ.....	10
1.1. Типи веб-додатків.....	10
1.1.1. Кастомні додатки.....	10
1.1.2. Ноукод-додатки.....	10
1.2. Архітектура веб-додатків.....	11
1.2.1. SPA (Single Page Application)	11
1.2.2. MPA (Multi Page Application)	12
1.2.3. PWA (www.wykop.pl) (Progressive Web Application).....	13
1.3. Етапи створення веб-додатків.....	13
1.3.1. Аналітика	13
1.3.2. UX-дизайн	13
1.3.3. UI-дизайн	14
1.3.4. Фронтенд	15
1.3.5. Бекенд	16
1.3.6. Тестування	16
1.4. Інтернет-магазин, як специфічна форма веб-додатків.....	17
1.4.1. Класифікація інтернет-магазинів	17
1.4.2. Архітектура інтернет-магазинів	18
1.5. Огляд типових інтернет-магазинів з продажу книжок.....	20
1.5.1. Огляд інтернет-магазину «Book24.ua»	20
1.5.2. Огляд інтернет-магазину видавництва «Ранок»	22
1.5.3. Інтернет-магазин Yakaboo	24

1.6.	Висновки за першим розділом	26
РОЗДІЛ 2. ІНСТРУМЕНТАЛЬНІ ЗАСОБИ СТВОРЕННЯ ІНТЕРНЕТ-МАГАЗИНУ		28
2.1.	Мова програмування JavaScript	28
2.2.	Технології створення серверної частини веб-додатку Node.JS	29
2.3.	СУБД PostgreSQL	31
2.4.	ORM Sequelize	33
2.5.	Бібліотеки React	35
2.5.1.	React Router	35
2.5.2.	React Mobx	37
2.5.3.	React-Bootstrap	38
2.6.	Середовище розробки Visual Studio Code	40
2.7.	Висновки за другим розділом	41
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ "ІНТЕРНЕТ-МАГАЗИН ПРОДАЖУ КНИЖОК"		42
3.1.	Призначення та функціональні можливості додатку.....	42
3.2.	Структура бази даних	43
3.3.	Інтерфейс користувача	47
3.4.	Адміністративна панель	51
3.5.	Висновки за третім розділом	54
ВИСНОВКИ.....		56
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		57
ДОДАТКИ.....		60
ДОДАТОК А.....		61
ДОДАТОК Б.....		62
ДОДАТОК В.....		63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Слово / словосполучення	Скорочення
A	
Application Programming Interface	API
B	
Business-to-Business	B2B
Business-to-Consumer	B2C
C	
Compact Disc	CD
Common Gateway Interface	CGI
D	
Digital Versatile Disc	DVD
Document Object Model	DOM
H	
HyperText Markup Language (eprints.library.odku.edu.ua)	HTML
I	
Identifier	ID
J	
JSON Web Token	JWT
M	
Multi Page Application	MPA
O	
Object-Relational Mapping	ORM

P	
Progressive Web Application	PWA
S	
Search engine optimization	SEO
Single Page Application	SPA
Structured Query Language	SQL
U	
User interface	UI
Uniform Resource Locator	URL
User eXperience	UX
Д	
дивитися	див
P	
рисунок	рис
С	
Система управління базами даних	СУБД

На сьогодні актуальність питання продажу та покупки майна, послуг стає все актуальнішим серед сучасних людей. В останні кілька років люди були вимушені максимально ефективно використовувати свій час. Фізичні походи по магазинам з витрачанням часу на дорогу і на проведення покупок з можливим результатом, що не буде придбано жодного товару. Щоб не займатися марними витратами часу все більше людей починають користуватися онлайн сервісами з продажу товарів.

В новому столітті все більше людей діджиталізуються, кожна людина має свій особистий смартфон який заміняє більшість матеріальних предметів які були носіями інформації.

Онлайн-шопінг став неймовірно популярним серед споживачів. Це пов'язано з тим, що не потрібно витратити зайвий час на поїздку до інтернет-магазину, а широкий асортимент і легкість пошуку товарів просто неймовірні. У магазинах з широким асортиментом товарів споживачам часто доводиться витратити багато часу на пошук та оцінку товарів, тоді як в Інтернеті все набагато простіше і доступніше. Наприклад, більшість сайтів мають корисні фільтри на додаток до категорій товарів, що дуже полегшує відображення результатів, особливо для товарів, які відповідають ключовим параметрам. Онлайн-шопінг є вибором багатьох споживачів через його численні переваги, включаючи більш вигідні ціни в порівнянні з реальними магазинами, тому дуже багато споживачів вибирають їх. Сьогодні не так важливо, що продавати – за певних навичок та сучасної розвиненості ринку продати можна майже все.

У випускній роботі розглянуто сучасні інтернет-магазини з продажу книжок, проаналізовано переваги й недоліки їхніх користувацьких якостей, та викладено повний процес створення сучасного веб-додатку, що має спеціалізацію книжкової крамниці, побудованої з використанням REACT JS.

РОЗДІЛ 1

ТИПИ ВЕБ ДОДАТКІВ ТА ЗАГАЛЬНІ ПРИНЦИПИ ЇХ СТВОРЕННЯ

1.1. Типи веб-додатків

1.1.1. Кастомні додатки

Існує кілька способів створення та розробки додатків. Один із способів розробки полягає в тому, що команда розробників створює власне унікальне рішення з нуля, створюючи додаток, написаний на одній із сучасних мов прикладного програмування з індивідуальним дизайном, який відповідає потребам замовника. Такий тип додатків називається кастомним і створюється вручну, від основи додатку до даху майбутнього додатку. Перевагами такого рішення є персоналізація та безпека даних. Недоліком такого підходу є висока ціна і час, витрачений на розробку програми. Спеціалізований тип розробки безпосередньо підходить для висококонкурентного типу ринку. Крім того, команда розробників повинна мати підтримку для підтримки продуктів, випущених за допомогою цього методу.

1.1.2. Ноукод-додатки

Інший спосіб розробки додатку – створити його за допомогою конструктора. Платформа no-code дозволяє створювати додатки, використовуючи готові шаблони для створення маркетплейсів або сторінок-вітрин. Потім можна редагувати шаблон, налаштовувати функціонал і завантажувати контент. No-code обирають, тому що це швидко і дешево. Основними недоліками no-code рішень є залежність від використовуваної платформи та безпека даних. Якщо намір розвивати ідею відсутній, то для

реалізації простого додатку достатньо використовувати конструктор. Сьогодні на ринку є багато білдерів з різними функціями, ось деякі з них:

- Appy Pie;
- Shoutem;
- Thunkable.[1]

1.2. Архітектура веб-додатків

1.2.1. SPA (Single Page Application)

Single Page Application – це односторінкова програма. Для розробки таких програм використовують HTML і JavaScript. По суті це розробка лендингу, тільки інтерактивного. Але SPA може бути досить складним. Суть односторінкових програм у тому, що на сервері зберігається одна HTML-сторінка, контент на якій оновлюється в міру прокручування або переходів за посиланнями. Тобто коли проходить ініціація події натискання на кнопку, не відбувається переадресація на нову сторінку – елементи додаються до вже завантаженої. Наприклад, за цим принципом працює Gmail.

Переваги:

- Розробляти такі програми простіше, ніж багатосторінкові.
- Програми працюють швидко: контент на сторінці просто змінюється в міру руху користувача, його завантаження не вимагає багато ресурсів.
- На одній сторінці простіше зробити одноманітний дизайн, а користувач точно не загубиться і не заплутається.

Недоліки:

- Налаштування SEO-оптимізації для SPA буде більш трудомістким, ніж для інших варіантів.
- Важко гарантувати безпеку таких сторінок: вони більше схильні до зламів і витоків, ніж MPA і PWA.

1.2.2. MPA (Multi Page Application)

Multi Page Application – це багатосторінкові програми. Вони дозволяють користувачеві не просто скролити вікно браузера, а переходити між окремими сторінками. Завантаження контенту в таких програмах відбувається цілими сторінками. Це означає, наприклад, що якщо користувач здійснив оплату, SPA підвантажить віконце з підтвердженням, а в MPA сторінка оплати повністю оновиться. Приклад багатосторінкової програми – інтернет-магазин Amazon.

Переваги:

- Проста SEO оптимізація.
- Більше звичний варіант для більшості користувачів, які звикли переходити між сторінками.
- Якщо програма складна, з великою кількістю функцій, вона точно повинна бути багатосторінковою. Перехід сторінок дозволяє користувачеві легко потрапляти на потрібні розділи. Скролити величезний односторінковий додаток на початок, коли раптово знадобилася якась інформація звідти, нікому не захочеться.

Недолік – складніша і дорожча розробка та дизайн, ніж для SPA.

1.2.3. PWA (Progressive Web Application)

Progressive Web Application – прогресивні веб-додатки. Це щось середнє між розробкою мобільного додатка та вебсайту. PWA можна відразу з браузера встановити на головний екран смартфона в обхід магазинів програм. А ще такі програми працюють офлайн і надсилають push-сповіщення, але при цьому відкриваються у браузері. Це можливо завдяки технології Service Worker – скрипту, через який проходять усі взаємодії між фронтендом та бекендом. Цей скрипт має доступ до кешу та даних. По суті, до більшості

сайтів можна дописати Service Worker – і вийде PWA. Тому часто на цю технологію переходять ЗМІ, наприклад, The Washington Post.

Переваги:

- Поєднання офлайн-і онлайн-режиму зручне для користувача.
- Такі програми можна порівняно швидко розробити.

Недоліки:

- Програма може працювати з перебоями у старих версіях браузера.
- Є браузери, які взагалі не підтримують подібні програми, наприклад, Firefox.

1.3. Етапи створення веб-додатків

1.3.1. Аналітика

Звертатися до студії варто лише тоді, коли вже є готова ідея для додатку і наявне загальне уявлення про те, як він працює. Для цього підприємець повинен мати досвід роботи в обраній галузі. А от дослідження ринку можна довірити експертам. Аналітики розберуться, яке місце може зайняти продукт на ринку, на яку аудиторію найвигідніше орієнтуватися, які функції потрібні цим людям і яку модель монетизації обрати.

В результаті, після роботи з аналітиком буде чітке розуміння того, для кого розроблюється додаток, які користувацькі завдання він вирішує і спосіб заробітку на ньому.

1.3.2. UX-дизайн

Сучасні користувачі звикли до добре продуманого та якісного інтерфейсу. У конкурентному середовищі, якщо щось у додатку здається дратівливим або незрозумілим, люди одразу ж переходять за іншими

посиланнями в пошукових системах. Тому важливо вивчити шлях користувача – яку послідовність дій виконає користувач і як на них відреагують елементи сайту – UX дизайнери прототипують, визначають ключові блоки та елементи, показують взаємодію між ними.

Результатом є готова діаграма додатку, що показує інтерфейсні блоки та взаємозв'язки між ними. Цей прототип можна використовувати для оцінки функціональності та доступності додатку.

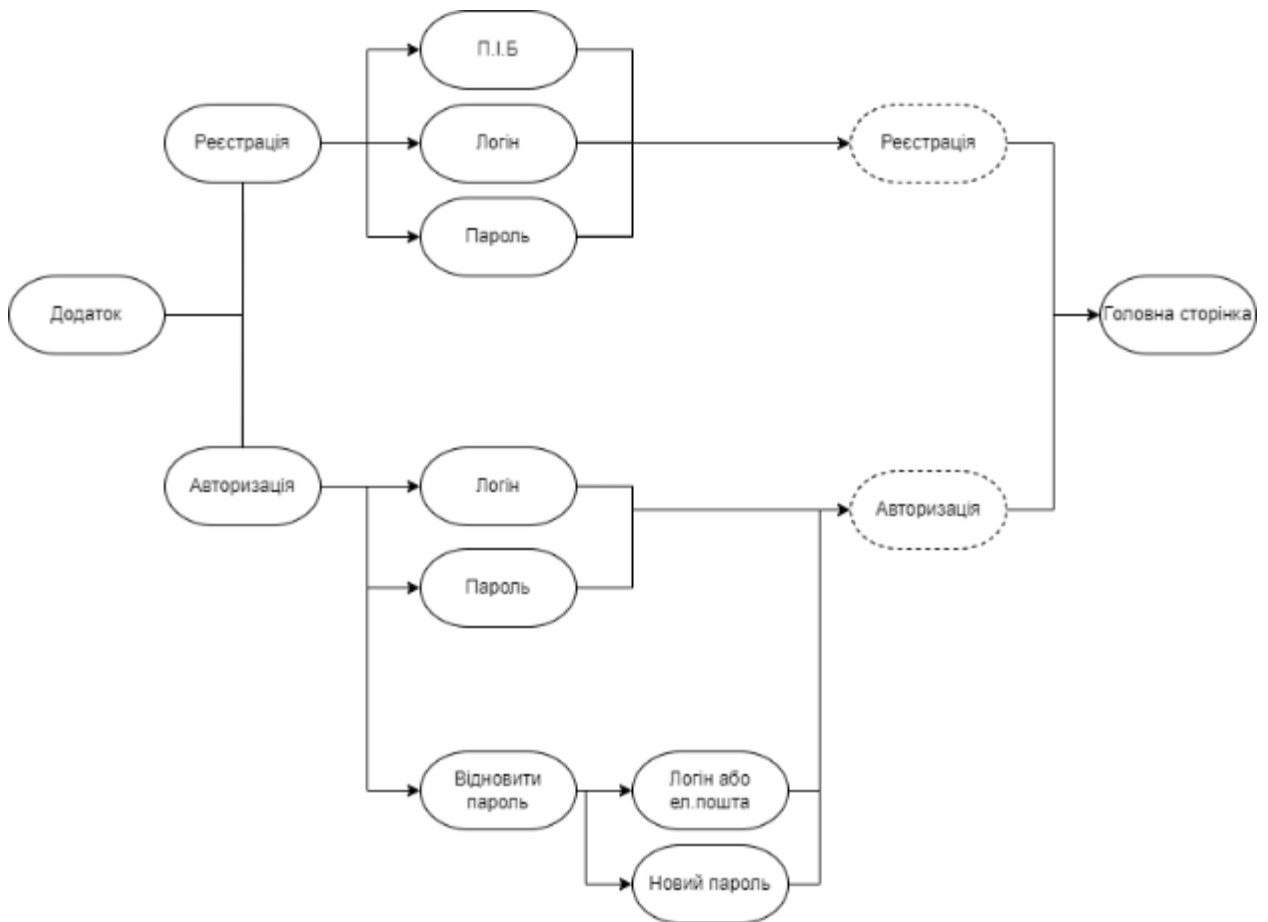


Рисунок 1.1 – Приклад схема взаємодії блоків інтерфейсу та взаємозв'язку між ними

1.3.3. UI-дизайн

Робота UI-дизайнера полягає у візуалізації будівельних блоків, створених UX-дизайнером. Дизайн інтерфейсу включає в себе розробку іконок

і кнопок, позиціонування екранів, вибір кольорів і шрифтів, а також створення посібників зі стилю інтерфейсу для подальшої розробки.

Після завершення робіт виходить майже готовий, але ще не функціонуючий додаток. Тільки після етапу проектування сайту можна узгодити точну вартість і вимоги до розробки. Це буде залежати від складності проекту та побажань клієнта щодо функціональності та дизайну.

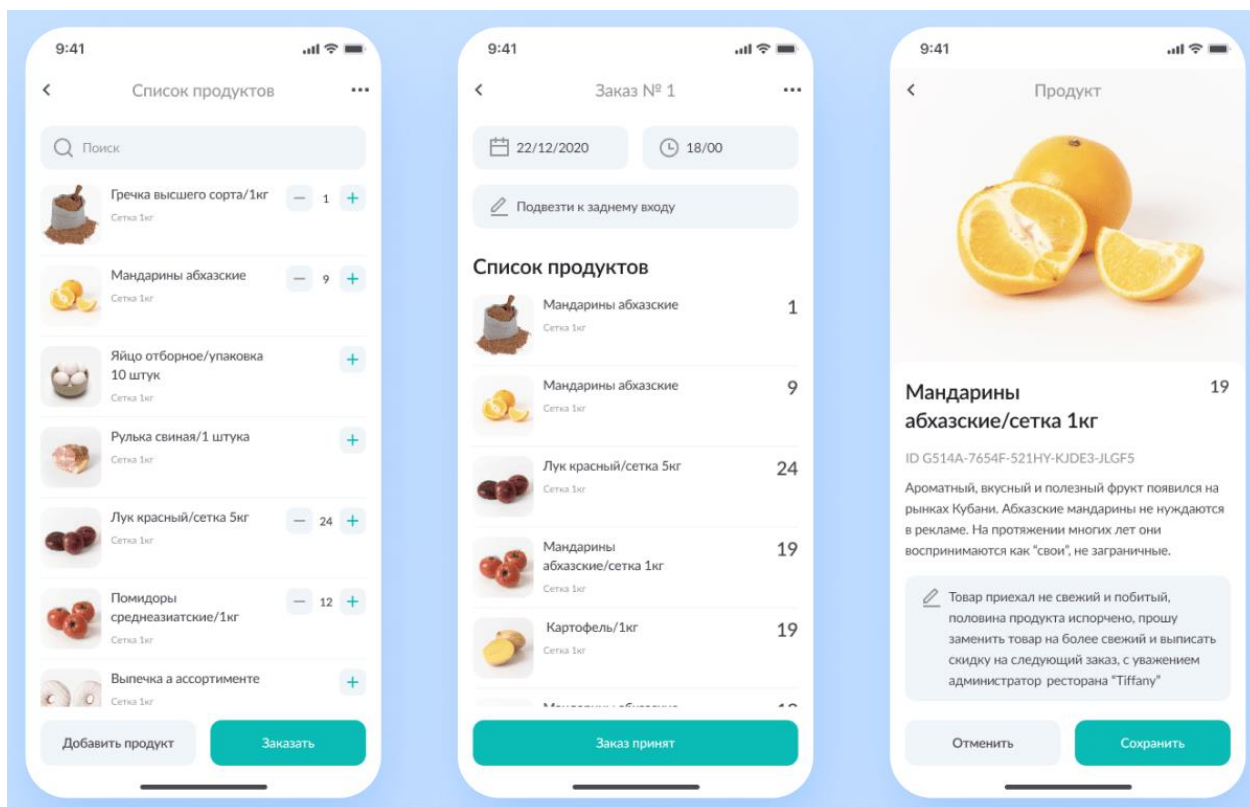


Рисунок 1.2 – Приклад демонстрації роботи UI дизайнера

1.3.4. Фронтенд

Фронтенд-розробники відповідають за зовнішню частину веб-сайту, яку бачать користувачі. Однак фронтенд – це не лише розробка макету. Фронтенд-розробники відповідають за адаптивний і чутливий дизайн, тобто за те, щоб веб-сайт виглядав правильно на різних пристроях і в різних версіях браузерів. На цьому етапі вони визначають мікро-взаємодії, такі як процес завантаження елементів, клікабельність та анімація.

В результаті дизайн став інтерактивним, дозволяючи користувачам натискати кнопки та переходити за посиланнями. Однак додаток ще не виконує свою функцію.

1.3.5. Бекенд

Наступний етап – бекенд-розробка. Це те, як додаток взаємодіє з базою даних і як обробляються такі процеси, як платежі та бронювання. Бекенд-розробник відповідає за належне функціонування веб-сайту, зв'язки між компонентами додатку, зберігання та структуру даних, логіку та алгоритми розрахунків, а також інтеграцію із зовнішніми сервісами, такими як платіжні системи та соціальні мережі.

Результатом цієї роботи є робочий додаток, який майже завершений або готовий до тестування.

1.3.6. Тестування

Це необхідний етап, щоб фінальний додаток працював так, як було задумано. Головне завдання тестувальника – перевірити роботу програми перед релізом, щоб команда вийшла на ринок із якісним продуктом. Тестувальники вивчають документацію продукту та складають тест-кейси — список функцій, які треба перевірити, та їх послідовність. Потім вони вручну імітують дії користувача у різних сценаріях чи пишуть скрипти для автоматичного тестування. Після цього розробники отримують звіт зі списком помилок та рекомендаціями щодо виправлення.

В результаті в протестованому стабільному додатку не виникло жодної помилки, а ризик помилок був зведений до мінімуму. [2]

1.4. Інтернет-магазин, як специфічна форма веб-додатків

1.4.1. Класифікація інтернет-магазинів

Інтернет-магазини є одним із видів електронної комерції. Продаж своїх товарів та послуг через Інтернет є основним комерційним використанням веб-сайтів для більшості виробничих та роздрібних підприємств. Компанії створюють веб-сайт, на якому розміщують інформацію про свої товари та послуги, ціни та гарантії для клієнтів. Інтернет-магазини стали для компаній одним із необхідних та ефективних інструментів для збільшення продажів, обороту та іміджу, а також для їх успішного розвитку.

Інтернет-магазини можна класифікувати за такими видами:

- **За моделлю бізнесу** – повністю онлайн магазин, поєднання офлайн-бізнесу з онлайн-бізнесом, (krs.chmnu.edu.ua) на аутсорсингу, за допомогою системи дропшипінг.
- За обсягами продажу – роздрібно-оптовий продаж.
- За видами продажу – послуги надаються іншим комерційним підприємствам (B2B) або кінцевому користувачу (B2C).
- **За способами отримання доходу** (krs.chmnu.edu.ua) – отримання доходу продажем товару або послуг від виробника чи офіційного представника, продаж товарів та послуг по спеціальним програмам (участь у чужих продажах), продаж інформації(контенту).
- **За товарним асортиментом** (krs.chmnu.edu.ua) – книги, накопичувачі пам'яті, комп'ютерна та побутова техніка, мобільні телефони, меблі, одяг, взуття, годинники та інші товари.
- По відношенню з постачальниками – наявність свого складу або робота за договорами на постачання.
- По представлених товарах у каталозі – інтернет-вітрини, інтернет-магазини, онлайн аукціони.

Однією з найважливіших особливостей інтернет-магазинів є те, що більшість взаємодії між продавцями та покупцями відбувається в режимі онлайн. Багато етапів процесу купівлі-продажу можна успішно здійснювати віддалено. Це важлива перевага. Водночас це одна з проблем, яку мають подолати інтернет-магазини. Тобто викликати у відвідувачів довіру та спонукати їх до покупки.[3]

1.4.2. Архітектура інтернет-магазинів

Архітектура веб-сайту – це організація інформації та навігації таким чином, щоб відвідувачі могли краще знаходити потрібні їм дані. Добре продумана і грамотна архітектура веб-сайту гарантує, що користувачі витратять менше часу на пошук потрібної інформації.

Побудову веб-сайту слід розробляти з урахуванням найважливішої інформації щодо просування товарів/послуг на інтернет-ринку. У процесі створення нової структури веб-сайту або оптимізації існуючої структури веб-сайту слід зосередити увагу споживачів на цій інформації та керувати доступом потенційних клієнтів веб-сайту до найбільш важливих частин веб-сайту відповідно до позиціонування продукту/послуги на ринку. Це необхідно зробити, правильно розставивши пріоритети розділів і сторінок на сайті, зробивши його головним входом на сайт, потенційні споживачі зможуть швидко знайти інформацію про необхідні їм товари/послуги, тим самим підвищуючи успіх вашого бізнесу в Інтернеті.

Під програмною частиною архітектури інтернет-магазину розуміють інтерфейс між операційною частиною та сервером. Операційна частина включає в себе середовище розробки інтернет-магазину. Серверна частина включає в себе хостинг інтернет-магазину на сайті провайдера, який підтримує технологію створення інтернет-магазину.



Рисунок 1.3 – Архітектура інтернет-магазину

У серверній частині архітектури виконується огляд роботи Інтернет-магазину в мережі Інтернет, взаємодії створеного програмного забезпечення магазину та сервісів, отримані наданням від власників серверів, провайдерів. Тому що після створення Інтернет-магазину з'являється необхідність в його розміщенні в інтернеті і вибором місць розміщення, простіше кажучи вибрати хостинг. Сервіси, які надають послуги хостингу, дозволяють розміщувати інформацію (веб-сторінки) на своїх серверах. Тож не доведеться встановлювати власний сервер онлайн, заощаджуючи багато грошей. Залежно від тарифного плану надається деякий простір для роботи з жорстким диском, електронною поштою, CGI, сервером MSQ тощо.

Адміністрування містить інструменти для керування онлайн-магазином, включаючи загальні налаштування магазину та спеціальні налаштування.

Адміністрація містить основні налаштування інтернет-магазину:

- Загальні налаштування магазину (назва магазину, адреса, номер телефону, адреса електронної пошти магазину тощо).
- Налаштування форм реєстрації клієнтів для інтернет-магазинів.
- Загальні умови відвантаження та пакування продукції.

- Облаштування складу.
- Файл, у якому записуються параметри реєстрації та службова інформація.
- Налаштування формату виведення товару в інтернет-магазині. Можливість налаштувати вихідний формат продукту відповідно до поставлених потреб.
- Усі можливі налаштування каталогу, напр. додавання, видалення, редагування продуктів та категорій, взаємодія з виробниками, імпортувати/експортувати продукти тощо.

У клієнтській частині архітектури на сторінках інтернет-магазину розробляються найбільш зручні та доступні проекти потенційних клієнтів. Розробка інтерфейсу, доступні та зрозумілі діалоги, зручна система оплати та доставки. Ключовим фактором є зворотній зв'язок. Це дозволяє клієнту висловити свою думку про той чи інший товар/послугу, якість обслуговування та магазину в цілому.[4]

1.5. Огляд типових інтернет-магазинів з продажу книжок

1.5.1. Огляд інтернет-магазину «Book24.ua»

Book24 – молодий книжковий онлайн-магазин, який швидко розвивається. Реклама магазину обіцяє створити максимально комфортні та вигідні умови для всіх любителів хорошої літератури та заявляє про запас понад 10 000 книг, що представлені онлайн-магазині.

Головна сторінка інтернет-магазину доволі приваблива але можна побачити контраст контенту який розміщено посередині, а саме банеру на фоні контенту який розташований збоку (рисунок 1.4). Сторінка зразу надає користувачу взаємодію з верхньою панеллю та з боковою для взаємодії з сайтом, можна помітити спосіб розміщення інформації і контенту по ходу серфінгу сайтом. [5]

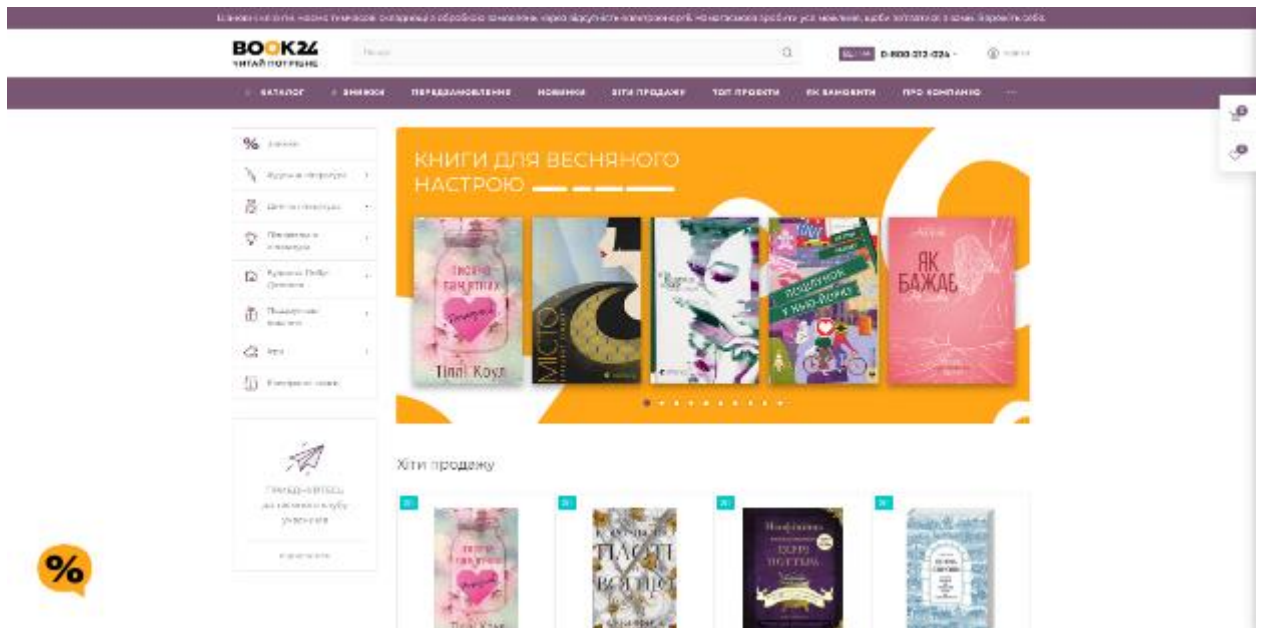


Рисунок 1.4. – Видгляд головної сторінки онлайн-магазину з продажу книжок book24.ua

Банер та вітрина хітів продажу друкованих творів – перший контент який бачить користувач. Якщо проскролити нижче головну сторінку, нарахується декілька вітрин з різними групами книг, слайдер видавництва, блог та відгуки.

При бажанні купити чи зацікавленості твором покупець потрапляє на сторінку з інформацією щодо продукту, який його зацікавив, та бачить доволі щільний та зосереджений контент (рисунок 1.5). Фотокартка книги, характеристика, блок з розрахунком ціни мають вигляд трьох прямокутників які розташовані паралельно один к одному.

Можна виділити суттєвий недолік: місце яке займає контент, що містить характеристику продукту, займає неприпустимо багато простору сторінки. Цей простір можна було б витратити на щось інше, наприклад – фотокартку та блок з формуванням ціни розмістити посередині а характеристику продукту видати в стислому виді в блоку ціни. Повну ж характеристику розмістити нижче. В іншому на сторінку продукту нарікань немає.

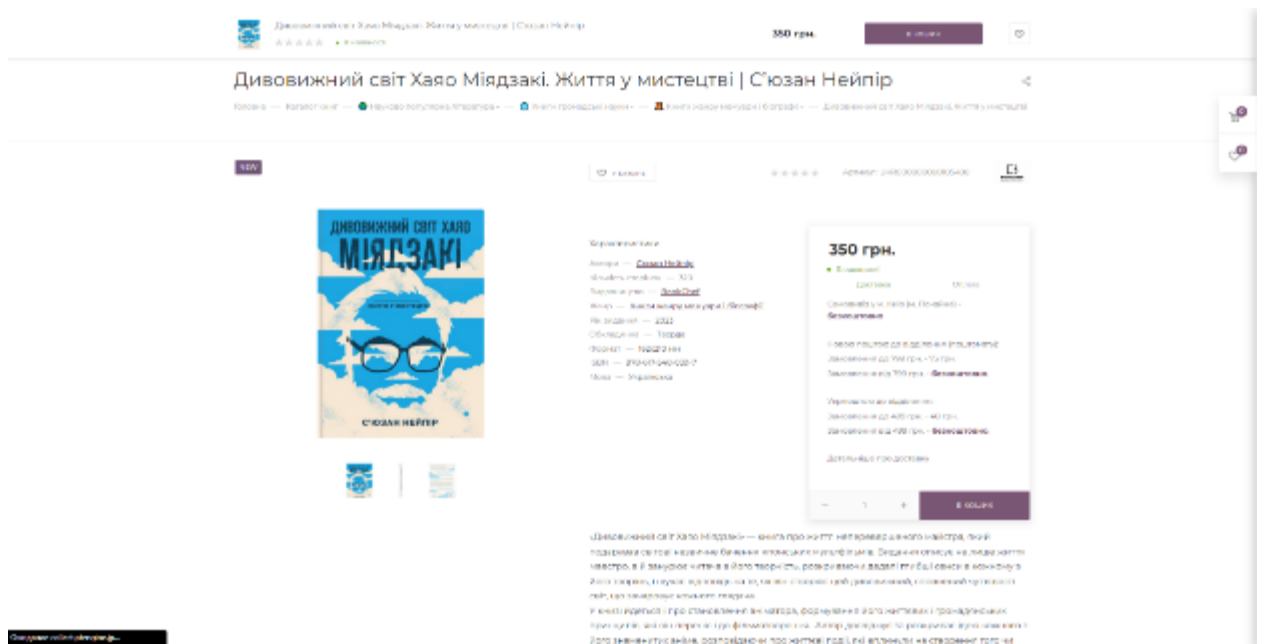


Рисунок 1.5. – Вигляд сторінки з інформацією товару в інтернет-магазині Book24.ua

Стосовно сайту сайту-магазину загалом, сайт має хороше оформлення та приємні для очей невиразні кольори. Дизайн головної сторінки та функціонал майже нічим на фоні конкурентів не виділяється.

1.5.2. Огляд інтернет-магазину видавництва «Ранок»

Ранок – харківське видавництво, засноване в 1997 році. Основні напрямки роботи – навчальна, методична та дитяча література. **Видавці створюють нестандартні за формою та змістом** книжки, розробляють, впроваджують і продають продукти, які забезпечують навчання від першого дня життя **до закінчення школи**. Компанія застосовує технології, які захищають природу, видає **книги на переробленому папері** та використовує **електронний документообіг і технології (uk.wikipedia.org)** для створення цифрових продуктів.

Інтернет-магазин має дуже красивий і зручний дизайн. [6] На головній сторінці сайту розміщено великий банер із останніми новинами зі світу читання, а також пропозиціями та новинками від українських авторів. Тут продають не лише книги, а й канцтовари, настільні ігри та саморобки.

Коли покупець натискає на будь-який товар на сайті, він потрапляє на сторінку з інформацією стосовно товару. Перед покупцем постає стислий центрований контент який затримує увагу виключно на середині екрану (рисунок 1.6). Контентне наповнення сторінки при переході на неї не містить нічого зайвого, тільки те, що повинен знати про продукт покупець.

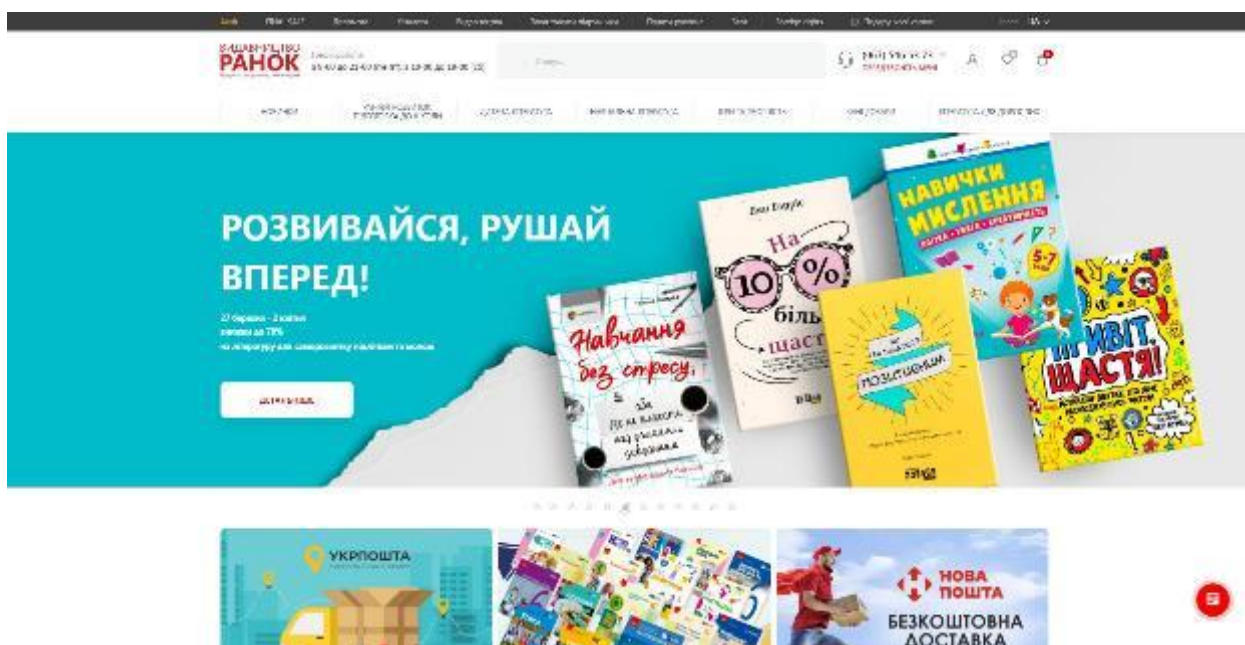


Рисунок 1.6. – Видгляд головної сторінки онлайн-магазину з продажу книжок ranok.com.ua

Якщо потрібна більш детальна інформація з описом продукту то цей контент знаходиться нижче, там можна ознайомитися з детальним описом товару, його змістом, фрагментом твору. Також є слайдер з переліком товарів які купують інші люди на основі жанру, теми, типу твору на сторінці якого знаходиться покупець.

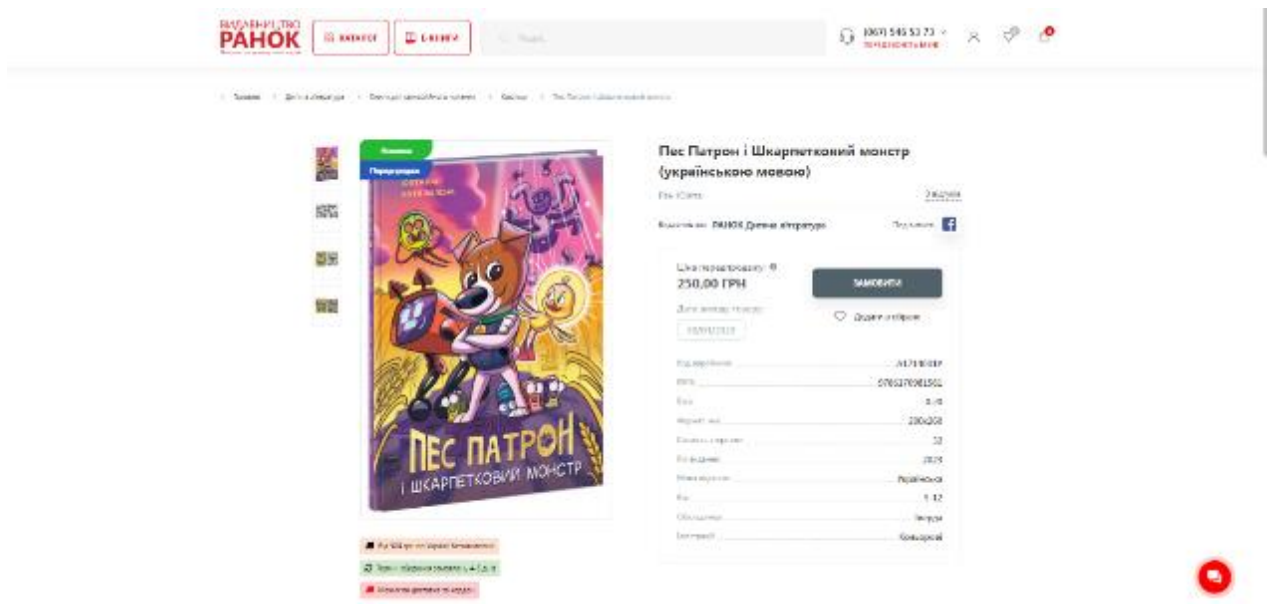


Рисунок 1.7 – Вигляд сторінки з інформацією товару в інтернет-магазині ranok.com.ua

Якщо резюмувати то інтернет-магазин дуже привабливий, він гарно виглядає і має величезний асортимент літератури і книжок. На сайті працюють подарункові картки які можна створювати самостійно та дарувати своїм близьким. Сайт інтернет-магазину має блог в якому ведуться пости новин сайту, статті з життя видавництва і не тільки. На сайті доступна послуга форми подання рукопису на розгляд з подальшою співпрацею з видавництвом. Співпраця проходить з авторами, художниками і перекладачами.

1.5.3. Інтернет-магазин Yakaboo

Yakaboo – найбільший книжковий онлайн-магазин України. Це масштабна книжкова екосистема, що працює з більшістю видавництв країни. [7]

Інтернет-магазин має приємний естетичний дизайн. На головній сторінці сайту (рисунок 1.8) розміщено великий банер, як і у всіх інших книжкових інтернет-магазинах, на якому виводиться актуальна та лімітована

інформація для спрямування зацікавленості користувача та майбутнього покупця на перегляд та затрату більшого часу на сайті з можливим оформленням покупки товару представленого в асортименті інтернет-магазину. Асортимент товарів фокусований не тільки на друковані письмові роботи але налічує аудіо книги, електронні книжки, книжкові аксесуари тощо.

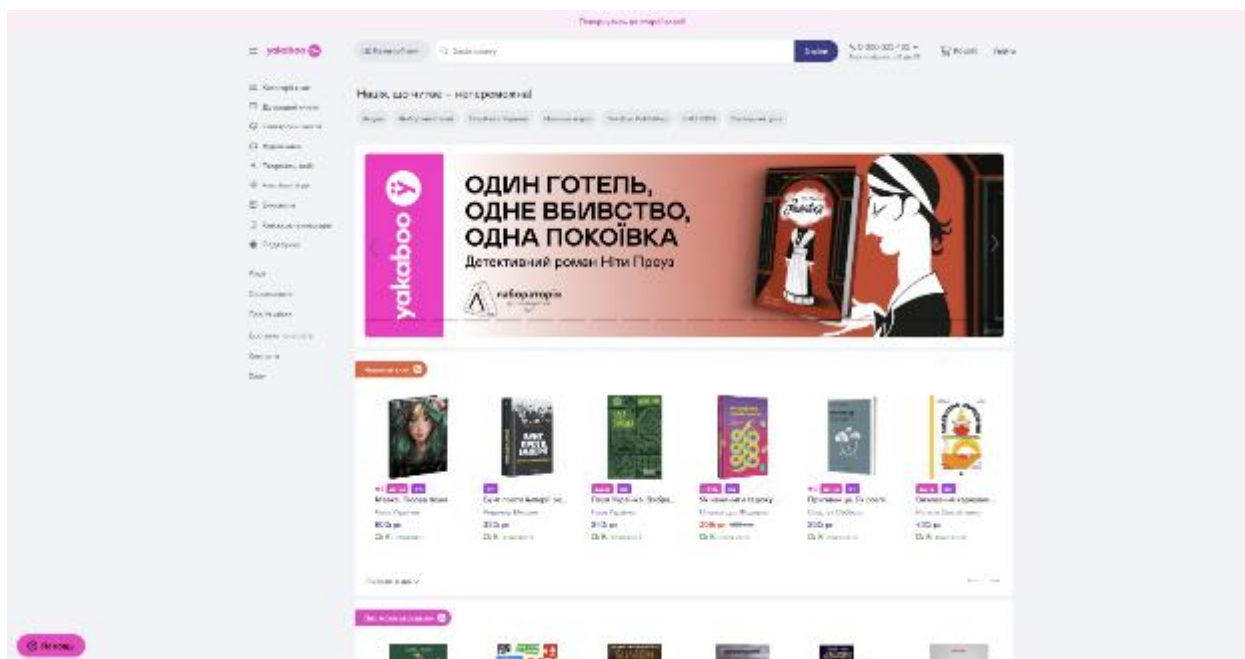


Рисунок 1.8. – Вигляд головної сторінки онлайн-магазину з продажу книжок Yakaboo

Коли покупець переходить на сторінку продукту, який представляється на сайті, він потрапляє на сторінку з інформацією про цей продукт. Покупець отримує обширний контент який розтянутий посередині екрану. Основний контент представляє із себе текст який відцентрований і знаходиться прямо перед очима покупця, фотокартка та блок з цінником знаходиться по бокам від тексту (рисунок 1.9).

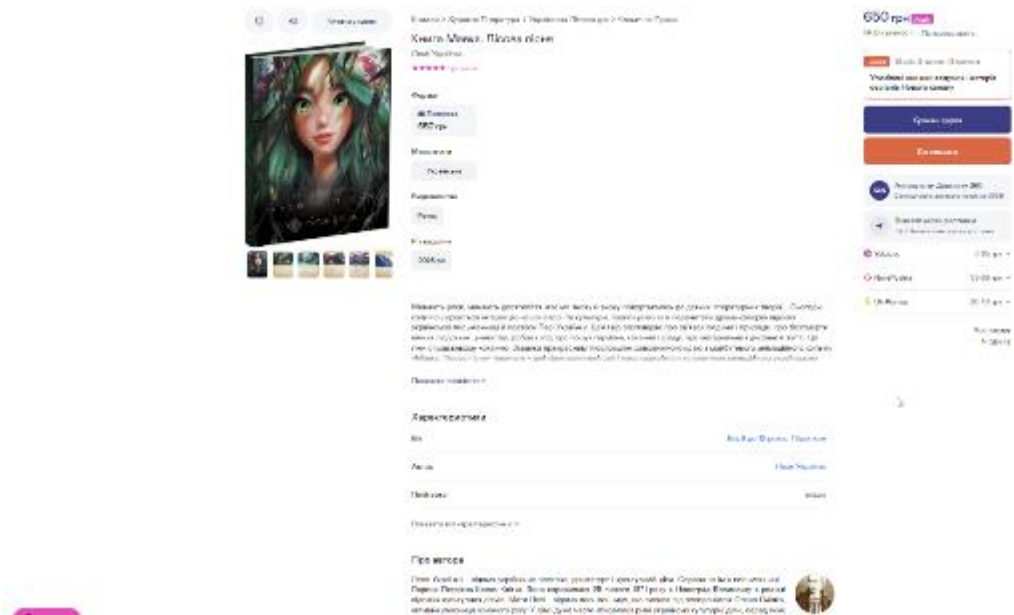


Рисунок 1.9. – Вигляд сторінки з інформацією товару в інтернет-магазині Yakaboo

Навігація по сторінці не відображає нічого зайвого, лише те, що покупцеві потрібно знати про товар, але в деяких місцях занадто багато місця відведено малій кількості інформації та нагромадження тексту яке зразу зустрічає покупця, і цей текст відображає детальний опис товару, його вміст тощо. Також є слайдери, які перераховують продукти, що були переглянуті.

Інтернет-магазин має привабливий і естетичний вигляд, вдалий підбір кольорової гамми, хороший асортимент товарів різного спектру, пов'язаних з літературою. Функціональність не велика, але повністю задовольняє потреби покупця. Можна вважати прикладом дуже грамотної роботи UI-дизайнера: переходи, анімації, слайдери виглядають дуже привабливо. Ще можна відмітити не перегружене меню для навігації і взаємодії з сайтом.

1.6. Висновки за першим розділом

Проаналізувавши типи веб-додатків було прийнято рішення про створення інтернет-магазину за типом кастомного додатку. Архітектура веб-додатку буде виглядати і мати властивості MPA (Multi Page Application).

Провівши огляд вже існуючих книжкових інтернет-магазинів на ринку було виведено основні функції які повинні бути наявні, якісь матимуть високий пріоритет розробки інші додатковий або другорядний. З приведених оглядів існуючих інтернет-магазинів привабливим і бажаним на орієнтування в розробці свого книжкового інтернет-магазину став сайт yakaoo.ua. Він має невибагливий дизайн та функціонал, що втілює в собі простоту і не перекрученість контенту з точки зору покупця.

РОЗДІЛ 2

ІНСТРУМЕНТАЛЬНІ ЗАСОБИ СТВОРЕННЯ ІНТЕРНЕТ-МАГАЗИНУ

2.1. Мова програмування JavaScript

JavaScript(JS) – це мова програмування, яка використовується розробниками для створення інтерактивних веб-сторінок. Від оновлення стрічок соціальних мереж до відображення анімації та інтерактивних карт – можливості JavaScript можуть покращити користувацький досвід веб-сайту. це мова програмування та одна з найфундаментальніших технологій Всесвітньої павутини. Наприклад, карусель зображень, що випадають у меню, які спостерігаються під час перегляду веб-сторінок, і динамічна зміна кольорів елементів на веб-сторінці створюються за допомогою JavaScript.[8]

Простіше кажучи, JavaScript – це багатопарадигмальна мова програмування, яку можна використовувати для широкого спектру завдань.

JavaScript класифікується як скриптова мова програмування з функціями прототипування (підмножина об'єктно-орієнтованої) та динамічної типізації. Окрім прототипування, JavaScript частково підтримує інші парадигми програмування (імперативні та часткові типи функцій), а також деякі особливості відповідної архітектури, зокрема динамічну та слабку типізацію, автоматичне керування пам'яттю, успадкування прототипів та функції як першокласні об'єкти. (elartu.tntu.edu.ua) [9]

JavaScript з боку клієнта – це спосіб роботи JavaScript у браузері. У цьому випадку інструментарій JavaScript знаходиться в коді браузера. Всі найпоширеніші та актуальні веб-браузери мають власні вбудовані інструментарії JavaScript.

Люди які займаються створенням веб-додатків пишуть код JavaScript з використанням різних функцій, пов'язаних з різними подіями які з'являються

на сторінці, такими як клацання миші або наведення курсору тощо. Ці функції вносять зміни в HTML і CSS.

Використання JavaScript з серверної сторони означає використання його у внутрішній логіці сервера. У цьому випадку двигун JavaScript знаходиться безпосередньо на сервері. Серверні функції JavaScript можуть отримувати доступ до баз даних, виконувати різні логічні операції і реагувати на різні події, викликані сервером. Основна перевага серверного програмування на мові JavaScript полягає в тому, що є можливість гнучкого нашалтування відповіді сайту стосовно до вимог які ставляться перед розробником, правами доступу та інформаційним запитам сайту.[10]

2.2. Технології створення серверної частини веб-додатку Node.JS

Node.js (Node) - це JavaScript-платформа з відкритим вихідним кодом, побудована на движку Chrome V8. Вона дозволяє писати програми командного рядка, а також серверний код для веб-додатків і динамічних веб-сторінок. Платформа базується на подієво-керованій моделі з неблокуючими операціями вводу/виводу, що робить її ефективною та легкою.

До Node.js додатки, написані на JavaScript, можна було запускати лише у браузері. З появою платформи з'явилася можливість писати на JavaScript **не тільки в браузері, але й на (repository.kpi.kharkov.ua) сервері.**[11]

Node.js в основному використовується як веб-сервер на серверах, але також може використовуватися для розробки настільних Windows-додатків (за допомогою NW.js, AppJS і **Electron для Linux, Windows і macOS**) (ela.kpi.ua) і мікроконтролерів (tessel, low.js, Espruino і т.д.) На Node.js також можна програмувати на основі керованого подіями, асинхронного (або реактивного) програмування і неблокуючого вводу/виводу.[12]

Браузерний JavaScript використовує веб-інтерфейси API, які надають змогу користуватися об'єктною моделлю документа і інтерфейсу користувача

веб-додатку чи сторінки. JavaScript серверної частини користується API, які дають доступ до файлової системи програми, http-запитів і потоків. Тобто іншими словами Node.JS це технологія використання JavaScript коду на backend частині.

Основні характеристики Node.JS:

- Реалізація на JavaScript – через простоту більша частина програмістів стикалася і працювала з JavaScript, знає принципи роботи та інші базові і просунуті концепції. Тому Node.js легко зрозуміти і вивчити. Крім того, JavaScript також **використовується для розробки (ela.kpi.ua)** клієнтських інтерфейсів у стеку технологій.
- Асинхронний характер – кілька різних процесів можуть працювати паралельно, не блокуючи один одного. Ці можливості дозволяють Node.js працювати дуже швидко і забезпечують кращий користувацький інтерфейс.
- Архітектура керування подіями – на початку виконання операції можна негайно передати функцію, яка буде виконуватися після завершення завдання.
- Однопоточна робота – у Node.js всі запити виконуються в один потік і збираються в циклі обробки подій; ця особливість Node.js усуває перезавантаження запитів і скорочує час обробки, що робить його економічно ефективним у використанні.
- Сумісність із кількома платформами – можливість використовувати на широкому спектрі систем, від Windows до Mac OS, Linux і навіть мобільних платформ.
- Швидка потокова передача даних – завдяки двигуну JavaScript V8 та використання того ж самого двигуна в браузері Google Chrome забезпечується висока поточна передача даних для веб-додатків.[13]

Для розробки та написання серверної частини в добірку к Node.js було використано фреймворк Express. Express – це мінімальний та гнучкий фреймворк для веб-додатків Node.js з відкритим вихідним кодом. При

використанні поверх Node.js він може забезпечити кращу веб-функціональність. Express є найпопулярнішим веб-фреймворком Node.js. Express дозволяє налаштувати веб-додаток, який обробляє кілька HTTP-запитів для заданої URL-адреси. Express є найпопулярнішим веб-фреймворком, оскільки він полегшує розробку веб-додатків, веб-сайтів та API.

Фреймворк має відкритий вихідний код, який надає високий рівень тестового покриття з акцентом на високу продуктивність. Підтримує кілька шаблонів. Може реагувати на певні URL-адреси, має простий механізм для швидкого виявлення помилок у застосунках.[14]

2.3. СУБД PostgreSQL

PostgreSQL – це програмне забезпечення для зберігання даних у (ela.kpi.ua) структурованому вигляді та доступу до них за допомогою SQL-запитів. Цей клас програмного забезпечення також називається RDBMS (Relational Database Management System - система управління реляційними базами даних). Перевагами PostgreSQL є його безкоштовна версія, функціональність і швидкість (для реляційних баз даних).[15]

PostgreSQL підтримує складні типи даних. Насправді, база даних побудована для великої кількості типів даних. Продуктивність бази даних знаходиться на одному рівні з конкурентами, такими як Oracle і SQL Server. AWS надає сервіс бази даних, який повністю підтримує PostgreSQL через Amazon Relational Database Service. Він також використовується для побудови Amazon Aurora.

Однією з причин популярності PostgreSQL є її набір функцій. База даних допомагає розробці додатків, підтримуючи цілісність даних. Адміністратори можуть створювати відмовостійкі середовища. Вона також може використовуватися на різних платформах і підтримує всі поширені мови програмування:

- Python;
- Java;
- Perl;
- .Net;
- Go;
- Ruby;
- C/C++;
- Tcl;
- ODBC.[16]

pgAdmin 4 – pgAdmin 4 - це безкоштовний інструмент адміністрування і управління базою даних (ela.kpi.ua) PostgreSQL з відкритим вихідним кодом. Він надає графічний користувальницький інтерфейс для управління базами даних, таблицями та іншими об'єктами (ela.kpi.ua) і дозволяє користувачам виконувати такі завдання, як створення запитів, генерація звітів і моніторинг (cqr.company) продуктивності.[17]

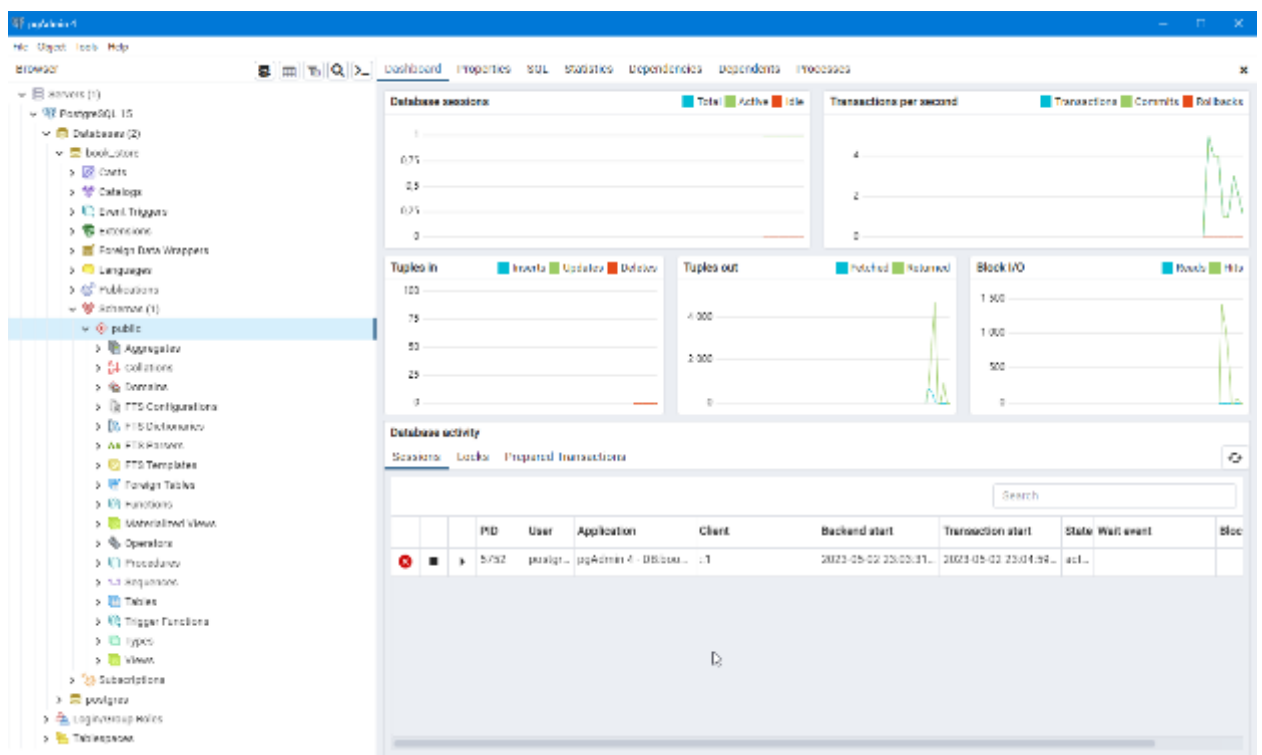


Рисунок 2.3 – Інструмент управління для Postgres pgAdmin4

PostgreSQL підтримує найновіші версії SQL. Корпорація Oracle наразі володіє проектом MySQL, який має кілька преміум-версій, за які користувачі повинні платити. PostgreSQL, з іншого боку, є безкоштовною базою **даних з відкритим вихідним кодом.** (cqr.company) **На відміну від** (ela.kpi.ua) PostgreSQL, MySQL повністю сумісна з ACID тільки при використанні двигунів NDB та InnoDB Cluster Collection.[16]

PostgreSQL трохи складніша за MySQL і може вимагати трохи більше технічних знань для використання; PostgreSQL - це об'єктно-реляційна база даних, яка заповнює прогалину між реляційним програмуванням та об'єктно-орієнтованим кодуванням PostgreSQL - це об'єктно-реляційна база даних, що означає, що її можна використовувати для визначення об'єктів та успадкування таблиць, створюючи таким чином більш складні системи баз даних.

MySQL ефективна в ситуаціях, коли швидкість читання є єдиним фактором, наприклад, в OLAP і OLTP-додатках. У той же час, продуктивність PostgreSQL ідеально підходить для додатків, які повинні обробляти складні пошукові запити. MySQL більш залежна при роботі з проектами бізнес-аналітики, які, як правило, більш складні для читання. З іншого боку, PostgreSQL краще працює в додатках для аналітики та сховищ даних. [18]

PostgreSQL – це, мабуть, одна з найдосконаліших **реляційних баз даних** (ela.kpi.ua) з відкритим вихідним кодом, і вона безкоштовна. Серед її численних переваг – сумісність з багатьма системами і здатність працювати з багатьма популярними мовами програмування і проміжним програмним забезпеченням. База даних також приваблива тим, що вона безкоштовна. Вона також підтримує дані у форматі JSON, що є необхідним для деяких проектів. [16]

2.4. ORM **Sequelize**

Sequelize – це інструмент для організації взаємодії між платформою Node.js та реляційними базами даних без використання спеціальної мови запитів SQL. Sequelize належить до Object Relational Mapper (ORM) і використовує СУБД (Postgres, MySQL, MariaDB, SQLite та Microsoft SQL Server) для створення віртуальних об'єктних баз даних з об'єктами JavaScript. (ela.kpi.ua) [19]

ORM є кращими, оскільки дозволяють взаємодіяти з базою даних мовою додатків (JavaScript), тобто без використання спеціально розробленої мови (SQL). Однак є випадки, коли запити до бази даних (repository.kpi.kharkov.ua) простіше виконати за допомогою SQL (або їх можна виконати тільки за допомогою SQL). [20]

Модель Sequelise – це абстракція, яка представляє таблицю бази даних. Вона (ela.kpi.ua) повідомляє системі свою назву, кількість стовпців і тип інформації, що міститься в ній. Кожна модель має власну назву. У більшості випадків (ela.kpi.ua) вона збігається з назвою таблиці.

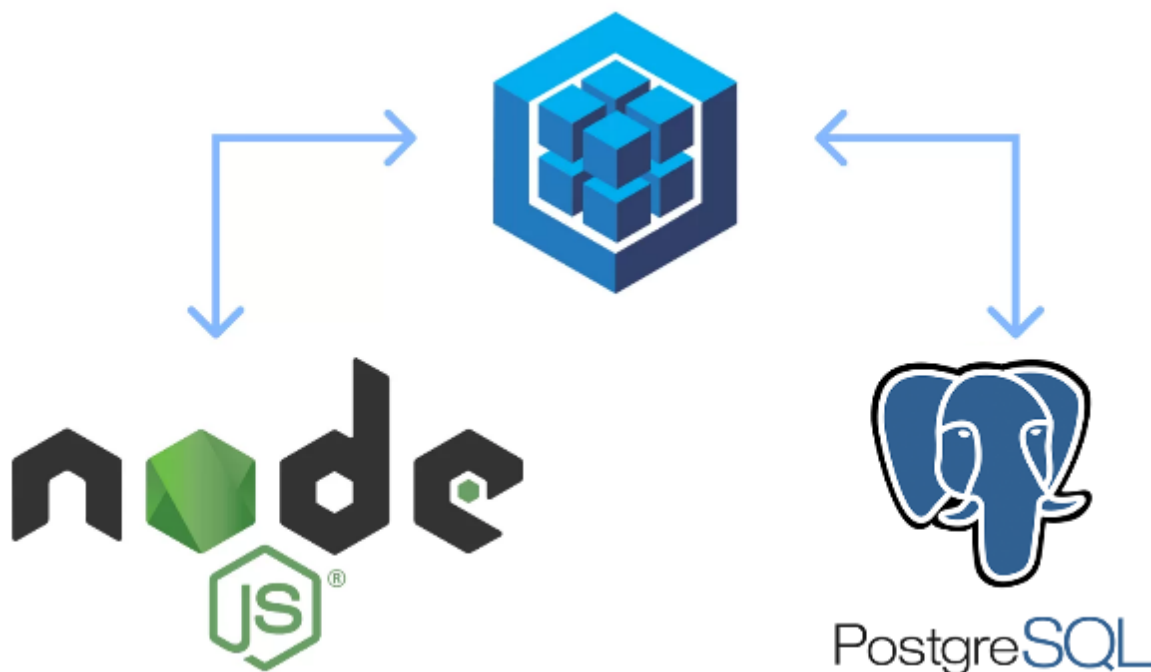


Рисунок 2.4 – Sequelize діє як маппер між базою даних SQL та додатком.

[19]

Наприклад, якщо ім'я таблиці – Accounts (множина), то ім'я моделі - Account (однина).

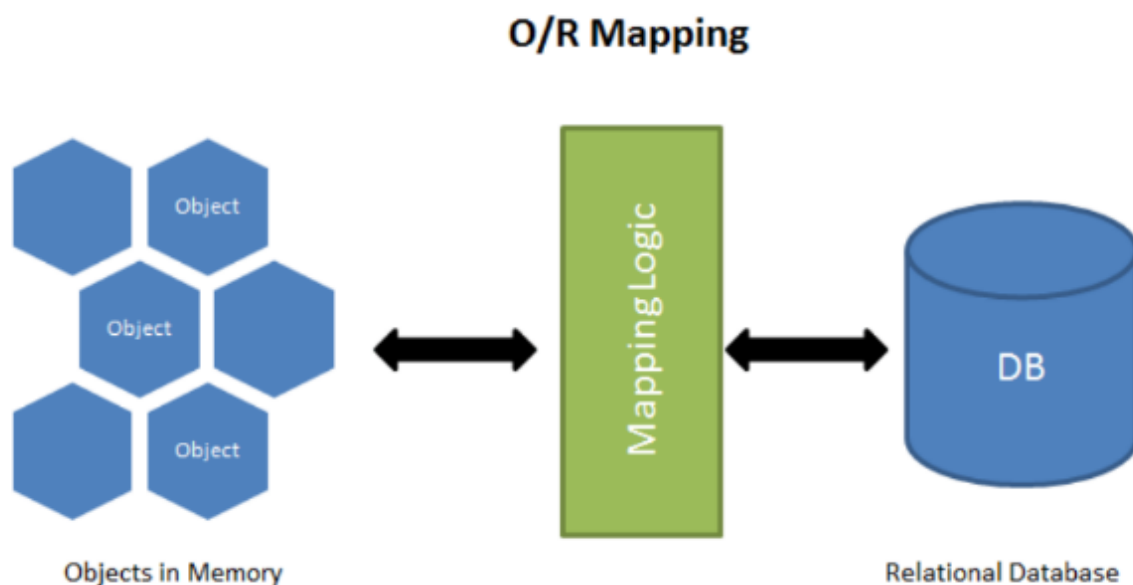


Рисунок 2.5 – Sequelize самостійно зіставляє об'єкти в пам'яті з записами в базі даних SQL.[20]

2.5. Бібліотеки React

2.5.1. React Router

Декларативна маршрутизація - це стиль кодування, що використовується в React та React Router. Декларативні маршрути в React є компонентами і використовують ту ж інфраструктуру, що і будь-який React-додаток. Ці маршрути використовують потужний рушій рендерингу та умовну логіку React, щоб програмно відкривати та закривати маршрути, пов'язуючи веб-адреси з певними сторінками або іншими компонентами.[21]

React Router DOM – це вузловий модуль, призначений для маршрутизації у веб-додатках. Він дозволяє інженерам створювати односторінковий маршрут React-додатку досить просто.

BrowserRouter, Route, Switch і Link є компонентами React-Router. Ці компоненти поділяються на три категорії.

Перша категорія – це маршрутизатори, такі як <BrowserRouter>; друга категорія – маршрутизатори маршрутів, такі як <Route> і <Switch> і остання категорія – навігація, такі як <Link> і <Redirect>.

- BrowserRouter – це додаток-маршрутизатор, який дозволяє здійснювати маршрутизацію в React **за допомогою HTML5 (ela.kpi.ua)** API, включаючи події pushState, replaceState і popState для синхронізації користувацького інтерфейсу з URL-адресою. Це основний компонент, який використовується для зберігання всіх інших компонентів і використовує звичайні URL-адреси.
- Route – це умовний компонент, який відображає компоненти на основі певної URL-адреси або URL-адреси, на яку вказує компонент. Іншими словами, це компонент, який відображає деякі користувацькі інтерфейси, коли шлях збігається з поточною URL-адресою.
- Link – компонент посилань використовуються для створення посилань на різні шляхи і реалізації навігації в додатку; посилання приймають властивість, яка вказує, куди посилання повинно привести користувача.
- Компонент Switch використовується для обробки лише першого маршруту, який відповідає місцезнаходженню, а не всіх маршрутів, які відповідають місцезнаходженню.

На додаток до цих потужних компонентів, є кілька дуже корисних хуків, які дійсно можуть допомогти, надаючи додаткову інформацію, яка може бути використана компонентом. Це: useHistory, useParams і useLocation.

useHistory згідно з документацією до react router, надає доступ до екземпляру історії, який можна використовувати для навігації. Через об'єкт історії можна отримати доступ до поточного стану історії браузера та керувати ним.

UseParam – хук, який повертає об'єкт пари ключ-значення для параметра URL. Ключем є ім'я параметра, а значенням – поточне значення параметра. Іншими словами, він надає доступ до параметрів пошуку за URL-адресою.

UseLocation – хук, еквівалентний useState, але повертає нове місце розташування при зміні URL-адреси. Простіше кажучи, хук useLocation повертає об'єкт розташування, що представляє поточну URL-адресу.[22]

2.5.2. React Mobx

Основна мета MobX – поліпшити управління станами для розробника і створити більш простий спосіб управління станом Javascript-додатків з меншою кількістю коду і шаблонів. MobX використовує спостережувані дані, щоб допомогти автоматично відстежувати зміни, полегшуючи життя розробникам. **На відміну від (ela.kpi.ua)** інших менеджерів станів, таких як Redux або Easy Peasy, MobX використовує декілька сховищ для керування станом програми. MobX легко і просто встановлюється, вимагає дуже мало шаблонного коду і не потребує спеціальних інструментів.

Як і інші подібні бібліотеки (наприклад, Redux, Recoil, Hook states), MobX дозволяє керувати станом вашої програми, але вона відрізняється простотою та масштабованістю.

Mobx розрізняє такі концепції:

- State (стан);
- Actions (дії);
- Derivations (похідні).

State State – це дані, які керують програмами. Вони містять різні типи даних, починаючи від масивів, рядків, чисел та об'єктів, з якими MobX дозволяє працювати. Все, що потрібно зробити, це переконатися, що всі властивості доступні для спостереження, щоб MobX міг їх відстежувати.

Action – це будь-який блок коду, який може змінювати такі дані: події користувача, внутрішні дані і т.д. Action схоже на людину, яка змінює дані в осередку електронної таблиці. Actions допомагають структурувати блок коду і запобігають постійній зміні стану, коли в цьому немає необхідності. Методи, які змінюють стан, називаються actions в MobX.

Derivations – те що отримано зі стану. Воно існує у різних формах, є такі види похідних MobX:

- Computed Values – можна відновити зі стану за допомогою функції clean, вони автоматично оновлюються MobX, а також призупиняються, коли не використовуються.
- Reactions – схоже на Computed Values, але з різницею **в тому що** ela.kpi.ua реагує на зміну стану, але натомість викликає побічні ефекти.[23]

2.5.3. React-Bootstrap

React-Bootstrap – це багаторазова бібліотека React-компонентів, яка реалізує функціональність популярного шаблону Bootstrap від Twitter і заробила майже 11 000 зірок на GitHub. Її простота та легкість у використанні, ймовірно, є причинами її популярності.[24]

Популярність односторінкових додатків за останні кілька років призвела до появи багатьох фронтенд-фреймворків, таких як Angular, React, Vue.js та Ember. Як наслідок, jQuery більше не є обов'язковим для створення веб-додатків. Сьогодні React є найпоширенішим JavaScript-фреймворком для створення веб-додатків, а Bootstrap став найпопулярнішим CSS-фреймворком.

Існує декілька способів додати Bootstrap до React-додатку. Три найпоширеніші способи:

- використання Bootstrap CDN;
- зробити Bootstrap залежним;

- пакет React Bootstrap.

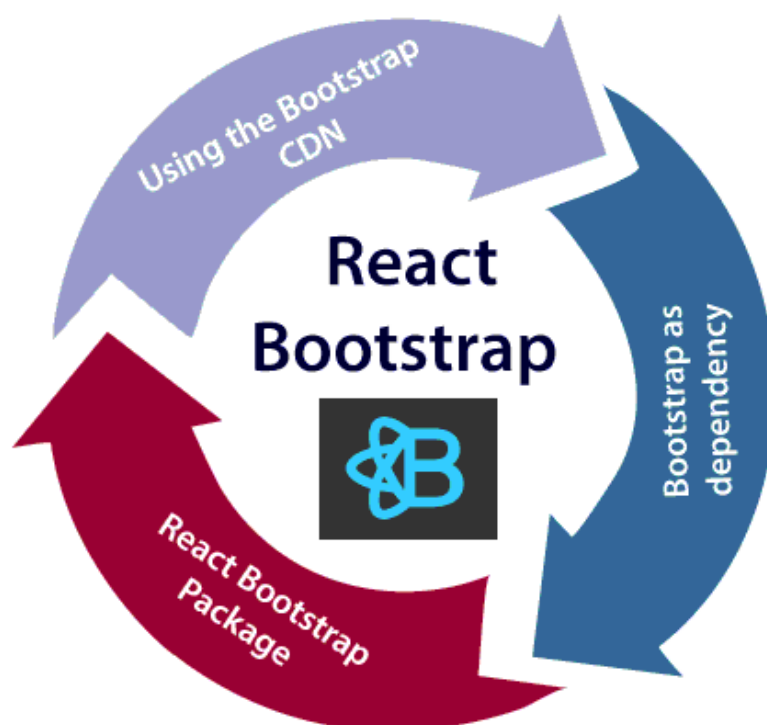


Рисунок 2.6 – Способи додавання Bootstrap до React-додатку.

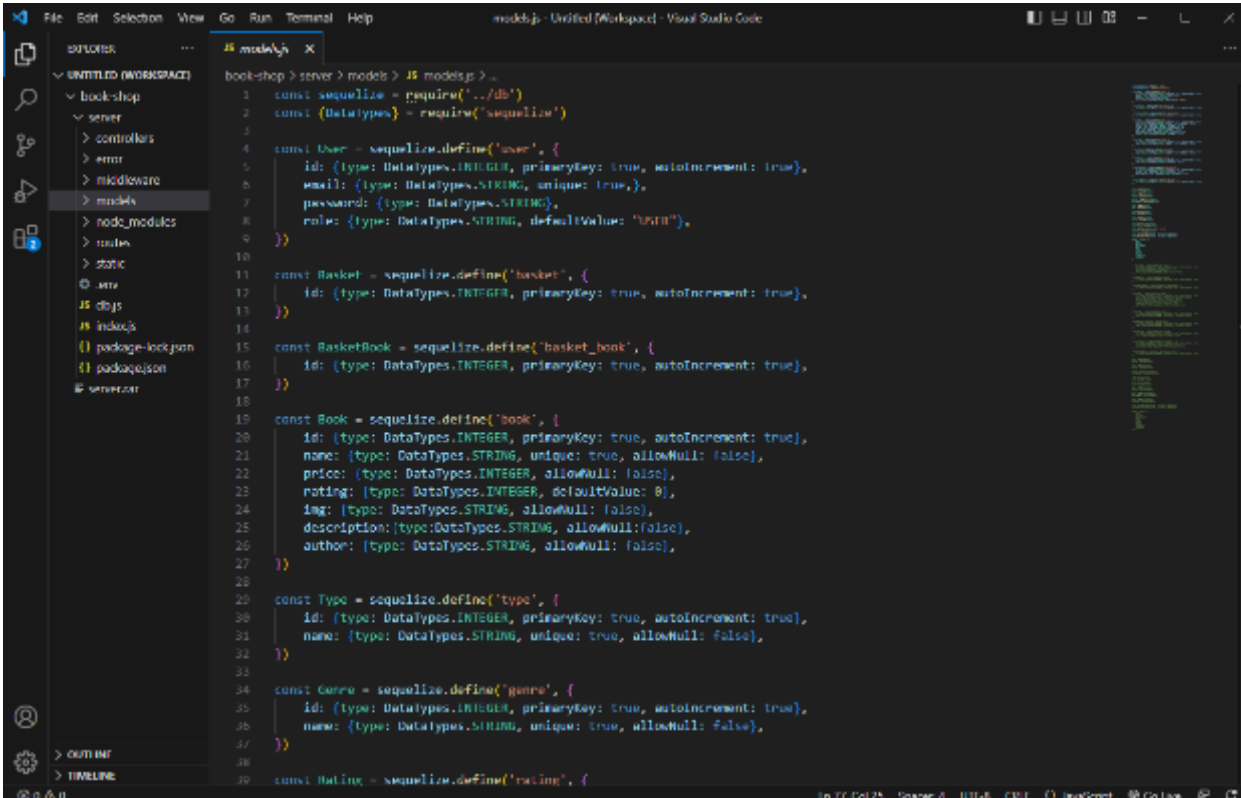
Пакети React Bootstrap – це найпоширеніший спосіб додати Bootstrap до React-додатків. Існує багато Bootstrap-пакетів, створених спільнотою, які мають на меті реструктуризувати Bootstrap-компоненти у React-компоненти. Два найпопулярніші пакети Bootstrap:

- react-bootstrap: Повна переробка Bootstrap-компонентів у React-компоненти без необхідності використання таких залежностей, як Bootstrap.js або jQuery. Після встановлення React setup та React-Bootstrap, у вас є все необхідне.
- reactstrap: Бібліотека, що містить компоненти React Bootstrap 4, з акцентом на композицію та контроль; вона не базується на jQuery або Bootstrap JavaScript. Однак, react-popover необхідний для розширеного компонування контенту, такого як підказки, спливаючі вікна та авто-спливаючі вікна.[25]

2.6. Середовище розробки Visual Studio Code

Visual Studio Code – один з найпопулярніших редакторів коду, розроблений компанією Microsoft. Він має відкритий доступ та безкоштовний і сумісний з усіма сучасними операційними системами. Visual Studio Code – це звичайний текстовий редактор, який дозволяє підключати різні плагіни, тому для розробки будь-якого коду можна використовувати різні мови програмування.[26]

Visual Studio Code підтримує такі мови, як JavaScript, Python, CSS, HTML і Java. Він надає змогу автоматичного завершення, аналізу коду на ходу, навігацію в коді, рефакторинг та інтеграцію з SCM. Ключові переваги інтегрованого середовища розробки є вбудований додаток, який дозволяє користувачам завантажувати плагіни та інші речі для **коду Visual Studio**, не виходячи з програми, а редактор має вбудовану функцію налагоджувача, **інструменти для роботи з Git'ом**, автозавершення типового синтаксису та контекстно-залежну довідку.



```
book-shop > server > models > JS models.js > ...
1  const sequelize = require('../db')
2  const {DataTypes} = require('sequelize')
3
4  const User = sequelize.define('user', {
5    id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
6    email: {type: DataTypes.STRING, unique: true},
7    password: {type: DataTypes.STRING},
8    role: {type: DataTypes.STRING, defaultValue: 'USER'},
9  })
10
11 const Basket = sequelize.define('basket', {
12   id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
13 })
14
15 const BasketBook = sequelize.define('basket_book', {
16   id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
17 })
18
19 const Book = sequelize.define('book', {
20   id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
21   name: {type: DataTypes.STRING, unique: true, allowNull: false},
22   price: {type: DataTypes.INTEGER, allowNull: false},
23   rating: {type: DataTypes.INTEGER, defaultValue: 0},
24   img: {type: DataTypes.STRING, allowNull: false},
25   description: {type: DataTypes.STRING, allowNull: false},
26   author: {type: DataTypes.STRING, allowNull: false},
27 })
28
29 const Type = sequelize.define('type', {
30   id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
31   name: {type: DataTypes.STRING, unique: true, allowNull: false},
32 })
33
34 const Genre = sequelize.define('genre', {
35   id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
36   name: {type: DataTypes.STRING, unique: true, allowNull: false},
37 })
38
39 const Rating = sequelize.define('rating', {
```

Рисунок 2.7. – Інтерфейс середовища розробки Visual Studio Code (moodle2.snu.edu.ua)

Середовище розробки дозволяє взаємодіяти з платформами ASP.NET та Node.js і позиціонується як легке рішення, яке можна використовувати без повністю інтегрованого середовища розробки. [27]

2.7 Висновки за другим розділом

У даному розділі було розглянуто основні програмні засоби, які (docplayer.net) можуть бути використані для (repository.kpi.kharkov.ua) розробки проекту, що дало змогу здійснити оптимальний підбір програмного інструментарію. Зокрема, в якості мови програмування було вирішено обрати багатопарадигмальну мову JavaScript, що забезпечує вирішення широкого спектру завдань. Для створення серверної частини на JavaScript оптимальним вибором виглядає платформа Node.js. Системою управління базами даних можна обрати одну з найдосконаліших безкоштовних реляційних баз даних з відкритим вихідним кодом (ela.kpi.ua) – PostgreSQL.

Для створення проекту буде цілком достатньо функціоналу середовища розробки Visual Studio Code, розробленого компанією Microsoft.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ "ІНТЕРНЕТ-МАГАЗИН З ПРОДАЖУ КНИЖОК"

3.1. Призначення та функціональні можливості додатку

Веб-магазин надає можливість онлайн, через браузер користувачам та потенційним покупцям ознайомитися з зацікавившим товаром, при цьому не будучи в магазині фізично. Ознайомитися з інформацією стосовно вибраного покупцем товару, та його ціною та з додатковими подробицями які очікуються під час замовлення товару. Можливість додати потенційний товар до кошика і перемикнутися назад на ознайомлення з іншим каталогом різних книжок. Книжки та інші рукописи в магазині поділені на типи та жанри які знайдуть свого потенційного читача.

Інтернет-магазин представляє із себе основну або інакше кажучи домашню сторінку яка є каталогом, на якому розміщується як сам список товарів які представлені у магазині так і інші компоненти сторінки, такі як основний банер, секція з блогами, тощо. В магазині є дві сутності або перефразувавши по іншому існують такі ролі як ADMIN і USER, і вся робота реалізованого функціоналу в додатку крутиться навколо цих сутностей. Для того щоб (ela.kpi.ua) магазин розумів, яка сутність перебуває на сайті є можливість авторизування в магазині для зареєстрованих користувачів, які за замовчуванням є звичайними користувачами з обмеженими можливостями в додатку. У адміністратора вже існує обліковий запис який має розширені повноваження такі, яких немає у звичайного користувача а адміністратор навпроти може робити все теж саме що і звичайний покупець інтернет-магазину.

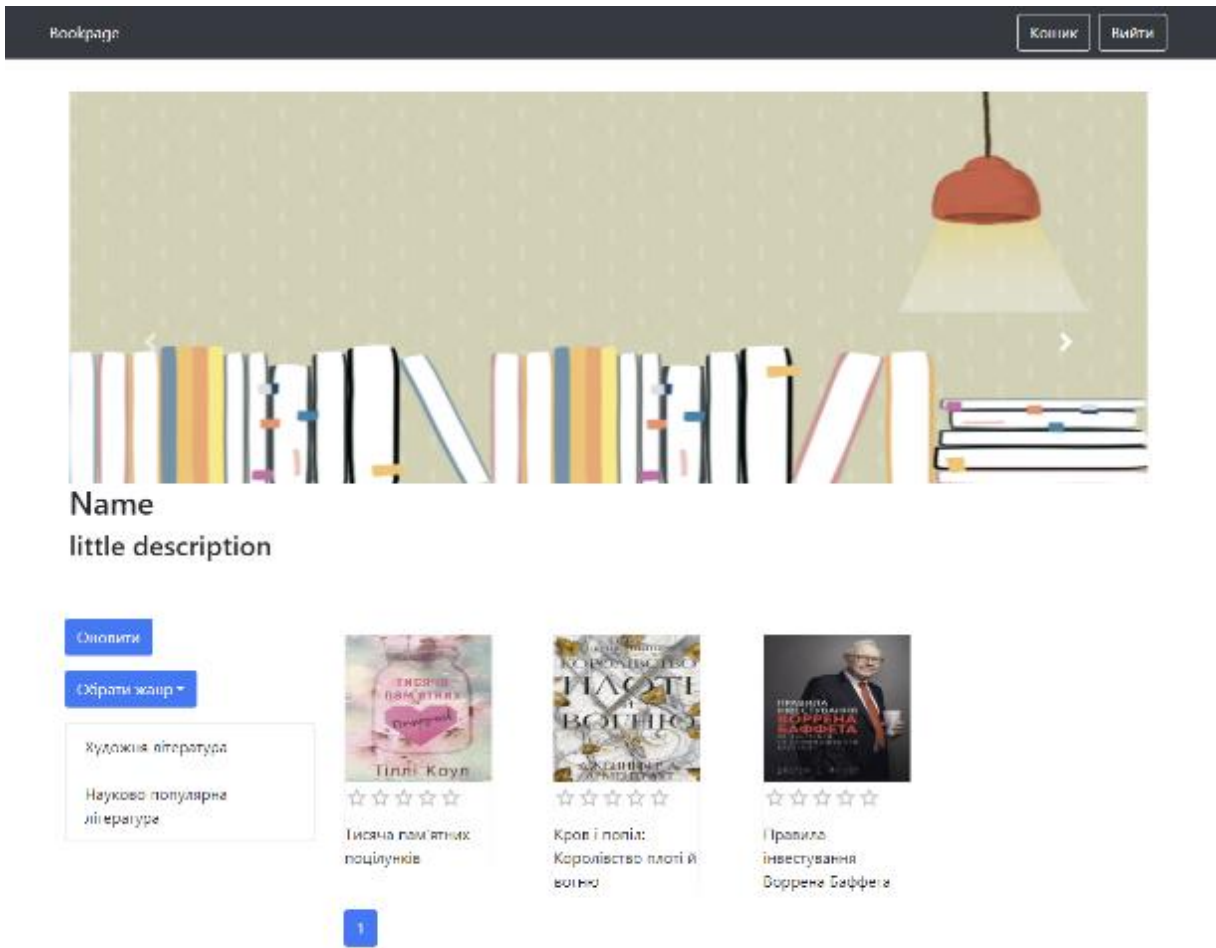


Рисунок 3.1. – Зображення зовнішнього вигляду інтернет магазину

3.2. Структура бази даних

Під час аналізу та створення концепції виконання завдання було прийнято необхідність у створенні таблиць Books, Users, Genres, Types, Book_infos, Baskets, Baskets_books, Rating та Blogs. Зв'язки між ними зображені на рисунку 3.2.

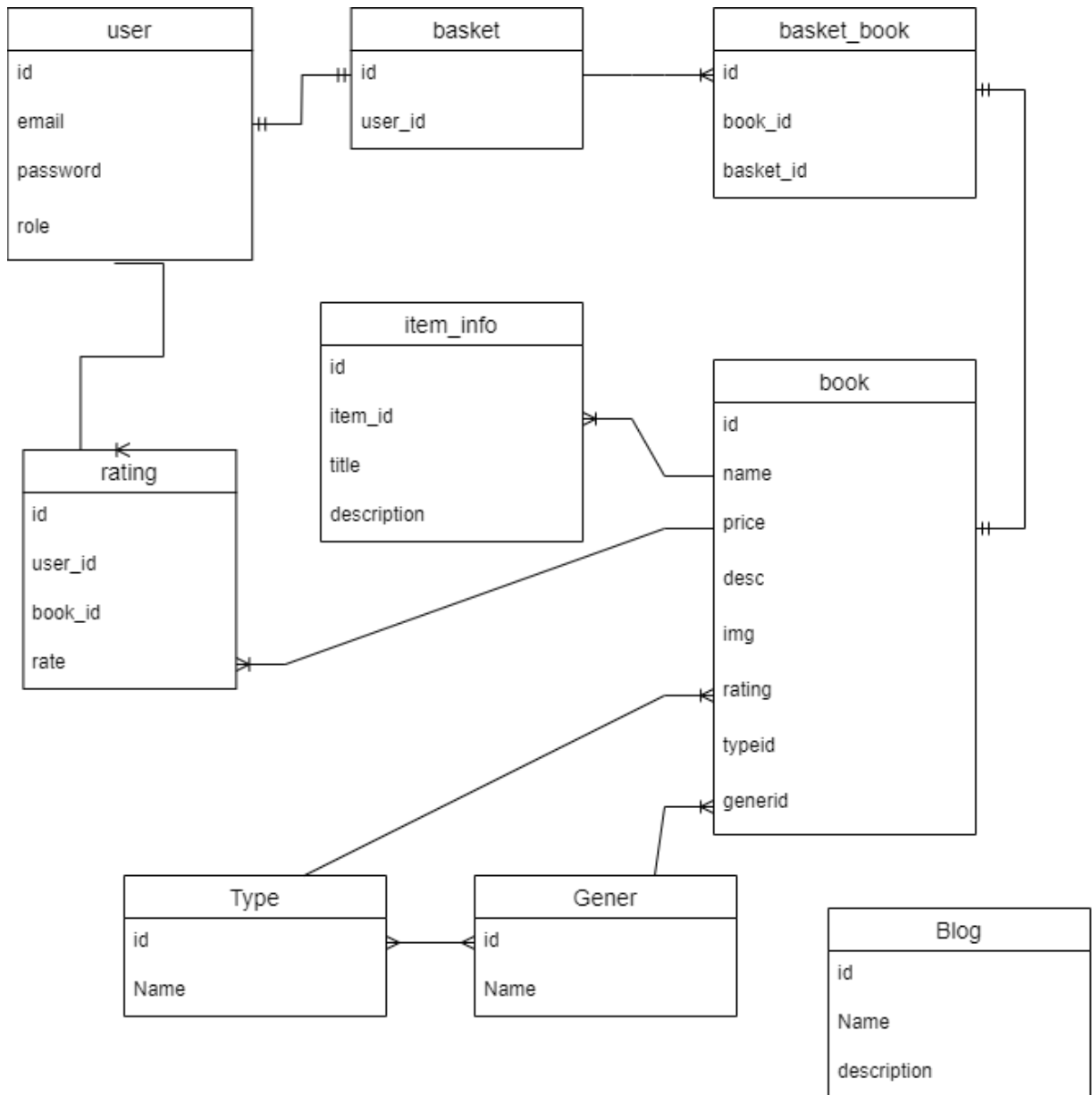


Рисунок 3.2. – Діаграма зв'язків таблиць у базі даних

Таблиця Books одна з основних таблиць в базі даних, вона має велику кількість колонок які відповідають і мають суть до унікальної потреби.

Id: ця колонка несе в собі унікальний код за яким проходить виконання багатьох функцій та методів.

Name: колонка з назвою книжки.

Price: колонка ціни.

Rating: колонка яка відповідає за рейтинг книжки

Bookdescription: колонка яка несе у собі короткий або об'ємний опис книжки

Author: Колонка яка відповідає за ім'я автора

TypeId: В цій колонці унікальний ідентифікатор типу книги який їй приписується при створенні книги, а номер ідентифікатора береться з таблиці яка містить номери та назви можливих типів книг.

GenreId: В цій колонці унікальний ідентифікатор жанру книги який їй приписується при створенні книги, а номер ідентифікатора береться з таблиці яка містить номери та назви можливих жанрів книг.

Таблиця users одна з головних таблиць в базі даних бо вона містить інформацію користувача інтернет-магазину.

Id: ця колонка несе в собі унікальний код за яким проходить виконання багатьох функцій та методів.

Email: ця колонка містить електронну пошту користувача або інакше кажучи логін за яким він авторизується в магазині.

Password: колонка зберігає в собі пароль користувача, але замість звичайного зберігання паролю наприклад «12345» а зберігається в форматі jwt токєну в якому вшити пароль.

Role: колонка яка містить роль користувача, за замовчуванням є 2 види користувачів «ADMIN» та «USER».

Таблиця genres містить в собі унікальні номери ідентифікації та назви жанрів які будуть використовуватися при пошуку, створенні книжки тощо.

Id: ця колонка несе в собі унікальний код жанру за яким проходить виконання багатьох функцій та методів.

Name: колонка в якій записується назва жанру який в подальшому буде приписаний до книги і за яким буде проходити фільтрація.

Таблиця types містить в собі унікальні номери ідентифікації та назви типів які будуть використовуватися при пошуку, створенні книжки тощо.

Id: ця колонка несе в собі унікальний код типу за яким проходить виконання багатьох функцій та методів.

Name: колонка в якій записується назва типу який в подальшому буде приписаний до книги і за яким буде проходити фільтрація.

Таблиця `book_infos` містить в собі додаткову інформацію яку або розширену викладку деталей яку хотів би знати покупець перед купівлею мову книжки, кількість сторінок тощо.

`Id`: ця колонка несе в собі унікальний код типу за яким проходить виконання багатьох функцій та методів.

`Title`: колонка в якій записується назва типу додаткової інформації чи характеристики (видавництво, кількість сторінок, обкладинка тощо).

`Description`: колонка описується значення додаткової інформації.

`BookId`: колонка містить унікальний ідентифікатор книжки до якої було занесено та або інша додаткова інформація.

Таблиця `types` містить в собі унікальні номери ідентифікації користувачів та їх кошків.

`Id`: ця колонка несе в собі унікальний код типу за яким проходить виконання багатьох функцій та методів.

`UserId`: колонка в якій приписується унікальний ідентифікатор користувача.

Таблиця (ela.kpi.ua) `types` містить в собі унікальні номери ідентифікації та назви типів які будуть використовуватися при пошуку, створенні книжки тощо.

`Id`: ця колонка несе в собі унікальний код типу за яким проходить виконання багатьох функцій та методів.

`BasketId`: колонка містить унікальний ідентифікатор кошика користувача якій додає книжку у свій кошик.

`BookId`: колонка містить унікальний ідентифікатор книжки яка була занесена до кошика.

Таблиця `rating` містить в собі рейтингові оцінки користувачів стосовно книжок.

`Id`: ця колонка несе в собі унікальний код типу за яким проходить виконання багатьох функцій та методів.

`Rate`: колонка містить оцінку книги користувача.

UserId: колонка містить унікальний ідентифікатор користувача який оцінив книгу.

BookId: колонка містить унікальний ідентифікатор книжки яка була оцінена користувачем.

Таблиця blogs містить в собі коротко-новостну інформації стосовно світу книжок або інтернет-магазину.

Id: ця колонка несе в собі унікальний код типу за яким проходить виконання багатьох функцій та методів.

Name: колонка містить заголовок назву блогу.

Description: колонка містить інформацію яка відноситься до заголовку блогу та несе опис назви тощо.

Лістинг коду створення моделі та типів даних наведено у додатку В.

3.3. Інтерфейс користувача

Інтерфейс користувача досить простий та має дуже невибагливий функціонал. Основні компоненти, які повинні бути доступні користувачеві – це перегляд та фільтрація з можливістю додавання товару до кошику. На рисунку 3.3 можна побачити головний банер з властивостями слайдера, який перемикається з деяким інтервалом і завмирає при наведенні курсора на будь-який елемент банеру на якому міститься новини зі світу книжок або події які проходять в інтернет-магазині, такі як розпродаж, тощо.

Нижче головного банеру розташовано основні компоненти інтернет-магазину такі як каталог товарів та функціонал фільтрації для пошуку потенційного товару для купівлі.

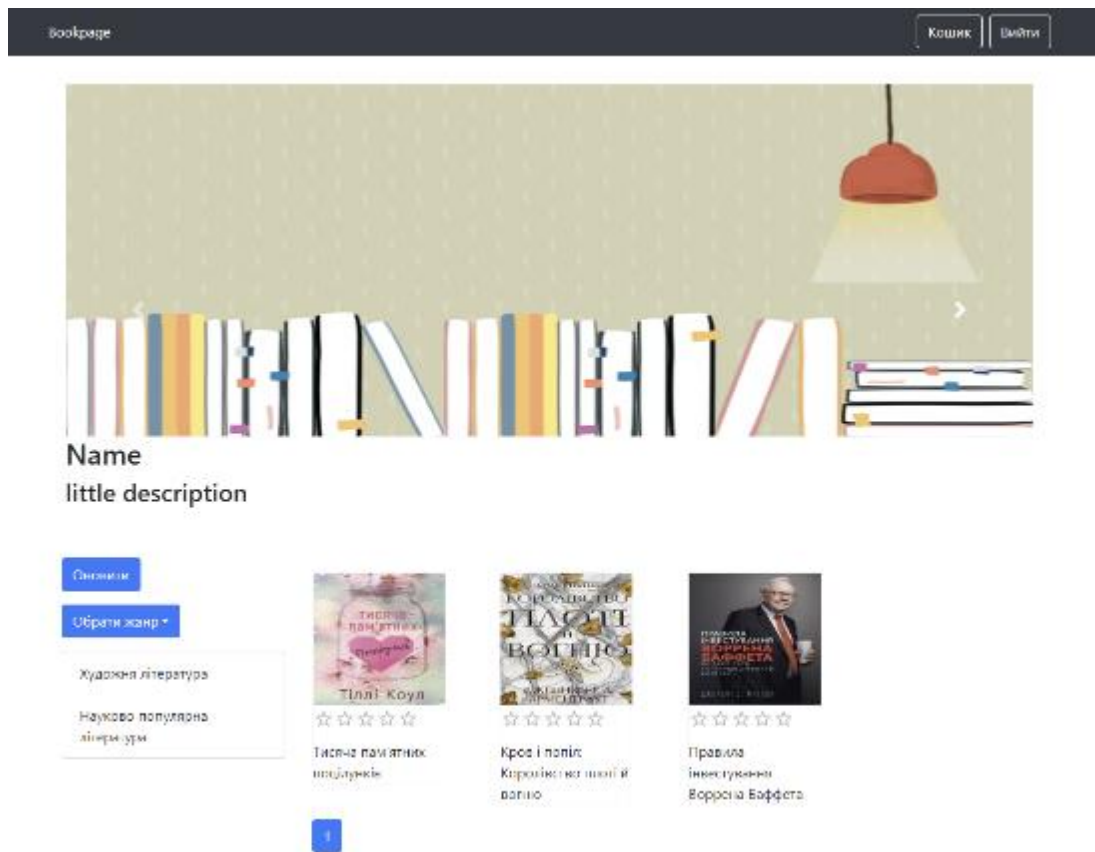


Рисунок 3.3. – Зображення основної сторінки веб-додатку

Зліва можна побачити компоненти фільтрації товару в каталозі інтернет-магазину такі як опція вибору жанру яка реалізована в вигляді випадаючого меню (рисунок 3.4).

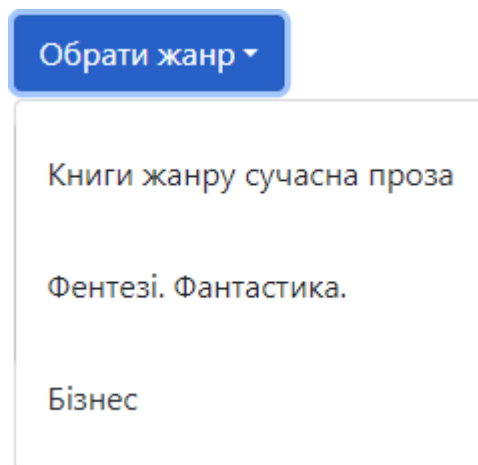


Рисунок 3.4. – Зображення випадаючого меню жанрів для пошуку

Можливість вибору типу має вигляд списку найменувань типів письмових товарів (рисунок 3.5).

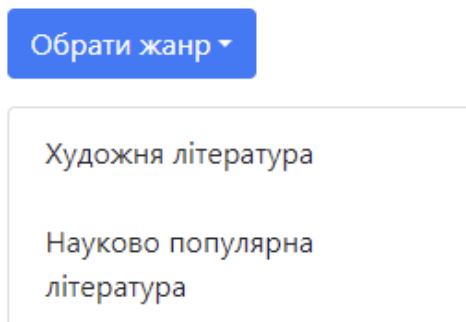


Рисунок 3.5. – Зображення списку із назв типів для пошуку

Також в верхній частині інтернет-магазину в навігаційній частині є кнопка «Кошик» яка при натисканні веде на сторінку кошика на якій з'являються продукти які були додані користувачем під час серфінгу товарів представлених у каталозі. Сторінка кошика представляє із себе список товарів з їх зображеннями, назвою та ціною. Присутній компонент який рахує суму товарів які знаходяться у кошику (рисунок 3.6).

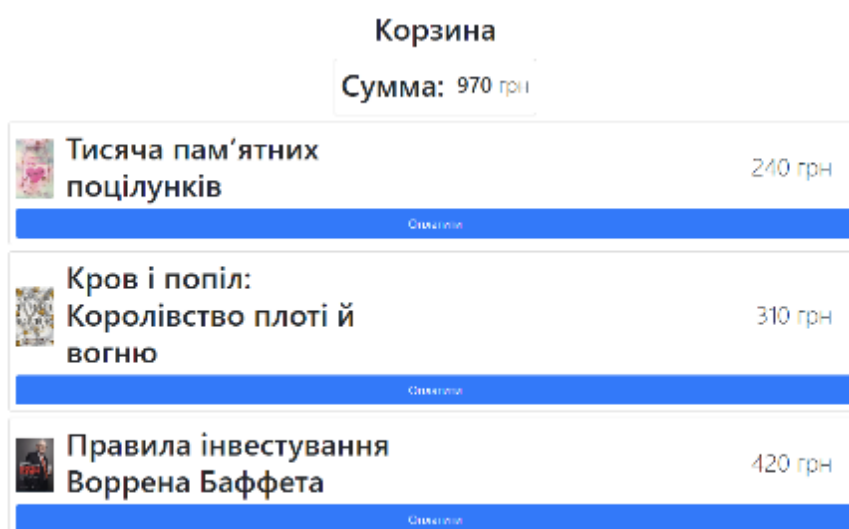



Рисунок 3.5. – Зображення списку із назв типів для пошуку

Після натискання на товар у каталозі, користувач переходить на сторінку товару, яким він був зацікавлений. Дизайн був створений під час огляду вже існуючих аналогів інтернет-магазинів книжок, і на основі оформлення сторінок товарів сайтів аналогів було розроблено концепцію дизайну, де на головне місце ставиться вигляд товару та опис, а коли користувач хоче стати потенційним покупцем книги і захоче додати невибагливий товар до своєї кошички то він зможе ознайомитися з додатковими деталями стосовно товару, його року видання, мови тощо. Дизайн є дуже мінімалістичним і спрямований виділити основні компоненти на сторінці з якими повинен ознайомитися користувач при перегляді сторінки з товаром який зацікавив його.

← Назад

Тисяча пам'ятних поцілунків
Тіллі Коул
☆☆☆☆☆

240 грн.
[Додати до кошика](#)

Кількість сторінок: 384
Видавництво: BookChef
Рік видання: 2022
Мова: Українська

Опис

Чи може кохання тривати вічно, попри перешкоди, відстань і час? Це можливо, якщо поєднуються дві половинки одного серця. Руну п'яти, він із батьками щойно переїхав із рідної Норвегії до маленького містечка американського штату Джорджія і дуже засмучений через це. Та підколи він знайомиться з дивакуватою сусідкою Поппі, його життя набуває яскравих кольорів і перетворюється на справжню пригоду. Поппі, виконуючи заповіт улюбленої бабусі, має на меті зібрати тисячу поцілунків, але не звичайних, а особливих, таких, коли серце мало не вистрибуватиме з грудей. Саме такі поцілунки обіцяє їй подарувати Рун. Історія кохання, яка розіб'є вам серце.

Рисунок 3.5. – Зображення сторінки товару інтернет-магазину

3.4. Адміністративна панель

Інтерфейс адміністратора має функціонал для створення та додавання товарів, жанрів тощо. Він майже не відрізняється від інтерфейсу та можливостей які надані звичайному користувачу, тільки ці можливості більш розширені, якщо проводити порівняння. Щоб потрапити до інтерфейсу адміністратора, потрібно натиснути кнопку, яка відповідає за панель адміністратора, що знаходиться в навігаційній панелі зверху, зліва від кошика (рисунок 3.6).

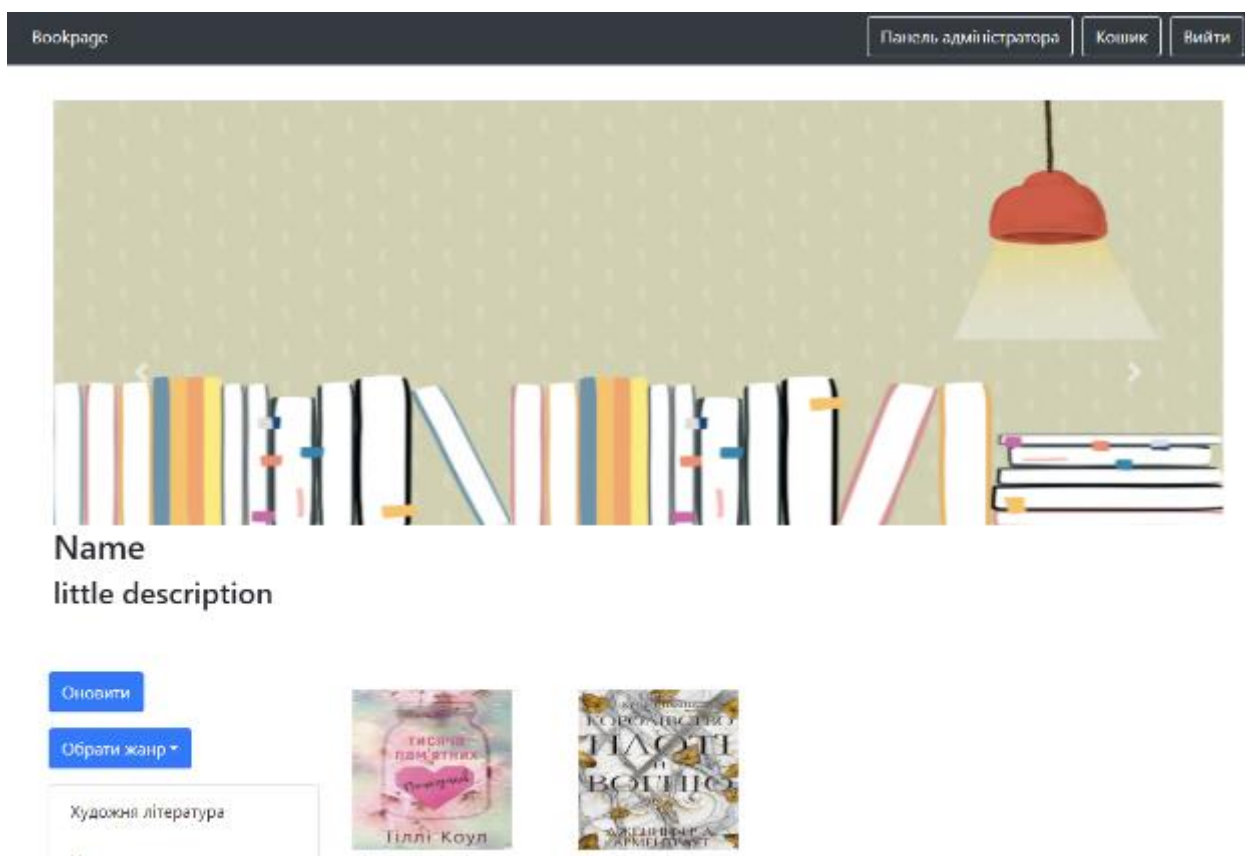


Рисунок 3.6. – Зображення вигляду головної сторінки для ролі адміністратора

Панель адміністратора представляє із себе декілька модальних вікон, з якими проходить взаємодія адміністратора з інтернет-магазином. Представленими можливостями для адміністратора є додавання типу, жанру, книги та блогу (рисунок 3.7, 3.8, 3.9, 3.10, 3.11).

Додати тип
Додати жанр
Додати книгу
Додати блог

Рисунок 3.7. – Зображення вигляду панелі адміністратора

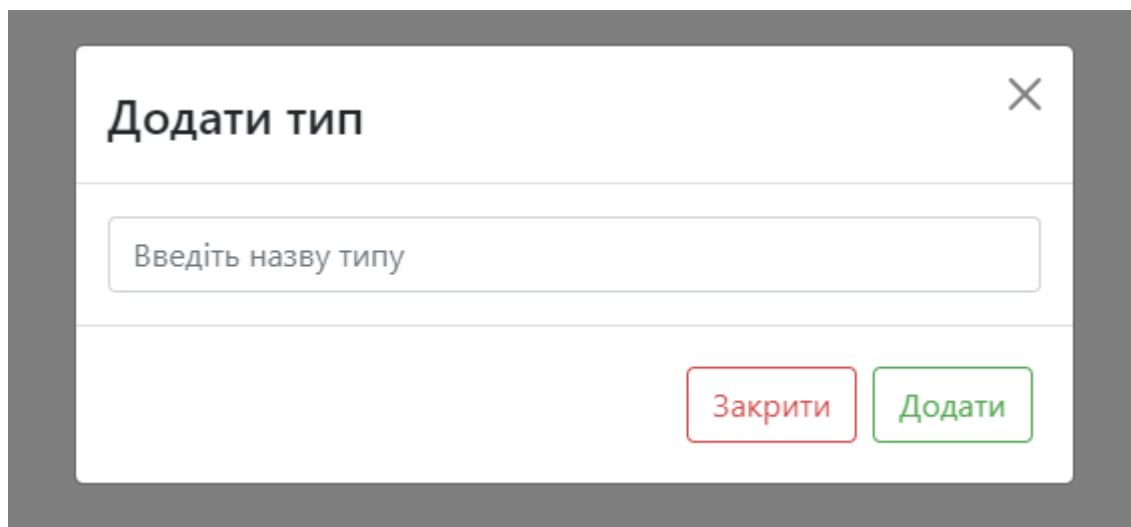


Рисунок 3.8. – Зображення модального вікна додавання типу



Рисунок 3.9. – Зображення модального вікна додавання жанру



Рисунок 3.10. – Зображення модального вікна додавання блогу

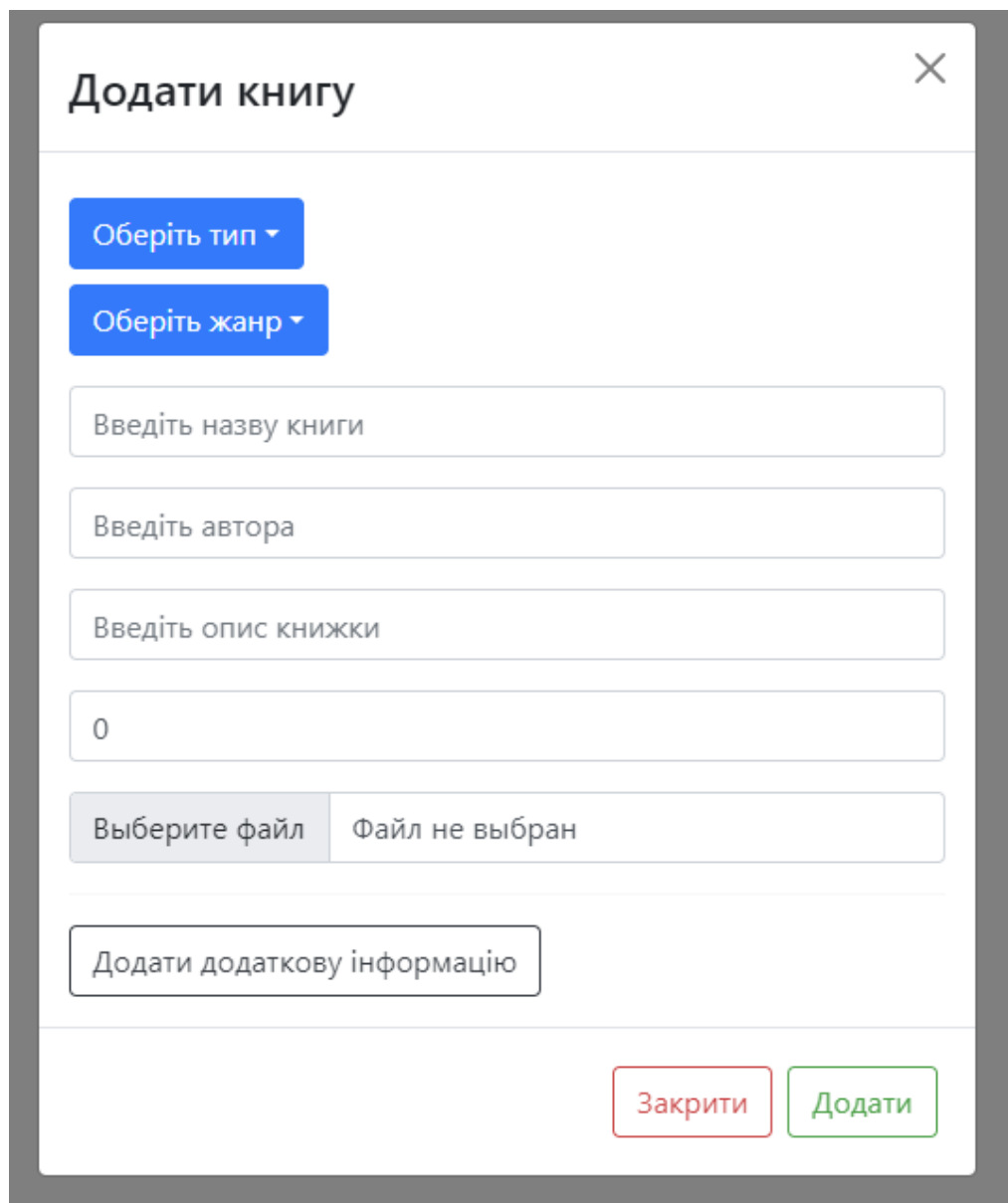


Рисунок 3.11. – Зображення модального вікна додавання книги

У модальному вікні можна побачити два випадаючих меню, в яких обирається тип та жанр рукопису, нижче йдуть поля, в які вводяться назва, автор, стислий опис книжки. Після цих полів слідує цифрове поле, в яке потрібно ввести ціну товару, який бажаєте додати. Присутня форма для завантаження картинки товару. Нижче форми розташована кнопка для додання додаткової інформації стосовно продукту, після натискання форми з'являються два поля (рисунок 3.12).

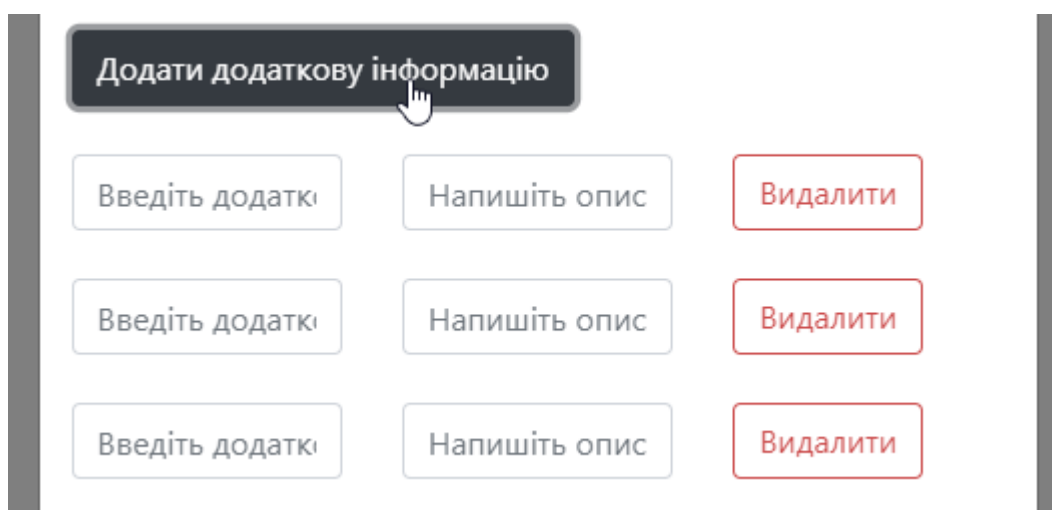


Рисунок 3.12. – Зображення модального вікна додавання книги

Перше відповідає за назву властивості, що її є надібність додати, інше має містити опис властивості.

3.5. Висновки за третім розділом

В практичній частині роботи було спроектовано та реалізовано веб-додаток "Інтернет-магазин з продажу книжок" із застосуванням ReactJS. Було представлено опис особливостей виконання основних функцій додатку та перевірено стабільність роботи сайту при виконанні регулярних задач які може зробити типовий користувач. Головна ідея створення додатку – стилістична стислість та зрозумілість і зручність користування. Незважаючи

на те, що деякі функції відсутні і потребують подальшого доопрацювання, в розробці реалізовано базові можливості:

- авторизація з реєстрацією користувача;
- можливість вибору фільтрування книг;
- додавання товарів та інших компонентів адміністратором;
- додавання товарів у кошик;
- перегляд каталогу представлених товарів.

ВИСНОВКИ

Огляд існуючих на сьогоднішній день книжкових інтернет-магазинів та порівняння їхніх можливостей дозволяє виділити основні функції, які обов'язково повинні бути наявними в проекті, та другорядні, які досить часто перевантажують додаток, тому від них бажано позбавитися. В якості прикладу реалізації зразкового співвідношення функціональних можливостей можна назвати сайт yakaoo.ua. Він має лаконічний дизайн та оптимальний функціонал з точки зору покупця.

В результаті аналізу типів сучасних веб-додатків було прийнято рішення про створення інтернет-магазину за типом кастомного додатку, архітектура якого буде виглядати і мати властивості MPA (Multi Page Application).

Для створення проекту вибрано наступний інструментарій:

- мова програмування – JavaScript;
- платформа серверної частини – Node.js;
- JavaScript-бібліотека для розробки інтерфейсів користувача ReactJS;
- СУБД – безкоштовна реляційна PostgreSQL;
- середовище розробки – Visual Studio Code.

Реалізація веб-додатка "Інтернет-магазин з продажу книжок" виконана із застосуванням ReactJS.

В додатку було втілено ідеологію розумного мінімалізму зі збереженням повного набору базових функцій магазину, зокрема, авторизації, каталогу товарів, фільтрації за споживацькими параметрами та кошиком.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Конструктори мобільних додатків чи кастомна технологія? [Електронний ресурс] / – Режим доступу: [www. URL: https://www.seonews.ru/analytics/konstruktory-mobilnykh-prilozheniy-ili-kastomnaya-razrabotka-chestnyy-razbor-ot-purrweb/#:~:text=Что%20такое%20кастомная%20разработка,нуля%20-%20простор%20для%20воображения%20шире.](http://www.seonews.ru/analytics/konstruktory-mobilnykh-prilozheniy-ili-kastomnaya-razrabotka-chestnyy-razbor-ot-purrweb/#:~:text=Что%20такое%20кастомная%20разработка,нуля%20-%20простор%20для%20воображения%20шире.) – 31.03.2023р.
2. Стаття «Розробка веб-додатків: Основні етапи» [Електронний ресурс] / – Режим доступу: [www. URL: https://www.purrweb.com/ru/blog/kak-sozdat-veb-prilozhenie/](http://www.purrweb.com/ru/blog/kak-sozdat-veb-prilozhenie/) – 31.03.2023р.
3. ІНТЕРНЕТ МАГАЗИНИ види, специфіка та переваги [Електронний ресурс] / – Режим доступу: [www. URL: https://mir-fin.ru/internet_magasin.html](http://www.mir-fin.ru/internet_magasin.html) – 31.03.2023р.
4. Стаття «Розробка архітектури інтернет-магазину» [Електронний ресурс] / – Режим доступу: [www. URL: https://studwood.net/1950154/informatika/razrabotka_arhitektury_internet_magazina/](http://www.studwood.net/1950154/informatika/razrabotka_arhitektury_internet_magazina/) – 31.03.2023р.
5. Інтернет-книгарня book24.ua [Електронний ресурс] / – Режим доступу: [www. URL: https://book24.ua/ua/https://book24.ua/ua/](http://www.book24.ua/ua/) – 31.03.2023р.
6. Сайт видавництва "Ранок" [Електронний ресурс] / – Режим доступу: [www. URL: https://www.ranok.com.ua/](http://www.ranok.com.ua/) – 31.03.2023
7. Інтернет-магазин Yakaboo [Електронний ресурс] / – Режим доступу: [www. URL: https://www.yakaboo.ua/ /](http://www.yakaboo.ua/) – 31.03.2023р.
8. Що таке JavaScript? [Електронний ресурс] / – Режим доступу: [www. URL: https://aws.amazon.com/ru/what-is/javascript/](http://www.aws.amazon.com/ru/what-is/javascript/) – 31.03.2023р.
9. Сайт посібник з програмування JavaScript.ru [Електронний ресурс] / – Режим доступу: [www. URL: https://learn.javascript.ru/](http://www.learn.javascript.ru/) – 03.05.2023р.

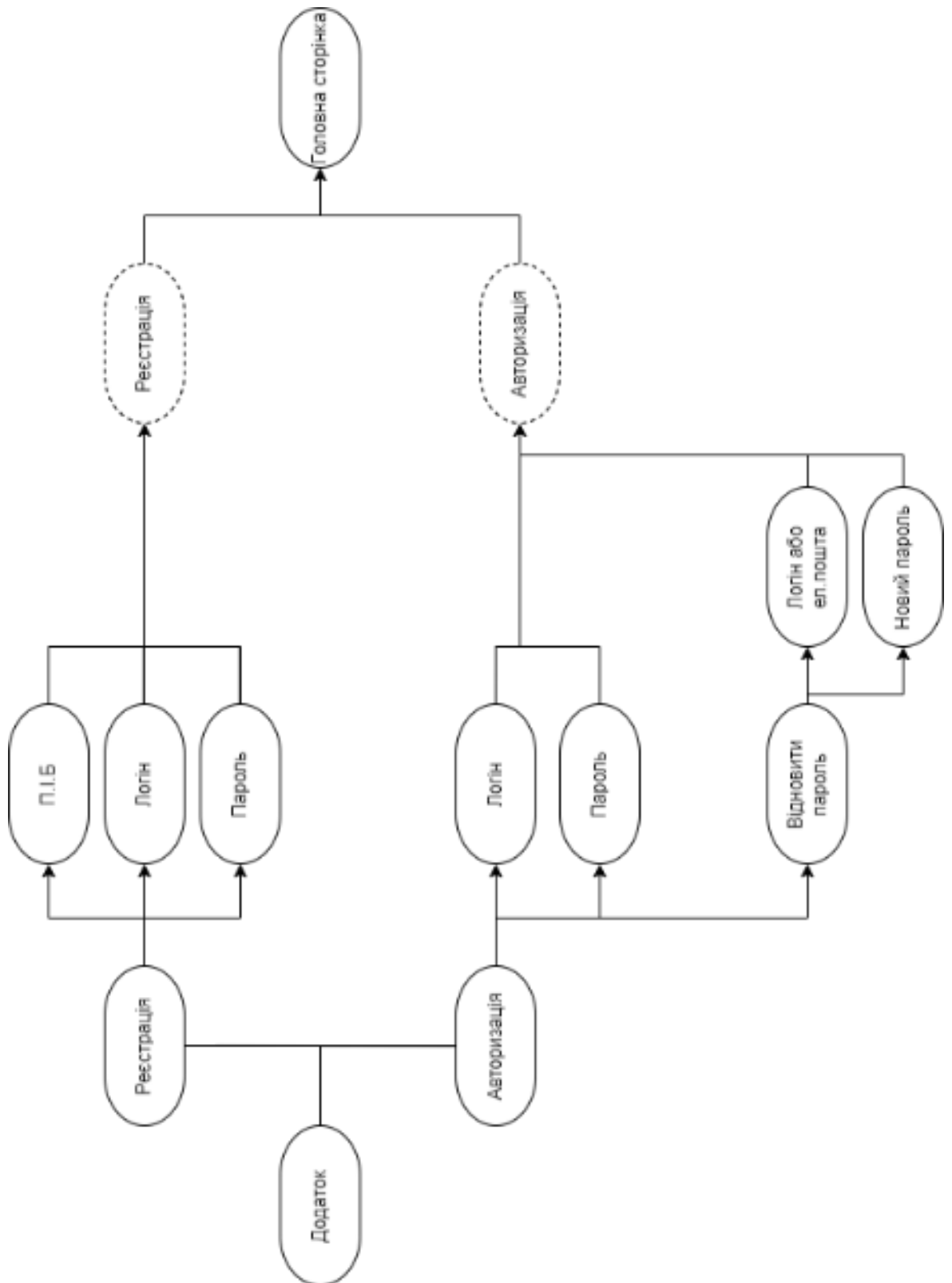
10. JavaScript [Електронний ресурс] / – Режим доступу: [www. URL: https://blog.skillfactory.ru/glossary/javascript/](http://www.skillfactory.ru/glossary/javascript/) – 03.05.2023р.
11. Node.js [Електронний ресурс] / – Режим доступу: [www. URL: https://blog.skillfactory.ru/glossary/node-js/](http://www.skillfactory.ru/glossary/node-js/) – 03.05.2023р.
12. Node.js – Wikipedia [Електронний ресурс] – Режим доступу до ресурсу: journals.uran.ua <https://en.wikipedia.org/wiki/Node.js> – 03.05.2023р.
13. Що це таке Node JS простими словами [Електронний ресурс] / – Режим доступу: [www. URL: \(journals.uran.ua\)](http://journals.uran.ua) <https://dan-it.com.ua/blog/chto-jeto-takoe-node-js-prostymi-slovami/> – 03.05.2023р.
14. Відмінності між Express.js та Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.back4app.com/ru/%D1%80%D0%B0%D0%B7%D0%BB%D0%B8%D1%87%D0%B8%D1%8F-%D0%BC%D0%B5%D0%B6%D0%B4%D1%83-express-js-%D0%B8-node-js/> – 03.05.2023р.
15. Що таке PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://www.dmosk.ru/terminus.php?object=postgresql> – 03.05.2023р.
16. Що таке PostgreSQL? [Електронний ресурс] – Режим доступу до ресурсу: <https://appmaster.io/ru/blog/chto-takoe-postgresql> – 03.05.2023р.
17. Мова Структурованих Запитів Postgres? [Електронний ресурс] – Режим доступу до ресурсу: <https://cqr.company/ua/wiki/protocols/postgres-structured-query-language/> – 03.05.2023р.
18. PostgreSQL проти MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://appmaster.io/ru/blog/postgresql-protiv-mysql> – 03.05.2023р.
19. Sequelize [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.skillfactory.ru/glossary/sequelize/> – 03.05.2023р.

- 20.Стаття «Ідеальний інструмент для роботи з СУБД без SQL для Node.js або Все, що ви хотіли знати про Sequelize.» [Електронний ресурс] / – Режим доступу: www. URL: <https://habr.com/ru/articles/565062/> – 03.05.2023р.
- 21.Стаття «Маршрутизація в React Router: як вона працює і чому її обирають розробники» [Електронний ресурс] / – Режим доступу: www. URL: <https://ru.hexlet.io/blog/posts/react-router-v6> – 05.05.2023р.
- 22.Розуміння React Router на прикладі простого блогу [Електронний ресурс] / – Режим доступу: www. URL: <https://webformyself.com/ponimanie-react-router-na-primere-prostogo-bloga/> – 05.05.2023р.
- 23.Стаття «Управління станом у React за допомогою MobX 2023» [Електронний ресурс] / – Режим доступу: www. URL: <https://vc.ru/u/1389654-machine-learning/597173-upravlenie-sostoyaniem-v-react-s-pomoshchyu-mobx-2023> – 05.05.2023р.
- 24.11 бібліотек (наборів компонентів) для React, про які варто знати у 2018-му [Електронний ресурс] / – Режим доступу: www. URL: <https://habr.com/ru/companies/ruvds/articles/346090/> – 23.05.2023р.
- 25.Посібник з програмування «React Bootstrap» [Електронний ресурс] / – Режим доступу: <https://www.javatpoint.com/react-bootstrap> – 03.05.2023р.
- 26.Visual Studio Code – Wikipedia [Електронний ресурс] / – Режим доступу: www. URL: https://uk.wikipedia.org/wiki/Visual_Studio_Code – 05.05.2023р.
- 27.Visual Studio Code [Електронний ресурс] / – Режим доступу: www. URL: <https://visualstudio.microsoft.com/ru/> – 05.05.2023р.

ДОДАТКИ

Додаток А

Схема взаємодії блоків інтерфейсу та взаємозв'язку між ними



Додаток Б

Діаграма зв'язків таблиць у базі даних

