МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки та програмної інженерії
Кафедра інженерії програмного забезпечення

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

Катерина НЕСТЕРЕНКО
“____”_____2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
Бакалавр

Тема: “ Інтернет магазин наукової літератури”

Виконавець: Михно Дмитро Петрович

Керівник: Гололобов Дмитро Олександрович

Нормоконтролер: Варнавський Вячеслав Володимирович

Київ 2024

‘

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки та програмної інженерії
Кафедра інженерії програмного забезпечення
Освітній ступінь Бакалавр
Спеціальність 121 Інженерія програмного забезпечення
Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Катерина НЕСТЕРЕНКО

"___" _____ 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи студента
Михна Дмитра Петровича

1. Тема кваліфікаційної роботи: «Інтернет магазин наукової літератури». затверджена наказом ректора від 08.12.2024 р. № 2483/ст

2. Термін виконання проекту: з 03.01.2024 р. до 29.02.2024 р.

3. Вихідні дані до роботи : Програмний продукт , інтернет-магазин наукової літератури, який є комплексним веб-сервісом для зручного пошуку, вибору та придбання наукових видань в електронному та друкованому форматах.

4. Зміст пояснювальної записки:
   1. Огляд предметній області.
   2. Методи та засоби реалізації програмного продукту.
   3. Реалізація програмного продукту.

5. Перелік обов'язкових слайдів презентації:
   1. Огляд предметної області
   2. Процеси організації продажу наукової літератури
   3. Автоматизація інтернет-магазину
   4. Технології та архітектура
   5. Функціональні і не функціональні вимоги

‘

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1.	Розробка та затвердження графіка роботи.	08.12.2023-28.12.2023	
2.	Підготовка та написання 1 розділу «огляд предметній області».	10.01.2024-17.01.2024	
3.	Підготовка та написання 2 розділу «методи та засоби реалізації програмного продукту». Відсилка керівнику.	18.01.2024-22.01.2024	
4.	Підготовка та написання 3 розділу «реалізація програмного продукту». Відсилка керівнику.	23.01.2024-31.01.2024	
5.	Загальне редагування та друк пояснювальної записки, графічного матеріалу	27.01.2024-31.01.2024	
6.	Проходження нормо-контролю, перевірка на антиплагіат, перепліт пояснювальної записки.	01.02.2024-09.02.2024	
7.	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації	01.02.2024-05.02.2024	
8.	Отримання відгуку керівника, рецензії.	10.02.2024 - 22.02.2024	
9.	Підготовка матеріалів для передачі секретарю ДЕК (ПЗ, CD-R з електронними копіями ПЗ, презентації, відгук керівника, рецензія) в папці	23.02.2024-29.02.2024	

Дата видачі завдання 08.12.2024 р.
Керівник дипломної роботи: _____ Дмитро ГОЛОЛОБОВ
Завдання прийняв до виконання: Дмитро МИХНО

‘

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Штучний інтелект для управління модулем пошуку, спостереження та наведення на об’єкт»: 51 сторінок, 20 рисунків, 0 таблиці, 12 використаних джерел.

НТЕРНЕТ-МАГАЗИН НАУКОВОЇ ЛІТЕРАТУРИ, АВТОМАТИЗАЦІЯ ПРОЦЕСІВ, ВЕБ-СЕРВІС, REACT, ASP.NET CORE , POSTGRESQL, HTML, CSS, JAVASCRIPT,

Об’єкт розробки – інтернет-магазин для продажу та розповсюдження наукової літератури, що інтегрує передові технології

Мета роботи – Розробка програмного продукту, який дозволить оптимізувати процеси пошуку, відбору та придбання наукової літератури, забезпечуючи високий рівень задоволеності користувачів та ефективність адміністрування контенту.

ABSTRACT

Explanatory note to the qualification work "Artificial intelligence for controlling the module of search, observation and targeting of the object": 51 pages, 20 figures, 0 tables, 12 references.

ONLINE STORE OF SCIENTIFIC LITERATURE, PROCESS AUTOMATION, WEB SERVICE, REACT, ASP.NET CORE, POSTGRESQL, HTML, CSS, JAVASCRIPT,

Object of development - The object of development is an online store for the sale and distribution of scientific literature that integrates advanced technologies.

Purpose of work - Development of a software product that will optimize the processes of searching, selecting and purchasing scientific literature, ensuring a high level of user satisfaction and efficiency of content administration.

‘

ЗМІСТ

‘

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

HTML - HyperText Markup Language

CSS - Cascading Style Sheets

СУБД - Система управления базами даних

DOM - Об'єктна модель документа

API - application programming interface

MVC - Модель вид контролер

SPA - Single Page Application

‘

ВСТУП

В контексті сучасного цифрового світу, роль інтернет-магазинів наукової літератури стає надзвичайно актуальною. Вони відіграють ключову роль у спрощенні доступу до наукових ресурсів, забезпечуючи важливий зв'язок між науковою спільнотою та широкою публікою. Ця тема набуває особливої важливості в епоху, коли інформаційні технології пронизують усі сфери людської діяльності, трансформуючи традиційні підходи до навчання, дослідження та поширення знань.

Цифровізація наукового контенту відкриває нові можливості для розповсюдження та використання наукових знань. Інтернет-магазини, які зосереджуються на продажу наукової літератури, стають важливими інструментами в цьому процесі. Вони не тільки надають легкий доступ до широкого спектру наукових матеріалів, але й сприяють більшій взаємодії в науковій спільноті, забезпечуючи платформу для обміну ідеями та дослідженнями.

Розвиток цих магазинів також відіграє важливу роль у підвищенні глобальної освітньої доступності та забезпеченні рівних можливостей для навчання. Це особливо важливо в контексті забезпечення доступу до останніх наукових досліджень та розробок у всьому світі. Такі платформи допомагають згладити розрив у доступі до наукових ресурсів, зокрема в регіонах, де традиційні академічні ресурси можуть бути обмежені.

З огляду на це, розробка інтернет-магазину наукової літератури є не лише відповіддю на сучасні технологічні вимоги, але й важливим кроком у напрямку розширення можливостей для наукової освіти та досліджень. Це стає мостом, що з'єднує виробників наукового контенту з його споживачами, відкриваючи нові шляхи для обміну знаннями та сприяння науковому прогресу.

‘

РОЗДІЛ 1.
ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз процесів організації продажу наукової літератури

Організація продажу наукової літератури відіграє важливу роль у формуванні інформаційного простору сучасного освітнього середовища, сприяючи науковому розвитку та поширенню знань. Ефективність цих процесів безпосередньо впливає на доступність та поширення академічних ресурсів, визначаючи можливості для глибокого дослідження та навчання.

Організацією продажу займаються магазини-книгарні наукової літератури. Книгарня наукової літератури - це спеціалізований магазин, який спеціалізується на продажі високоспеціалізованих книг, журналів, наукових робіт і матеріалів, призначених для академічних досліджень, вивчення наукових тем та поглибленого розуміння різних наукових дисциплін. Такі книгарні зазвичай мають багатий асортимент та можуть надавати доступ до актуальних наукових відкриттів та досліджень у різних галузях науки.

Для організації продажу наукової літератури магазин повинен організувати наступні заходи:

- *Закупівля та управління запасами:*
  - Визначення Потреб : Аналіз ринку та попиту, для визначення, які книги та в якій кількості потрібно закупити. Враховуючи тренди, сезонність та інші фактори, що впливають на попит.
  - Вибір Постачальників: Пошук надійних постачальників, які пропонують якісні книги за конкурентними цінами.

‘

- Створення договорів з постачальниками: укладення договорів із чіткими умовами про ціни, терміни доставки, умови повернення товарів та гарантії.
- Система відстеження запасів: Розробка системи для відстеження наявності книг та реагування на зміни у попиті.
- Аналіз запасів: Постійний аналіз запасів для ідентифікації надлишкових або нестач запасів та визначення оптимальних рівнів запасів на основі історії продажів.
- Організація логістики та доставки: Створення ефективної системи логістики для забезпечення своєчасної доставки книг до магазину або складу.

- *Класифікація та розміщення книг:*
  - Систематизація за Жанрами та Темами: Книги слід упорядковувати за жанрами, темами, або навіть науковими дисциплінами, щоб покупці могли легко знайти потрібну літературу. Наприклад, окремі розділи для історії, фізики, літера
  - Розміщення за Авторами: Ви можете також упорядковувати книги за авторами, особливо якщо мова йде про відомих письменників або вчених. Це полегшує пошук для тих, хто шукає конкретного автора.турознавства тощо.
  - Легкий пошук та навігація: У онлайн-магазинах необхідно забезпечити зручний пошук із допомогою фільтрів та категорій. В офлайн-магазинах - легка навігація між розділами.
- *Ціноутворення:*
  - Аналіз Витрат і Конкурентоспроможності: Визначення вартості закупівлі книг і ціноутворення конкурентів.
  - Аналіз цільової аудиторії: Аналіз фінансові можливості цільової аудиторії платити за певні види літератури.
  - Використання цінових стратегій: Використання різних цінових стратегій, як MSRP, кількісне ціноутворення, ціноутворення вище/нижче за конкуренцію, а також стратегії приваблення покупців, такі як нижчі ціни на окремі товари для привернення уваги до інших товарів з вищою маржею.

‘

- Промоції: Тимчасові знижки та акцій, які можуть сприяти збільшенню продажів, особливо під час сезонних піків або заходів.

- *Презентація продукту:*
  - Використання приваблевого преставлення товару: Створення привабливого представлення книг, організації товарів в каталогі та інших методів для збільшення привабливості та видимості товарів
  - Інтерактивні Елементи: Включення інтерактивних елементів, таких як читацькі куточки, мультимедійні стенди або презентаційні екрани.
- *Продаж та обслуговування клієнтів:*
  - Консультації Клієнтів: Надавання клієнтам корисних порад та рекомендацій.
  - Допомога у Виборі Літератури: Виявлення потреби та інтереси клієнтів.
  - Обробка Покупок: Управління оплатами, видачу чеків та упаковку книг.
  - Видача Товарів: Організація надійної та своєчасної пердачі товару.
- *Маркетинг та реклама:*
  - Рекламні Кампанії: Проведення цільових рекламних кампаній у цифрових та традиційних медіа для привернення уваги до магазину та його продуктів.
  - SEO та Контент-Маркетинг: Оптимізація вебсайту магазину для пошукових систем, створення якісного контенту, який приваблює відвідувачів та підвищує рейтинги у пошукових системах.
  - Партнерство та Співпраця: Співпраця з видавництвами, авторами, освітніми закладами та іншими організаціями для проведення спільних маркетингових акцій.
- *Фінансовий облік та аналітика:*
  - Ведення Обліку Продажів, Витрат та Прибутків
  - Автоматизація Обліку: Впровадження автоматизованих систем обліку
  - Аналіз Ефективності Продажів: аналіз різних факторів, такі як ціноутворення, рекламні кампанії та зміни на ринку, впливають на продажі.
- *Повернення та обмін:*
  - Чітке визначення умов повернення та обміну: Створення чітких умов повернення та обміну товарів.

‘

- Проведення легкодоступної комунікації про політику повернення
- Ефективна обробка повернень та обмінів: Швидка організація процесу повернення з чіткими інструкціями та підтримкою.
- Моніторинг та аналіз повернень: аналіз показників повернень та обмінів, щоб виявити можливі проблеми з якістю товарів або рівнями задоволеності клієнтів.

- *Зв'язок з клієнтами та збір відгуків:*
  - Систематичний збір відгуків: Впровадження структурованих механізмів для збору відгуків, включаючи опитування після покупки, цифрові форми відгуку на вебсайті та аналіз соціальних мереж.
  - Реагування на відгуки та запити: Організація надавання відповіді на відгуки та запити клієнтів у встановлені терміни, демонструючи увагу до потреб клієнтів та готовність до діалогу.
  - Інтеграція відгуків у рішення щодо поліпшень: Використання зібраних відгуків як бази даних для прийняття рішень щодо покращення продуктів, сервісів та операційних процесів.
  - Прозорість та відкритість у комунікації: Підтримання прозорості у взаємодії з клієнтами, публічне визнання проблем та інформування про вжиті заходи.

Враховуючи вищезазначені аспекти, можна стверджувати, що успішна організація продажу наукової літератури вимагає комплексного та уважного підходу до кожного етапу бізнес-процесу. Від ефективної закупівлі та управління запасами до вдосконалення системи класифікації та розміщення книг, від ретельно продуманого ціноутворення до креативної презентації продуктів, кожен елемент має важливе значення. Обслуговування клієнтів та продажі, маркетинг та рекламні стратегії, а також точний фінансовий облік та аналітика — всі ці складові формують основу для ефективного функціонування та зростання магазину наукової літератури.

1.2.Автоматизація інтернет магазину

В умовах сучасного цифрового світу автоматизація інтернет-магазину наукової літератури стає ключовим елементом покращення продажу та організації

‘
роботи. Цифрові технології та автоматизовані системи можуть значно оптимізувати процеси, які були описані вище, та забезпечити більш ефективне та гнучке управління ресурсами.

Автоматизація процесу закупівлі та управління запасами:

Інтеграція автоматизованих систем управління запасами дозволяє точніше аналізувати попит, відстежувати рівні запасів у реальному часі, а також автоматично замовляти нові книги, коли рівні запасів опускаються до певної позначки. Це зменшує ризики втрати продажів через відсутність товару та знижує витрати на зберігання надлишкових запасів.

Ефективна класифікація та розміщення книг:

Використання розширених алгоритмів класифікації та пошукових систем дозволяє покупцям швидко знаходити потрібні книги. Це може включати функції фільтрації за авторами, жанрами, темами чи науковими дисциплінами. Штучний інтелект може також рекомендувати книги на основі попередніх покупок або переглядів користувачів.

Оптимізація процесу ціноутворення:

Автоматизація дозволяє здійснювати динамічне ціноутворення на основі зміни попиту, конкурентоспроможності та витрат. Системи можуть аналізувати великі обсяги даних для визначення оптимальних цін, що збільшує прибутковість та конкурентоспроможність.

Вдосконалення презентації продукту:

Інтернет-магазин дозволяє використовувати різноманітні засоби візуалізації, такі як високоякісні зображення обкладинок, віртуальні тури по книгарні, відеоогляди та фрагменти книг. Це покращує враження від шопінгу та сприяє кращому розумінню продукту.

Поліпшення обслуговування клієнтів:

Чат-боти та автоматизовані служби підтримки можуть надавати відповіді на запитання клієнтів, допомагати з вибором літератури та управляти процесом покупки. Це підвищує задоволеність клієнтів та знижує навантаження на персонал.

Інтеграція маркетингу та реклами:

‘

Інтернет-магазин дозволяє інтегрувати різні канали маркетингу та реклами, включаючи електронну пошту, соціальні мережі, SEO та контент-маркетинг. Автоматизовані інструменти дозволяють точно цілитися на цільову аудиторію та аналізувати ефективність кампаній.

Фінансовий облік та аналітика:

Автоматизовані системи фінансового обліку та аналітики дозволяють точно вести облік продажів, витрат та прибутків, а також здійснювати глибокий аналіз різних аспектів бізнесу для виявлення можливостей для розвитку.

Покращення системи повернень та обмінів:

Автоматизація процесу повернення та обміну забезпечує більш просту та зручну систему для клієнтів, знижуючи час обробки та покращуючи загальне враження від обслуговування.

Зв'язок з клієнтами та збір відгуків:

Автоматизація дозволяє систематично збирати та аналізувати відгуки клієнтів, використовуючи дані для вдосконалення продуктів та послуг.

1.3.Аналіз аналогів інтернет магазину в предметній області

Після детального аналізу процесів організації продажу та можливості автоматизації продажу наукової літератури, наступним кроком є дослідження того, як ці процеси реалізовані на існуючих вебсайтах, які спеціалізуються на продажу наукової літератури. Це дозволить зрозуміти, які рішення та функціонал є ефективними і можуть бути адаптовані або покращені для створення власного інтернет-магазину наукової літератури.

Провівши ґрунтовний аналіз інтернет-магазинів наукової літератури, я зосередив увагу на двох провідних міжнародних платформах: Wiley Online Library та Springer. Ці ресурси представляють собою визначні приклади успішної інтеграції широкого спектру наукових публікацій з різних дисциплін, демонструючи високий рівень доступності та функціональності для користувачів по всьому світу.

Важливим аспектом цього аналізу було також дослідження локального ринку України, де особливо виділяється інтернет-магазин "Університецька книга". Цей

‘
ресурс ілюструє адаптацію міжнародних стандартів до місцевих потреб, забезпечуючи доступ до великої колекції академічних та наукових робіт, важливих для української освітньої та наукової спільноти.

Wiley Online Library представляє собою комплексний цифровий ресурс, який забезпечує доступ до широкого спектру академічних матеріалів, у тому числі книг та наукових статей. Платформа пропонує як окремі покупки книг, так і підписки, що дозволяють користувачам отримувати доступ до обширної бібліотеки літератури. Цікавою особливістю є існування публікацій з відкритим доступом, а також можливість колаборації з університетами, що забезпечує безкоштовний доступ для студентів.

Для кожної книги на сайті надається детальна інформація, включаючи обкладинку, автора, ISBN номери (друкований та онлайновий), дату публікації, короткий опис, зміст, опис кожного розділу та список використаних джерел. Статті також містять інформацію про автора, редактора, дату публікації, короткий опис та використану літературу.

Пошук на платформі є гнучким та розгалуженим. Користувачі можуть здійснювати пошук за предметами, напрямками, темами, а також використовувати фільтри для відбору книг і статей за типом публікації, датою, місцем публікації та автором. Каталог можна сортувати за датою, актуальністю або назвою.

Wiley Online Library також пропонує можливість отримання сповіщень про новини в наукових дослідженнях, анонси подій та аналізувати потреби користувачів. Після покупки, до книг забезпечується онлайн доступ та можливість скачування у форматі PDF.

Однак, інтерфейс сайту може здатися складним, оскільки він вимагає різних рівнів реєстрації для доступу до різних функцій, включаючи бібліотеку, покупки та перегляд блогу. Сайт також має функціонал для різних типів користувачів - авторів, звичайних користувачів та адміністраторів, дозволяючи кожній групі виконувати свої специфічні завдання.

Доповнюється великою кількістю частих запитань (FAQ) та можливістю зв'язку зі службою підтримки. Сайт також забезпечує аналітику продажів та використання

‘
книг і статей, є адаптованим для мобільних користувачів та постійно оновлюється, аби відповідати змінам у науковому світі. Особливо важливою є інтеграція сайту з сотнями академічних баз даних, бібліотечних систем та платформ, що розширює його функціональність та доступність.

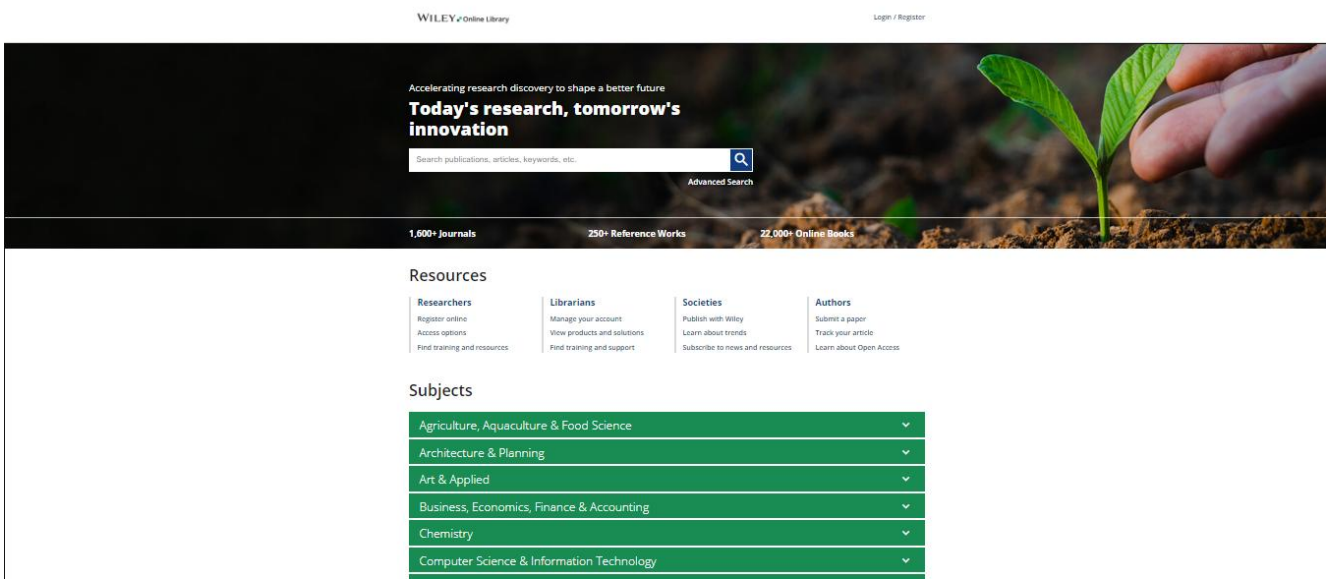

Рис. 1.1. Wiley Online Library

Springer, визнана у всьому світі як видавнича компанія, що спеціалізується на академічних ресурсах, пропонує широкий спектр матеріалів, включаючи електронні та фізичні книги, наукові статті, доповіді конференцій, довідкові роботи, матеріали конференцій, журнали та відеоматеріали. Особливістю Springer є не лише її різноманітний каталог, але й можливість купівлі окремих розділів книг, а також публікацій з відкритим доступом. Видавництво також активно співпрацює з університетами по всьому світу, забезпечуючи доступ до наукових матеріалів.

Кожна публікація на платформі Springer містить детальну інформацію, таку як фото обкладинки, дані про автора та редактора, дату публікації, короткий опис книги, зміст, описи кожного розділу та список використаної літератури. Це забезпечує користувачам глибоке розуміння матеріалу перед придбанням.

Функціонал пошуку на сайті дозволяє виконувати запити за предметами, при цьому можна фільтрувати результати за типом публікації, датою публікації, мовою,

‘
дисципліною та піддисципліною. Каталог можна сортувати за датою або актуальністю, що забезпечує зручність у навігації та пошуку актуальних досліджень.

Springer надає своїм користувачам можливість отримувати сповіщення про нові наукові дослідження та події, а також аналізує потреби користувачів для покращення своїх послуг. Після покупки електронної книги користувачам доступний онлайн доступ та можливість скачування у форматі PDF. Фізичні книги також доступні для замовлення з доставкою по всьому світу.

Інтерфейс сайту Springer є інтуїтивно зрозумілим та легким у використанні, що сприяє комфортному користувацькому досвіду. Крім того, сайт підтримує різні міжнародні мови, забезпечуючи доступність для глобальної аудиторії. Наявність розширеного розділу частих запитань (FAQ) та служби підтримки допомагає вирішувати запитання та проблеми користувачів.

Сайт Springer адаптований для мобільних користувачів, що робить доступ до наукових матеріалів зручним з будь-якого пристрою. Крім того, сайт постійно оновлюється, відображаючи найсвіжіші тенденції та зміни у науковому світі. Важливою особливістю є інтеграція Springer з сотнями академічних баз даних, бібліотечних систем та платформ, що забезпечує широкий доступ до наукових ресурсів.

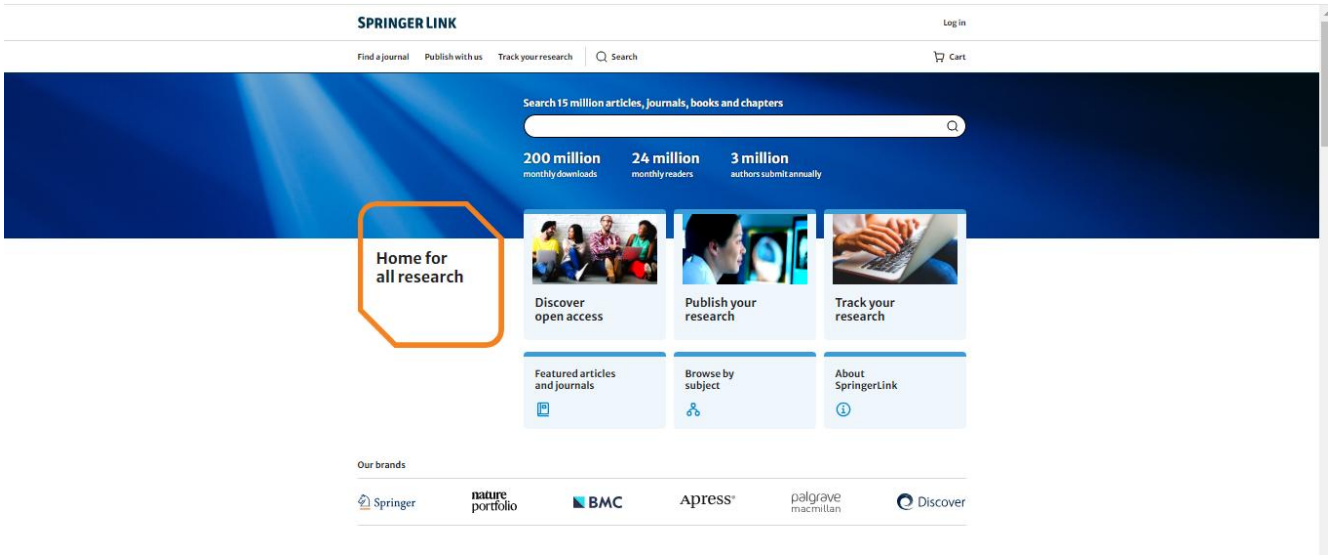

Рис. 1.2. Springer

Вебсайт "Університетська книга" представляє собою інноваційний цифровий

майданчик, спеціалізований на розповсюдженні фізичних книжкових видань, основною аудиторією якого є академічне співтовариство вищих навчальних закладів України. Характеристичною рисою цього ресурсу є його здатність надавати глибокий аналітичний огляд кожного видання, що включає в себе детальну візуалізацію обкладинки, обсяжні дані про авторство, ISBN, дату публікації, а також анотацію, змістовий опис та можливість ознайомлення з вибраними фрагментами тексту у форматі PDF. Це забезпечує користувачам унікальну можливість до глибшого розуміння матеріалу перед придбанням.

Система пошуку на сайті демонструє високий рівень адаптивності, оскільки дозволяє ефективно фільтрувати літературу за критеріями, що охоплюють дисципліну, тип видання, цінову категорію та формат. Така структурованість пошукового процесу не лише сприяє оптимізації вибору публікацій, але й відіграє ключову роль у задоволенні специфічних освітніх та наукових запитів користувачів.

Механізм електронної розсилки, що імплементовано на сайті, виступає в якості інструменту для забезпечення постійного зв'язку із користувачами, пропонуючи їм оновлену інформацію щодо новинок та акційних пропозицій. Інтерфейс сайту, характеризуючись своєю простотою та інтуїтивністю, сприяє легкій орієнтації та доступності ресурсу.

Проте, слід зазначити, що відсутність розділу з детальною документацією або FAQ секції може викликати певні складнощі у користувачів, які шукають вичерпну інформацію або специфічні роз'яснення. Респонсивний дизайн вебсайту, адаптований для мобільних пристроїв, компенсує цей недолік, надаючи користувачам гнучкість у використанні різноманітних гаджетів для доступу до контенту.

Подальшу цінність цьому цифровому ресурсу надає наявність служби підтримки та інтерактивного функціоналу, який дозволяє залишати коментарі та оцінки, тим самим формуючи активну спільноту читачів, що сприяє обміну думками та досвідом. Незважаючи на це, обмеженість у сфері інтеграції з партнерськими організаціями може впливати на розширення спектру послуг та функціональності цього вебсайту.

‘

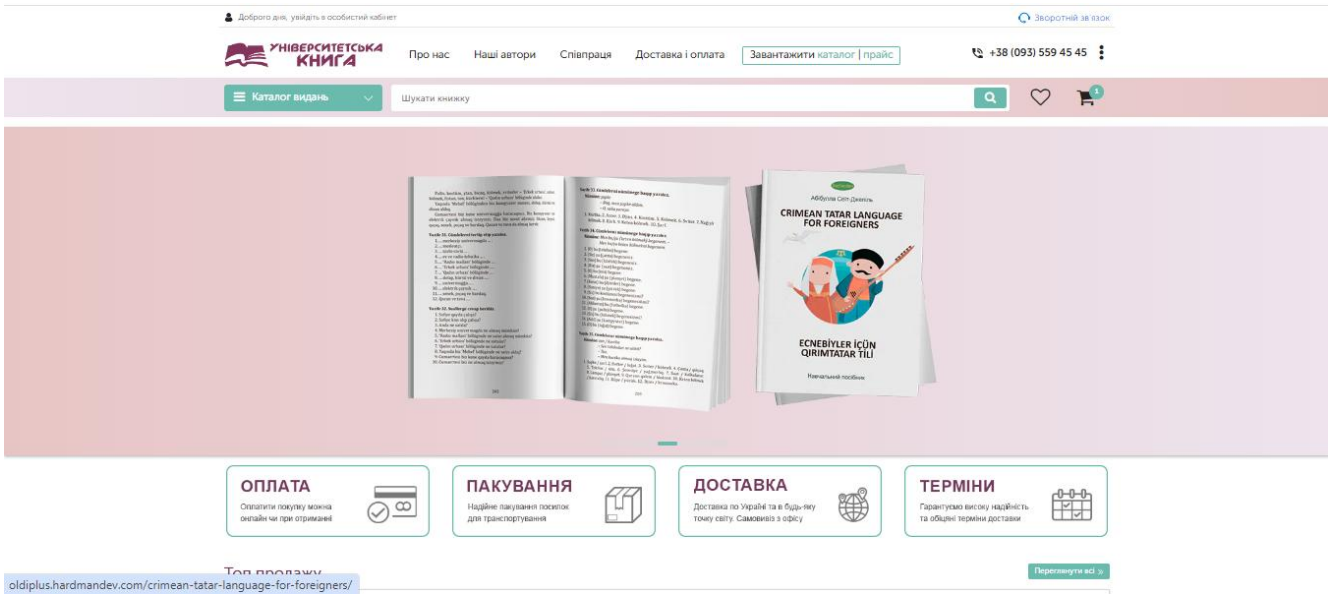

Рис. 1.3. Університетська книга

Отже, в контексті аналізу існуючих програмних засобів, виявлено, що успіх веб-ресурсів для продажу наукової літератури значною мірою залежить від їх здатності інтегрувати широкий спектр функціоналу та вмісту. На прикладах Wiley Online Library та Springer було видно, як ефективне управління великими обсягами академічних матеріалів та наукових публікацій, разом із розширеними можливостями пошуку та персоналізації, може забезпечити високу користувацьку цінність.

Також було встановлено, що адаптація до специфічних ринкових умов та потреб користувачів, як показано на прикладі "Університетської книги", є ключовим фактором для забезпечення успіху на локальних ринках. Це підтверджує важливість гнучкості та відповідності контенту, функціоналу та дизайну вебсайтів до конкретних потреб та культурних особливостей різних аудиторій.

Зрештою, цей аналіз надає цінні вказівки для розробки та оптимізації програмних засобів власного інтернет-магазину, підкреслюючи необхідність інноваційного підходу, зорієнтованого на користувача, а також акцентуючи на важливості неперервного розвитку та оновлення.

Висновок до 1 розділу

‘

Аналізуючи процеси організації продажу наукової літератури та потенціал автоматизації інтернет-магазину, можна дійти важливих висновків. Розглянувши ключові аспекти та виклики, які виникають у традиційному управлінні продажами наукової літератури, стає зрозумілим, що впровадження автоматизованих систем значно оптимізує бізнес-процеси, підвищуючи ефективність роботи та забезпечуючи кращу якість обслуговування клієнтів.

Автоматизація інтернет-магазину сприяє підвищенню доступності наукових матеріалів, що є критично важливим для сучасного освітнього та наукового середовища. Це дозволяє забезпечити ширший доступ до академічних ресурсів, спрощує пошук та вибір літератури, і в цілому, забезпечує більш комфортне та ефективне споживання наукових матеріалів.

‘

РОЗДІЛ 2.
МЕТОДИ ТА ЗАСОБИ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ

2.1. Архітектура програмного забезпечення

У процесі розробки програмного продукту, зокрема, інтернет-магазину наукової літератури, використання архітектурного шаблону Model-View-Controller (MVC) відіграє ключову роль. Архітектура MVC розділяє програму на три взаємопов’язані компоненти: модель, вид та контролер, забезпечуючи таким чином чітке розмежування відповідальностей. Цей підхід сприяє створенню структурованого, ефективно організованого коду, що є особливо важливим для розробки великих та складних систем, таких як веб-магазини.

Однією з основних переваг MVC є поліпшення масштабованості та гнучкості програми. Архітектура дозволяє розробникам модифікувати або розширювати окремі компоненти системи незалежно один від одного, що знижує ризик виникнення помилок та спрощує процес оновлення програми. Завдяки розділенню відповідальностей, команда розробників може ефективно працювати над різними аспектами проекту, не порушуючи функціональності інших частин.

Модель у MVC відповідає за бізнес-логіку та обробку даних. Вона взаємодіє з базою даних, забезпечуючи обробку запитів, їх зберігання та вилучення. У контексті інтернет-магазину модель управляє даними про продукти, замовлення, користувачів та інші важливі бізнес-процеси.

Вид у MVC представляє інтерфейс користувача, через який відбувається взаємодія з програмою.

‘

Він відповідає за візуалізацію даних, отриманих від моделі, у зручному та зрозумілому форматі. Це включає веб-сторінки з каталогом продуктів, формами для замовлення, інтерфейсами управління користувацькими рахунками та іншими важливими елементами інтерфейсу користувача.

Контролер виступає як посередник між моделлю та видом, обробляючи вхідні дані від користувача, використовуючи модель для отримання чи зміни даних та вибираючи відповідний вид для відображення цих даних. Контролер грає ключову роль у управлінні логікою користувальницького інтерфейсу та бізнес-процесами.

Однак, використання MVC також ставить перед розробниками певні виклики. Основною складністю є управління залежностями між трьома компонентами, особливо у великих та складних системах, де кількість видів та контролерів може бути значною. Також, залежність від фреймворків, які імплементують архітектуру MVC, може обмежити гнучкість розробки. Однак, переваги, що надає MVC, зокрема у термінах чіткої організації коду, легкості у тестуванні та підтримці, а також ефективності співпраці в команді, роблять цей архітектурний шаблон вибором номер один для багатьох сучасних веб-додатків.

У сумі, використання MVC у розробці інтернет-магазину наукової літератури забезпечує не тільки ефективну структуру програми, але й створює умови для її масштабування та адаптації до змінюваних вимог бізнесу. Це робить MVC незамінним інструментом у сучасній веб-розробці.

‘

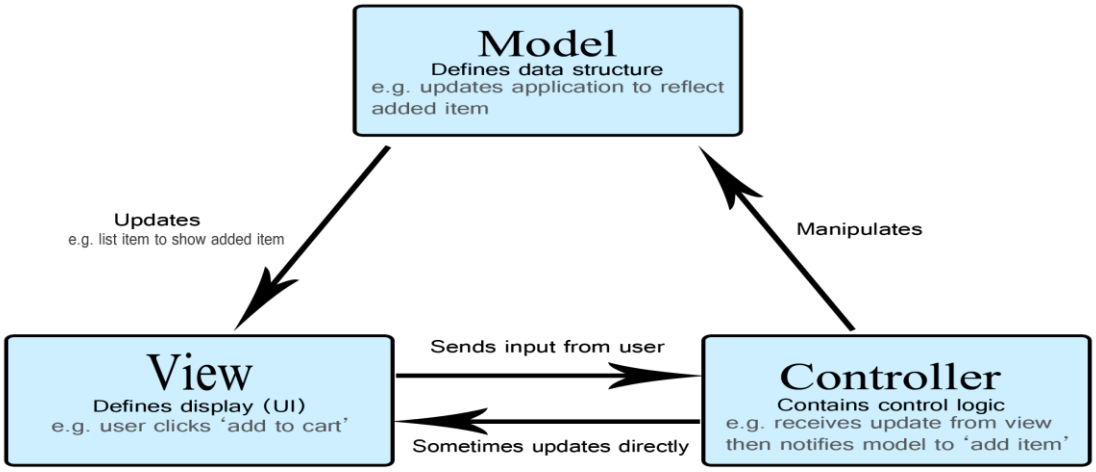

Рис. 2.1. Модель MVC

2.2.Html

HTML, або HyperText Markup Language, становить основу для створення веб-сторінок і веб-додатків, використовуючи систему тегів для організації та представлення контенту. Від заголовків, які описуються тегами <h1>, <h2>, до абзаців через <p> і гіперпосилань через <a>, HTML формує каркас веб-сторінки, на якому розміщуються текст, зображення, медіа-файли та інші елементи. Ці теги не лише структурують вміст, але й надають йому семантичне значення, роблячи контент зрозумілим як для користувачів, так і для браузерів.

Структура документа HTML відповідає ієрархічній моделі, де кожен елемент, від кореневого тега <html> до внутрішніх елементів, як <head> та <body>, виконує певну роль. Секція <head> містить мета-інформацію про документ, тоді як <body> є контейнером для основного вмісту сторінки.

Значення HTML розкривається ще більше завдяки взаємодії з Document Object Model (DOM), програмним інтерфейсом для HTML-документів. DOM перетворює всі елементи HTML-документа в об'єкти, створюючи тим самим "дерево" об'єктів. Кожен тег у HTML-документі відповідає вузлу в DOM, що дозволяє програмам динамічно змінювати структуру, стиль та вміст сторінки.

‘

Ця взаємодія між HTML та DOM є фундаментальною для динамічних веб-додатків. Вона дозволяє реалізувати інтерактивність на сторінках, забезпечуючи можливість зміни контенту без перезавантаження сторінки. Наприклад, коли користувач додає товари до кошика в інтернет-магазині, DOM може оновити відображення кошика в реальному часі, використовуючи інформацію з HTML-структури сторінки.

Таким чином, HTML і DOM разом формують основу для створення та взаємодії з веб-сторінками. Вони забезпечують структуру та засоби для динамічної та багатофункціональної веб-розробки, яка є необхідною для сучасних інтернет-магазинів.

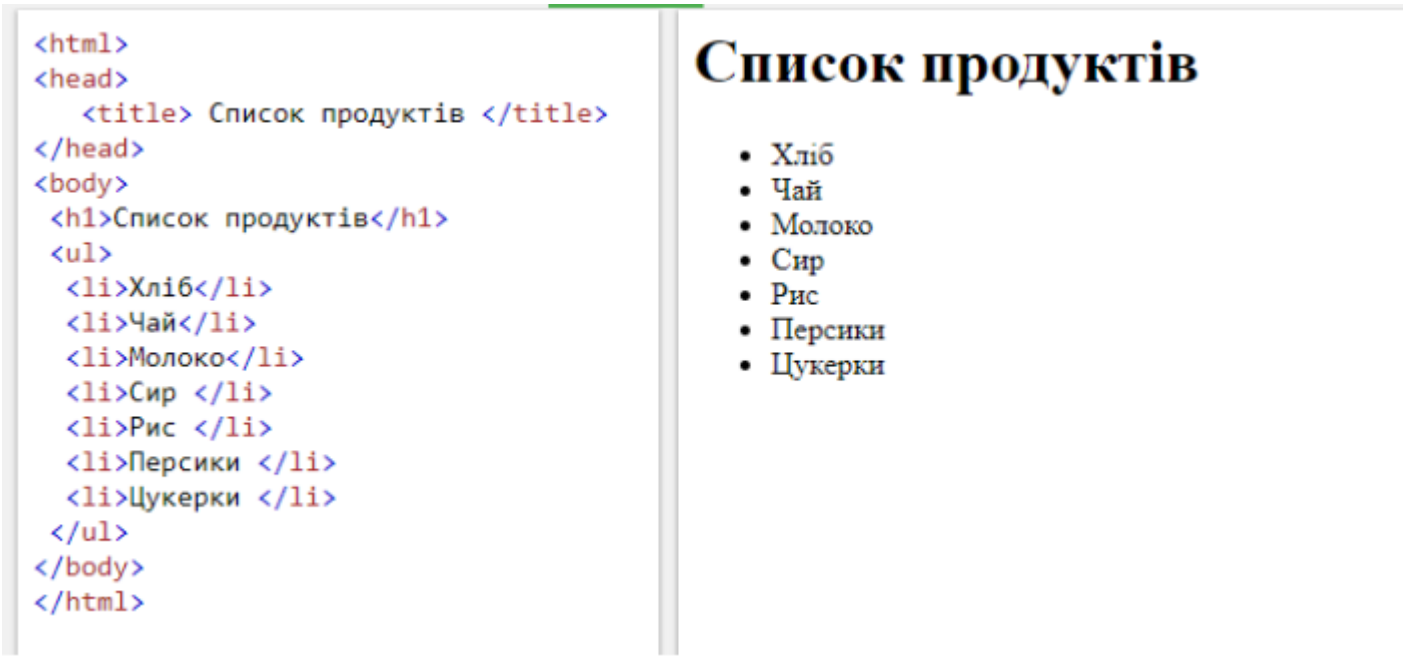

Рис. 2.2. Прилкад роботи HTML

2.3.CSS

CSS (Cascading Style Sheets) відіграє ключову роль у веб-розробці, надаючи інструменти для стилізації та візуального форматування веб-сторінок. Він працює у тандемі з HTML, який визначає структуру веб-сторінки, дозволяючи розробникам контролювати вигляд сторінки, від шрифтів та кольорів до компонування та розташування елементів. Основними перевагами використання CSS є його гнучкість

‘
і здатність до створення консистентного і однорідного візуального досвіду користувача на різних пристроях, що є критично важливим для інтернет-магазинів, орієнтованих на надання зручності та привабливості для користувачів.

Однією з ключових концепцій CSS є каскадність, що дозволяє розробникам визначати загальні стилі для всього сайту та модифікувати їх для окремих елементів або сторінок. Ця особливість значно спрощує управління стилями, оскільки не потрібно переписувати весь код при необхідності внесення змін.

Адаптивний дизайн, реалізований за допомогою медіа-запитів у CSS, є однією з найважливіших тенденцій у веб-розробці. Він забезпечує автоматичну адаптацію веб-сторінок до різних розмірів екранів, зокрема на мобільних пристроях, що є необхідним для забезпечення високої доступності та зручності сайтів у сучасному світі, де значна частина трафіку припадає на мобільні пристрої.

CSS також пропонує широкий спектр фреймворків та препроцесорів, які значно спрощують і оптимізують процес розробки. Фреймворки, такі як Bootstrap, надають готові до використання компоненти дизайну, які можуть бути легко інтегровані та налаштовані для створення професійних дизайнів. Препроцесори, такі як SASS або LESS, надають розширені можливості для написання CSS, включаючи змінні, функції та міксини, що дозволяє створювати більш гнучкі та масштабовані стилі.

Загалом, CSS є незамінним інструментом у веб-розробці, який дозволяє розробникам створювати привабливі, ефективні та користувацьки зручні веб-сайти. Його здатність до адаптації, широкий спектр можливостей для стилізації та підтримка різноманітних інструментів і фреймворків роблять його незамінним у створенні сучасних інтернет-магазинів, що відповідають вимогам сучасного цифрового світу.

‘

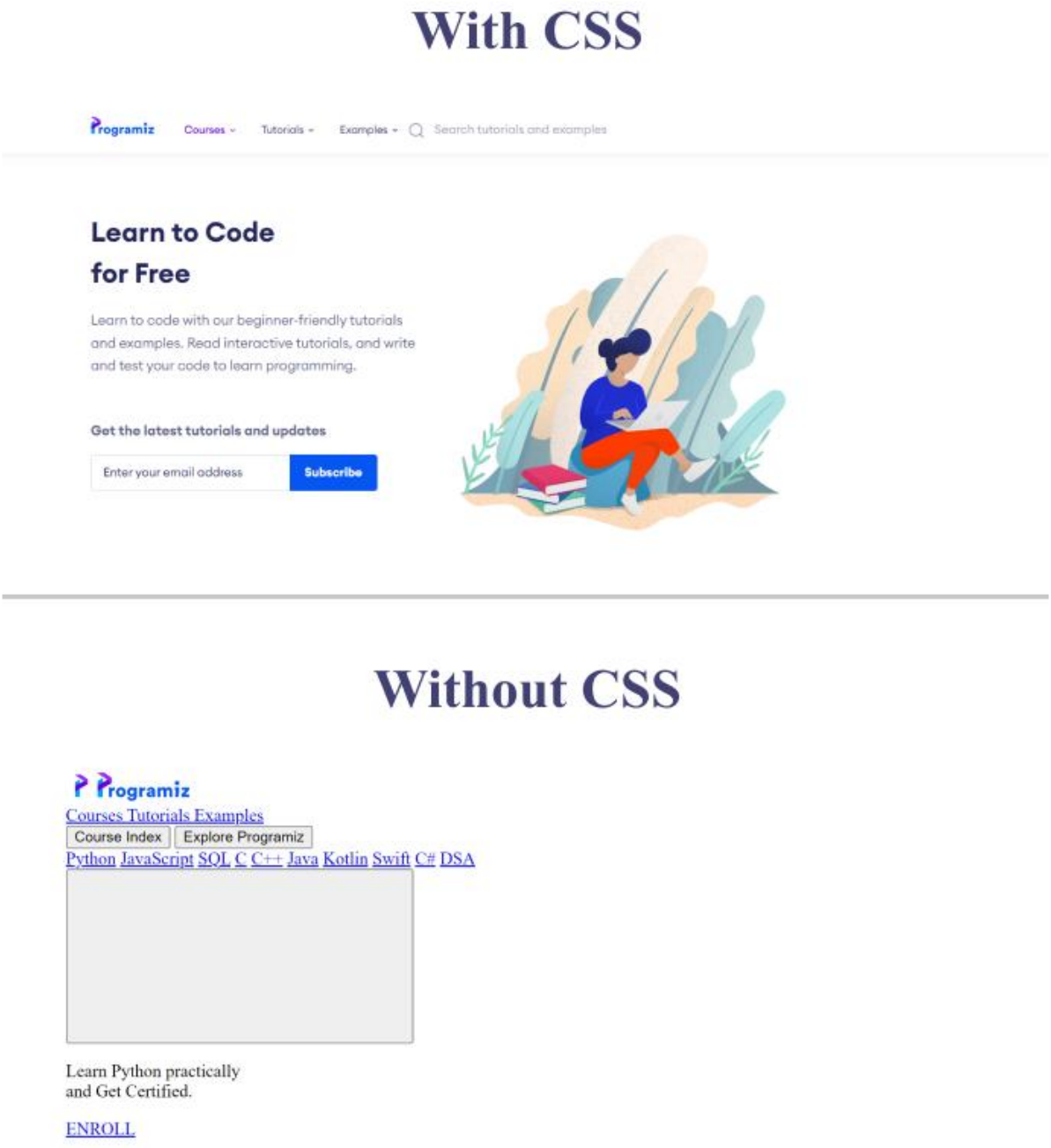

Рис. 2.3. Прилкад роботи CSS

2.4.React

React, розроблена Facebook, є однією з найпопулярніших JavaScript бібліотек для створення користувацьких інтерфейсів. Особливо ефективна для розробки односторінкових додатків, React дозволяє розробникам створювати великі веб-додатки, які можуть оновлювати дані без перезавантаження сторінки. Це робить

‘
React ідеальним інструментом для інтернет-магазинів, де важливо швидко реагувати на запити користувачів та забезпечувати постійне оновлення інформації.

React використовує компонентний підхід, де інтерфейс додатку розбивається на незалежні, повторно використовувані частини. Ці компоненти можуть бути розроблені, тестовані та впроваджені незалежно один від одного, що робить процес розробки більш гнучким та ефективним. Крім того, такий підхід сприяє більшій структурованості коду та спрощує його підтримку та масштабування.

Основною особливістю React є використання віртуального DOM, який оптимізує процес оновлення інтерфейсу. На відміну від традиційного DOM, віртуальний DOM дозволяє React проводити мінімальні маніпуляції з реальним DOM, що значно покращує продуктивність додатків, особливо при високому обсязі даних або складних інтерфейсах.

Інтеграція React з TypeScript вносить додаткові переваги. TypeScript надає статичну типізацію, що підвищує надійність коду, забезпечуючи перевірку типів на етапі компіляції. Це допомагає уникнути помилок у роботі з даними та підвищує якість спільної роботи розробників. Використання TypeScript з React також підвищує читабельність коду, роблячи його більш зрозумілим та легким для підтримки, особливо у великих проектах.

React і TypeScript разом створюють потужне середовище для розробки інтернет-магазинів. Вони дозволяють розробникам створювати додатки, які швидко реагують на дії користувачів, ефективно обробляють дані та легко адаптуються до змін, що є особливо важливим для електронної комерції. Це дозволяє створювати не тільки візуально привабливі, але й технічно передові рішення, які підвищують задоволеність користувачів та сприяють зростанню продажів.

Крім того, React підтримує широкий спектр додаткових інструментів та бібліотек, таких як Redux для управління станом додатків та React Router для навігації в SPA. Ці інструменти дозволяють розробникам створювати більш складні та функціонально багаті додатки. У випадку інтернет-магазинів, це може означати більш гнучке управління користувацькими сесіями, персоналізовані рекомендації

‘
продуктів, інтеграцію з системами аналітики та CRM, що в кінцевому підсумку підвищує загальну ефективність та конкурентоспроможність інтернет-магазину.

2.5.Bootstrap

Bootstrap, як один з найбільш впізнаваних фронтенд фреймворків, відіграє ключову роль у розробці інтернет-магазинів, надаючи розробникам інструменти для створення ефективних, адаптивних веб-інтерфейсів. Цей фреймворк, заснований на HTML, CSS і JavaScript, забезпечує велику гнучкість у дизайні, дозволяючи легко створювати адаптивні макети, які відмінно виглядають на різних пристроях, від мобільних телефонів до настільних комп'ютерів.

Основою Bootstrap є його сіткова система, яка дозволяє розробникам легко створювати складні макети з використанням готових класів. Це знімає потребу в ручному кодуванні стилів, значно прискорюючи процес розробки. Система сітки в Bootstrap гнучка та легко адаптується до різних розмірів екрану, забезпечуючи рівномірне та приємне представлення контенту.

Bootstrap також включає в себе велику кількість готових компонентів, таких як кнопки, форми, навігаційні панелі, інструменти для створення модальних вікон, вкладок, індикаторів прогресу та багато іншого. Ці компоненти можна легко кастомізувати, що дає можливість створювати унікальний дизайн, не втрачаючи при цьому переваг стандартизованого коду.

Додатковою перевагою Bootstrap є його дружність до мобільних пристроїв. Він містить вбудовані стилі для мобільної адаптивності, що дозволяє веб-сайтам автоматично адаптуватися до різних розмірів екранів. Ця особливість надзвичайно важлива в сучасному світі, де велика кількість користувачів віддає перевагу мобільним пристроям для інтернет-шопінгу.

Використання Bootstrap значно спрощує процес інтеграції з іншими технологіями та бібліотеками. Наприклад, Bootstrap часто використовується у поєднанні з JavaScript фреймворками, такими як React або Angular, що дозволяє розробникам створювати більш інтерактивні та динамічні веб-додатки.

Окрім того, наявність великої спільноти та обширної документації робить

‘
Bootstrap доступним навіть для новачків у веб-розробці. Широка підтримка та регулярні оновлення забезпечують стабільність та безпеку використання фреймворку у комерційних проектах.

У підсумку, Bootstrap є надзвичайно потужним інструментом у руках веб-розробника. Його здатність забезпечити швидку та ефективну розробку, при цьому підтримуючи гнучкість та адаптивність дизайну, робить його незамінним вибором для створення сучасних, візуально привабливих та функціональних інтернет-магазинів.

2.6.ASP Core

ASP.NET Core, розроблений Microsoft, є відкритим та крос-платформенним фреймворком для створення сучасних інтернет-застосунків та API. Він є еволюцією старшого ASP.NET, пропонуючи більшу модульність, покращену продуктивність та підтримку крос-платформенної роботи. ASP.NET Core пропонує збалансоване поєднання продуктивності, масштабованості та гнучкості, що робить його ідеальним для використання у різноманітних сценаріях веб-розробки, включаючи створення інтернет-магазинів.

Однією з основних переваг ASP.NET Core є його модульна архітектура. Вона дозволяє розробникам включати лише ті компоненти, які їм потрібні для їх конкретного застосунку, тим самим знижуючи навантаження на систему та покращуючи продуктивність. Ця модульність також сприяє кращому управлінню залежностями та забезпечує більш чистий та організований код.

Ще однією ключовою характеристикою ASP.NET Core є підтримка крос-платформенності. Це означає, що застосунки, розроблені на ASP.NET Core, можуть бути розгорнуті та виконуватися на різних операційних системах, таких як Windows, Linux та macOS. Така гнучкість є важливою для сучасних бізнес-сценаріїв, де застосунки можуть потребувати розгортання на різноманітних платформах.

ASP.NET Core також включає покращену підтримку для роботи з контейнерами, включаючи Docker, що дозволяє легко упаковувати, розгортати та масштабувати веб-застосунки. Це відіграє велику роль в сучасних облачних

‘
сценаріях, де можливість швидкого масштабування та ефективного управління ресурсами є ключовою.

У контексті безпеки, ASP.NET Core має вбудовані функції безпеки, що включають автентифікацію, авторизацію, захист від атак XSS та CSRF. Це забезпечує розробникам набір інструментів для створення захищених веб-застосунків, що є критично важливим для інтернет-магазинів, де обробляються чутливі дані користувачів.

ASP.NET Core також пропонує значні можливості для оптимізації продуктивності. Використання асинхронного програмування за допомогою async та await дозволяє ефективно використовувати ресурси сервера, забезпечуючи високу продуктивність застосунків навіть при високих навантаженнях. Це особливо важливо для електронної комерції, де швидкість обробки запитів може впливати на задоволеність користувачів та конверсію продажів.

ASP.NET Core також включає покращену підтримку для розробки API, що є основою для мікросервісної архітектури та взаємодії з різними клієнтськими застосунками, включаючи мобільні додатки та односторінкові веб-додатки. Це дозволяє створювати більш гнучкі, масштабовані та взаємодіючі системи, які можуть легко адаптуватися до змінюваних бізнес-вимог.

На закінчення, ASP.NET Core є вибором багатьох розробників для створення надійних, масштабованих та ефективних веб-застосунків. Його використання в інтернет-магазинах забезпечує не тільки високу продуктивність та безпеку, але й гнучкість та масштабованість, необхідні для адаптації до постійно змінюваних вимог ринку та поведінки користувачів.

2.7.Entity Core

Entity Framework Core, відомий як EF Core, є сучасним ORM фреймворком Microsoft, який використовується для розробки .NET додатків. Він дозволяє розробникам ефективно працювати з базами даних, перетворюючи об'єкти .NET на SQL запити та навпаки. Це робить розробку інтернет-магазинів більш інтуїтивною та продуктивною, оскільки розробники можуть використовувати відомий їм C# код

‘
для управління даними.

EF Core призначений для прискорення процесу розробки і забезпечує високу гнучкість. Його можливість автоматичного відображення класів C# на таблиці бази даних дозволяє розробникам уникнути багато ручної роботи, яка пов'язана з традиційними підходами до роботи з базами даних. Використовуючи EF Core, розробники можуть швидко створювати, читати, оновлювати та видаляти дані, зосереджуючись на бізнес-логіці, а не на взаємодії з базою даних.

Однією з ключових переваг EF Core є його підтримка Code-First підходу. Це означає, що розробники можуть спочатку визначити свої моделі даних у коді, а потім EF Core сформує відповідні таблиці бази даних. Цей підхід значно спрощує процеси міграції та оновлення схем бази даних.

EF Core також відрізняється високою продуктивністю. Він оптимізований для роботи з великими наборами даних, що є критично важливим для інтернет-магазинів, де потрібно обробляти великі обсяги інформації про товари, замовлення та користувачів.

Таким чином, використання EF Core в проектах інтернет-магазину забезпечує ефективне управління даними, підвищує продуктивність розробки та дозволяє розробникам зосередитися на створенні якісного продукту.

2.8.ProstageSQL

PostgreSQL, відома як надійна і потужна система управління базами даних з відкритим кодом, є вибором багатьох розробників для використання в інформаційних системах, зокрема в інтернет-магазинах. Вона надає широкий спектр можливостей для управління даними, включаючи складні запити SQL, транзакційну цілісність, роботу з великими обсягами даних та підтримку розширення.

Однією з ключових переваг PostgreSQL є її масштабованість, яка дозволяє впоратися з зростаючим навантаженням веб-сайту інтернет-магазину. Ця СУБД здатна ефективно обробляти великі транзакції та запити, що є життєво важливим для забезпечення стабільності бізнес-операцій та високої якості обслуговування клієнтів. Така масштабованість робить PostgreSQL привабливою для інтернет-

магазинів, які планують збільшення обсягу продажів та асортименту товарів.

Безпека в PostgreSQL забезпечується через різноманітні механізми, такі як шифрування даних та розширені можливості аутентифікації. Це дозволяє захистити конфіденційну інформацію, особливо важливу для інтернет-магазинів, які зберігають дані про кредитні картки та особисті дані користувачів.

Простота використання PostgreSQL також виявляється в її інтуїтивно зрозумілому інтерфейсі, який полегшує адміністрування баз даних та роботу з даними, не потребуючи від користувачів глибоких знань в області СУБД.

Гнучкість PostgreSQL дозволяє реалізувати складні бізнес-вимоги інтернет-магазину, адаптувати систему під конкретні завдання та інтегруватися з іншими системами та додатками.

Окрім того, широка підтримка PostgreSQL спільнотою та розробниками забезпечує постійне оновлення і вдосконалення системи, що робить її сучасною та актуальною для використання в інтернет-магазинах, що шукають стабільну та перевірену платформу для своїх даних.

2.9.Середовище розробки Visual Studio

Visual Studio, розроблене компанією Microsoft, являє собою одне з найбільш вдосконалених інтегрованих середовищ розробки (IDE), що використовується у всьому світі для створення широкого спектру програмних продуктів. Це середовище є вибором багатьох професіоналів у галузі розробки програмного забезпечення, завдяки його гнучкості, потужним інструментам і широким можливостям для розробки як настільних, так і веб- та мобільних застосунків.

Однією з основних особливостей Visual Studio є його багатофункціональність. IDE підтримує багато мов програмування, включаючи C#, C++, Visual Basic.NET, JavaScript, TypeScript, Python та інші, що робить його універсальним інструментом для розробників різних спеціалізацій. Відмінна підтримка різних мов програмування дозволяє розробникам використовувати Visual Studio для створення різноманітних проектів, від простих настільних додатків до складних хмарних рішень.

Visual Studio також відоме своїми передовими можливостями для

‘
налагодження та діагностики. Це IDE надає розширені інструменти для виявлення та усунення помилок у коді, що є критично важливим для ефективної розробки програмного забезпечення. Вбудовані інструменти налагодження дозволяють розробникам швидко знаходити та виправляти помилки, оптимізувати продуктивність і забезпечувати високу якість кінцевого продукту.

Іншою важливою характеристикою Visual Studio є інтеграція з системами контролю версій, такими як Git. Це забезпечує ефективне управління кодом і сприяє кращій командній роботі. Розробники можуть легко вести роботу над спільними проектами, координувати зміни та відстежувати історію розвитку проекту, використовуючи можливості цього середовища.

Visual Studio також пропонує потужні можливості для розробки та управління базами даних. Воно включає інструменти для проектування, розробки та тестування баз даних, що дозволяє розробникам інтегрувати роботу з даними безпосередньо в процес розробки програмного забезпечення.

Враховуючи ці характеристики та можливості, Visual Studio виступає як незамінний інструмент в арсеналі сучасного розробника програмного забезпечення.

Висновок до 2 розділу

У другому розділі дипломної роботи було розглянуто ключові методи та засоби реалізації інтернет-магазину наукової літератури, зокрема, застосування архітектури MVC, технологій HTML, CSS, JavaScript, а також використання фреймворків та бібліотек як React і Bootstrap, середовища розробки Visual Studio, і систем управління базами даних, зокрема PostgreSQL. Аналіз цих технологій та підходів дозволив виявити їх значний потенціал у створенні ефективного, гнучкого та масштабованого веб-сайту, орієнтованого на забезпечення доступу до наукових матеріалів.

Використання архітектурного шаблону MVC сприяє чіткому розділенню бізнес-логіки, інтерфейсу користувача та взаємодії з даними, що забезпечує високу якість коду, спрощує подальшу підтримку та розвиток проекту. HTML і CSS формують основу для створення користувацького інтерфейсу, забезпечуючи його

коректне відображення та адаптивність. React додає інтерактивності та динаміки веб-сторінкам, підвищуючи залученість користувачів, тоді як Bootstrap сприяє швидкій та ефективній розробці візуально привабливого дизайну з мінімальними зусиллями.

ASP.NET Core та Entity Framework Core забезпечують потужний та гнучкий бекенд, оптимізуючи взаємодію з базами даних і бізнес-логіку додатку. Використання PostgreSQL як системи управління базами даних гарантує високу продуктивність, надійність та масштабованість проекту, важливі для обробки великої кількості наукових матеріалів. Нарешті, Visual Studio як інтегроване середовище розробки спрощує процес кодування, тестування та впровадження програмного забезпечення, підвищуючи ефективність розробки.

У підсумку, детальний аналіз методів та засобів реалізації інтернет-магазину наукової літератури підкреслив їх важливість для створення сучасного, ефективного та користувацьки орієнтованого веб-сайту. Ці технології та підходи не лише сприяють оптимізації робочих процесів та підвищенню якості сервісу, але й відкривають широкі можливості для масштабування проекту та його адаптації до змінюваних вимог та очікувань користувачів.

‘

РОЗДІЛ 3.
РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1.Аналіз функціональних і нефункціональних вимог

Для створення інтернет-магазину наукової літератури, важливо ретельно спланувати та реалізувати низку базових функціональних вимог, що визначатимуть функціональність магазину .

1. Каталог Продукції:

- Систематизований каталог, що міститиме широкий асортимент наукових книг та публікацій.
- Кожен продукт у каталозі супроводжуватиметься детальним описом, що включатиме інформацію про авторів, зміст, обсяг, мову публікації та ISBN.

2. Система Пошуку та Фільтрації:

- Інтернет-магазин забезпечить функцію розширеної можливості пошуку та фільтрації, дозволяючи користувачам ефективно відшукувати літературу за специфічними критеріями.

3. Корзина:

- Функціонал користувацької корзини для зручності збору обраних книг та оформлення замовлень.

4. Система Реєстрації та Авторизації:

- Система для створення та управління користувацькими профілями.
- Забезпечення захищеної авторизації та надійного зберігання персональних даних користувачів.

‘

5. Оформлення Замовлення:

- Процес оформлення замовлення.
- Функція збереження історії покупок для кожного користувача для зручності перегляду попередніх замовлень.

6. Система Оплати:

- Інтеграція з надійними платіжними системами для забезпечення безпеки фінансових операцій.
- Підтримка різноманітних способів оплати: кредитні картки, електронні гаманці та банківські перекази.

7. Управління Змістом:

- Панель управління для адміністраторів, що дозволить додавати, редагувати або видаляти продукцію, а також управляти замовленнями та даними користувачів.

8. Підтримка та Допомога Користувачам:

- Надання допомоги та підтримки клієнтів через різні канали, включаючи електронну пошту та телефон.
- Імплементація розділу частих запитань (FAQ) для самостійного вирішення стандартних проблем користувачів.

9. Функціонал Оцінювання та Відгуків:

- Функціонал для користувачів залишати відгуки на книги.

Для розробки інтернет-магазину наукової літератури, наступні нефункціональні вимоги будуть взяті до уваги, забезпечуючи оптимальну роботу та користувацький досвід:

1.Час Відгуку Системи:

- Система повинна відповідати на запити користувача в межах 2-3 секунд, забезпечуючи швидке завантаження сторінок та відгук на дії користувача.

2.Продуктивність під Високим Навантаженням:

- Система має зберігати стабільну продуктивність навіть під час піків навантаження, наприклад, під час розпродажів або промоційних акцій.

3.Надійність та Відсоток Часу Роботи:

‘

- Мета – забезпечити 99.5% часу безперебійної роботи, мінімізуючи час простою системи.

4.Сумісність з Браузерами:

- Система має бути сумісною з основними версіями популярних веб-браузерів, таких як Chrome, Firefox, Safari та Edge.

5.Масштабованість:

- Система має бути розроблена з можливістю легкого масштабування, аби витримувати збільшення кількості користувачів та обсягу даних.

3.2 Розробка Моделі веб-сервісу та бази даних

Визначення структури бази даних, яка є необхідною для забезпечення ефективної роботи інтернет-магазину з продажу наукової літератури. Важливість цього процесу полягає в створенні оптимальної архітектури, котра дозволить ефективно зберігати, викликати та обробляти дані, включаючи каталоги товарів, інформацію про користувачів, замовлення, та дані транзакцій. Метою є розробка надійної, масштабованої та безпечної системи, котра сприятиме поліпшенню користувацького досвіду, підвищенню продуктивності бізнес-процесів та забезпеченню гнучкості у управлінні даними.

Сутності Бази Даних:

- Сутність Author описує авторів у базі даних з такими атрибутами:
  - AuthorId: Унікальний ідентифікатор автора.
  - FirstName та SecondName: Ім'я та прізвище автора відповідно.
  - Discipline: Дисципліна автора.
  - AboutAuthor: Короткий опис автора .
  - HasBooks: Список книг, написаних автором, який встановлює зв'язок між авторами та їх творами.
- Сутність Book описує книги у базі даних з наступними атрибутами:
  - Id: Унікальний ідентифікатор книги.
  - Name: Назва книги.
  - Description: Опис книги.

‘

- Year: Рік видання.
- ISBN: Міжнародний стандартний книжковий номер.
- isEBook: Позначення електронного формату книги.
- Page: Кількість сторінок.
- BookCost: Вартість книги.
- Language: Мова книги.
- ImgRef: Посилання на обкладинку книги.
- RelatedBookId: Ідентифікатор пов'язаної книги.
- DataAdd: Дата додавання книги до бази.
- HasAuthors: Список авторів книги.
- HasSubjects: Список предметів або тем, до яких належить книга.

- Сутність CardItem відображає елемент корзини покупок у базі даних, містить наступні атрибути:
  - Id: Унікальний ідентифікатор елемента корзини.
  - BookId: Ідентифікатор книги, яка додається до корзини.
  - Quantity: Кількість копій книги у корзині.
  - ShopingCardId: Ідентифікатор корзини покупок, до якої належить елемент.
- Сутність ShopingCard представляє корзину покупок користувача і включає:
  - Id: Унікальний ідентифікатор корзини.
  - UserId: Ідентифікатор користувача, власника корзини.
  - CardItems: Колекція елементів (CardItem), які додаються до корзини.
- Сутність OrderHeader описує заголовок замовлення з такими полями:
  - OrderHeaderId: Унікальний ідентифікатор заголовка замовлення.
  - PickUpName, PickUpPhone, PickUpEmail: Інформація для зв'язку з покупцем.
  - UserId : Ідентифікатор користувача, який робить замовлення.
  - OrderTotal: Загальна сума замовлення.
  - StripePaymentId: Ідентифікатор платежу в Stripe.
  - OrderDate: Дата створення замовлення.
  - OrderStatus: Статус замовлення.
  - OrderTotalItem: Загальна кількість товарів у замовленні.

‘

- OrderDetails: Деталі замовлення, що включають товари замовлення.

- Сутність OrderDetail описує деталі замовлення в базі даних, включаючи:
  - OrderDetailId: Унікальний ідентифікатор деталі замовлення.
  - OrderHeaderId: Ідентифікатор заголовку замовлення, до якого належить ця деталь.
  - BookId: Ідентифікатор книги в замовленні.
  - Quantity: Кількість копій книги в замовленні.
  - IteamName: Назва товару (книги).
  - Price: Ціна за одиницю товару.
- Сутність Subject визначає предмет книг у базі даних:
  - SubjectId: Унікальний ідентифікатор предмета.
  - SubjectName: Назва предмета.
  - HasBooks: Список книг, асоційованих з цим предметом.
- Сутність User включає:
  - Name і SecondName: Ім'я та прізвище користувача.
  - FullName: Автоматично сформоване повне ім'я користувача, що є комбінацією Name та SecondName.
- Сутність BookComment в базі даних включає:
  - BookCommentId: Унікальний ідентифікатор коментаря.
  - BookId: Ідентифікатор книги, до якої залишено коментар.
  - UserId: Ідентифікатор користувача, який залишив коментар.
  - Text: Текст коментаря.

У процесі розробки бази даних для інтернет-магазину наукової літератури, крім створення специфічних сутностей було також інтегровано використання таблиць Identity Framework. Ця інтеграція дозволила ефективно управляти користувачами та їхніми ролями в системі, забезпечуючи надійні механізми безпеки, такі як шифрування паролів та управління доступом.

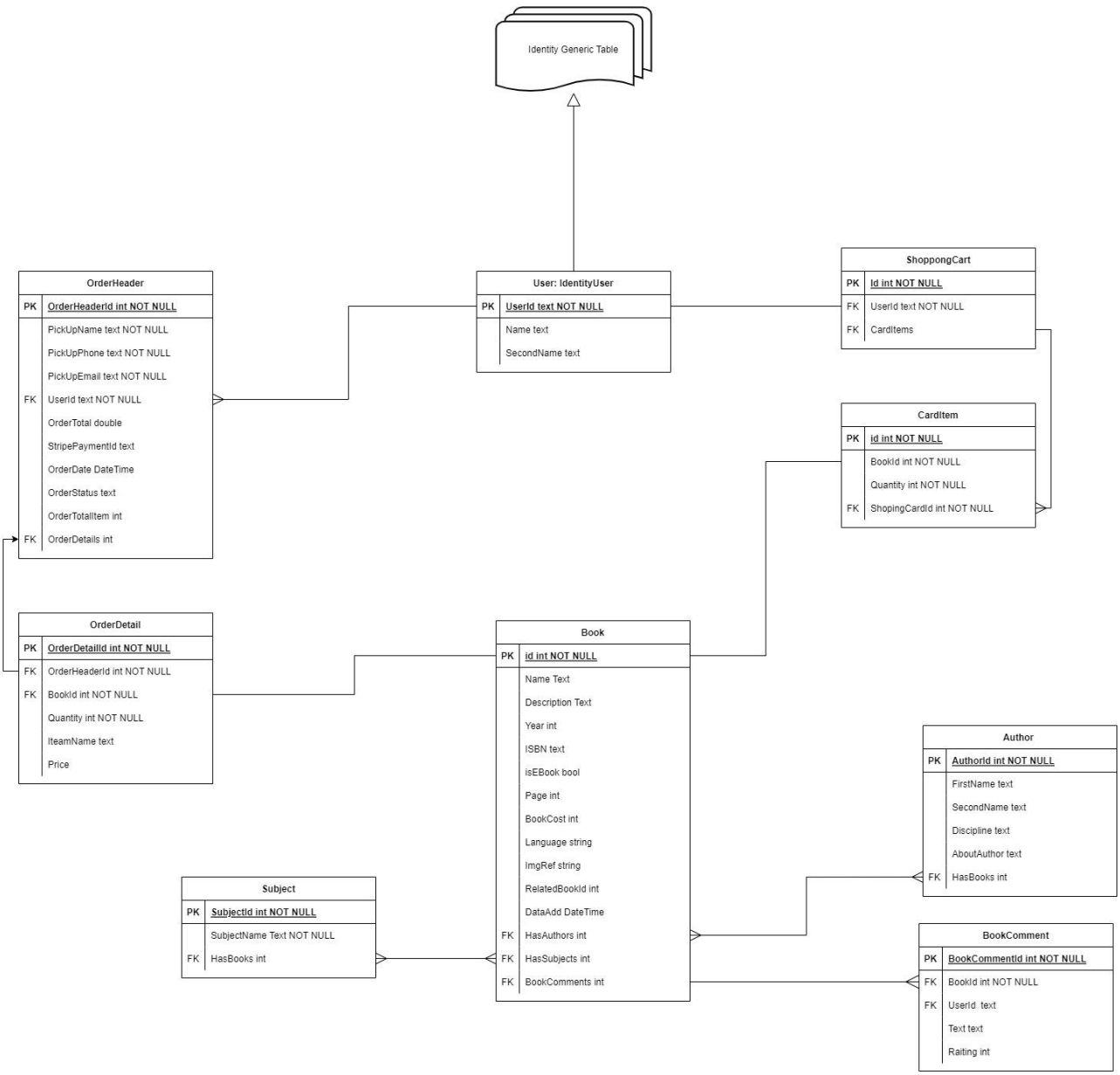

Рис. 3.1. ER діаграма веб сервісу

Для створення бази даних моєї дипломної роботи було використано Entity Framework (EF) як основний інструмент ORM. Цей інструмент відіграв фундаментальну роль у процесі розробки, дозволивши ефективно взаємодіяти з базою даних через об'єктно-орієнтоване програмування.

Першочерговою задачею було налаштування DbContext - спеціалізованого класу, який служить в якості мосту між сутностями в коді та таблицями в базі даних. Створення такого класу дало змогу визначити моделі даних, які представляють необхідні сутності, що відповідають за зберігання інформації про книги, авторів, жанри та замовлення у базі даних. Цей підхід значно спростив роботу з даними,

дозволивши виконувати складні запити та операції з базою даних на високому рівні абстракції.

```
using Microsoft.EntityFrameworkCore;
namespace Back.Data
{
    Ссылок: 27
    public class AppDbContext : IdentityDbContext<User>
    {
        Ссылок: 0
        public AppDbContext(DbContextOptions option):base(option) {   }
        Ссылок: 8
        public DbSet<Book> Books { get; set; }
        Ссылок: 3
        public DbSet<User> Users { get; set; }
        Ссылок: 7
        public DbSet<Author> Authors { get; set; }
        Ссылок: 0
        public DbSet<Subject> Subjects { get; set; }

        Ссылок: 5
        public DbSet<ShopingCard> shopingCards { get; set; }
        Ссылок: 0
        public DbSet<CardItem> CardItems { get; set; }
        Ссылок: 1
        public DbSet<OrderDetail> orderDetails { get; set; }
        Ссылок: 5
        public DbSet<OrderHeader> OrderHeaders { get; set; }
        Ссылок: 0
        public DbSet<BookComment> BookComments { get; set; }



    }
}
```

Рис. 3.2. Клас AppDbContext

Далі була здійснена конфігурація ConnectionString у файлі конфігурації проекту. Цей рядок з'єднання містить критично важливі деталі, які забезпечують з'єднання додатку з базою даних, включаючи ім'я сервера, назву бази даних, а також параметри аутентифікації. Завдяки цьому етапу, встановлено надійний канал комунікації між програмним додатком та ProstageSQL

```
"ConnectionStrings": {
  "DefaultDBConnection": "Host=localhost;Database=DiploDB;Username=postgres;Password="
},
```

Рис. 3.3. Налаштування ConnectionString

Наступним важливим кроком стала реєстрація DbContext як сервіс у файлі Program.cs проекту, використовуючи метод AddDbContext. Це дозволило

‘
інтегрувати контекст бази даних у систему управління залежностями додатку, забезпечуючи легкий доступ до нього з будь-якої частини програми. Така організація сприяла більш ефективному та гнучкому управлінню ресурсами бази даних, а також спрощенню розробки додатку за рахунок використання принципів ін'єкції залежностей.

```
builder.Services.AddDbContext<AppDbContext>(option =>
{
    option.UseNpgsql(builder.Configuration.GetConnectionString("DefaultDBConnection"));
});
```

Рис. 3.4. Реєстрація DbContext

Завершальним етапом роботи над базою даних стало виконання міграцій за допомогою команд Entity Framework. Міграції автоматично генерують код, необхідний для оновлення схеми бази даних на основі актуального стану моделей та конфігурацій, визначених в DbContext.

```
namespace Back.Migrations
{
    /// <inheritdoc />
    Ссылок: 1
    public partial class AddComent : Migration
    {
        /// <inheritdoc />
        Ссылок: 0
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "BookComments",
                columns: table => new
                {
                    BookCommentId = table.Column<int>(type: "integer", nullable: false)
                        .Annotation("Npgsql:ValueGenerationStrategy", NpgsqlValueGenerationStrategy.IdentityByDefaultColumn),
                    BookId = table.Column<int>(type: "integer", nullable: false),
                    UserId = table.Column<string>(type: "text", nullable: true),
                    Text = table.Column<string>(type: "text", nullable: true),
                    Raiting = table.Column<int>(type: "integer", nullable: false)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_BookComments", x => x.BookCommentId);
                    table.ForeignKey(
                        name: "FK_BookComments_AspNetUsers_UserId",
                        column: x => x.UserId,
                        principalTable: "AspNetUsers",
                        principalColumn: "Id");
                    table.ForeignKey(
                        name: "FK_BookComments_Books_BookId",
                        column: x => x.BookId,
                        principalTable: "Books",
```

Рис. 3.5. Згенерована Міграція

У підсумку, процес створення бази даних для інтернет-магазину наукової літератури з використанням Entity Framework виявився досить структурованим та ефективним.

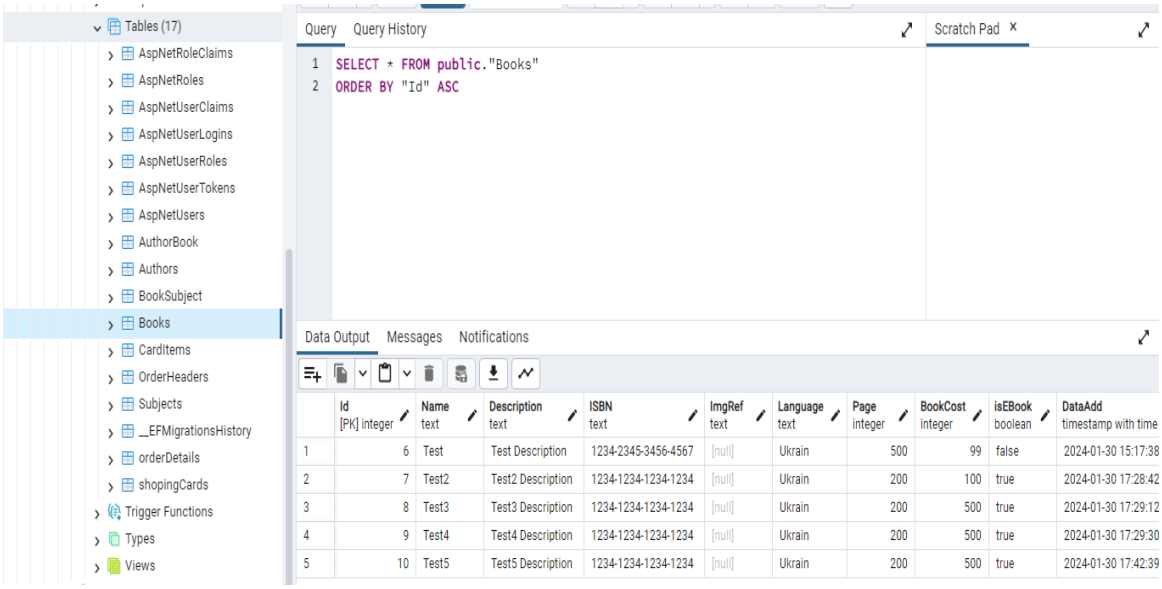

Рис. 3.6. Результат створення БД

3.3. Розробка контролерів веб-сервісу

У роботі я зосередив увагу на розробці контролерів, які відіграють базові ролі у взаємодії між користувацьким інтерфейсом і базою даних. Також створив і використав структуру ApiResponce, що дозволило стандартизувати відповіді API, забезпечуючи зручність обробки результатів на клієнтській стороні.

Контроллер AuthorController, відповідає за управління інформацією про авторів.

Опис методів AuthorController:

GetAllAuthor - цей метод відповідає за видачу списку всіх авторів. Він перевіряє, чи існують автори в базі даних, і у випадку їх відсутності повертає статус 404 (Not Found), інакше повертає список авторів зі статусом 200 (OK).

GetOneAuthor - забезпечує отримання інформації про конкретного автора за його ідентифікатором. Якщо переданий ідентифікатор невалідний або автор не знайдений, метод повертає статус 404. У іншому випадку, повертає дані автора зі статусом 200.

AddNewAuthor - цей метод дозволяє додавати нового автора до бази даних. Він приймає дані автора, перевіряє їх на валідність і, у разі успішного додавання, зберігає автора в базі даних та повертає статус 201 з інформацією про створеного автора.

‘

UpdateAuthor - метод призначений для оновлення даних існуючого автора. Він вимагає передачі ідентифікатора автора та оновлених даних. Якщо автор з вказаним ідентифікатором існує, його дані оновлюються. У разі успішного оновлення, метод повертає оновлену інформацію зі статусом 200.

DeleteAuthor - відповідає за видалення автора з бази даних. Якщо автор з вказаним ідентифікатором не знайдений, метод повертає статус 404. У разі успішного видалення, повертає повідомлення про видалення зі статусом 200.

Контроллер BookController, відповідає за управління інформацією про книги. Опис методів BookController:

GetAllBooks - цей метод дозволяє отримати список усіх книг, доступних у магазині. У випадку, коли книги відсутні, метод повертає відповідь із статусом 404 (Not Found), інформуючи користувача про відсутність доступних книг.

GetOneBook - за допомогою цього методу можна отримати детальну інформацію про конкретну книгу за її ідентифікатором. Якщо книга з заданим ідентифікатором не знайдена, повертається статус 404 (Not Found).

AddNewBook - цей метод призначений для додавання нової книги до каталогу магазину. Він приймає дані книги і, після валідації, додає книгу до бази даних, повертаючи інформацію про додану книгу зі статусом 201 (Created).

UpdateBook - метод оновлення даних про книгу дозволяє модифікувати існуючі записи в базі даних. Якщо книга для оновлення присутня в базі, її дані оновлюються відповідно до переданих значень.

DeleteBook - відповідає за видалення книги з бази даних. Успішне видалення книги підтверджується поверненням статусу 200 (OK) з відповідним повідомленням.

AddComentar - цей метод дозволяє додавати коментарі до книг, забезпечуючи користувачам можливість обмінюватися враженнями та оцінками.

DeleteComentar - видаляє коментар за ідентифікатором, надаючи адміністрації магазину інструмент для модерації вмісту.

GetBookComentar - метод для отримання усіх коментарів до конкретної книги, що дозволяє користувачам ознайомитись з думками та відгуками інших читачів.

‘

Контроллер IdentityController, відповідальний за автентифікацію та авторизацію користувачів. Опис методів AuthorController:

RegisterUser - цей метод призначений для реєстрації нових користувачів з роллю "Користувач". Він перевіряє, чи роль "Користувач" існує в системі, і в разі відсутності створює її. Після валідації даних, метод спробує створити нового користувача та додати його до вказаної ролі, повертаючи інформацію про успішну реєстрацію або помилку.

RegisterAdmin - аналогічно методу реєстрації користувачів, цей метод дозволяє реєструвати адміністраторів. Він перевіряє наявність ролей "Адміністратор" та "Користувач" і створює їх при необхідності. Метод реєструє нового користувача з роллю "Адміністратор".

Login - метод, який дозволяє користувачам увійти в систему за допомогою імені користувача або електронної пошти та пароля. У разі успішної автентифікації генерується JWT-токен, який повертається користувачеві. Це забезпечує доступ до захищених ресурсів інтернет-магазину.

Контроллер OrderController, який є відповідальним за управління замовленнями в системі. Опис методів BookController:

GetOrders - цей метод дозволяє отримати список усіх замовлень або замовлень, зроблених конкретним користувачем. Він забезпечує фільтрацію замовлень за ідентифікатором користувача, якщо такий надано, та повертає результат у відповіді зі статусом 200 (OK).

GetOrder - за допомогою цього методу можна отримати детальну інформацію про конкретне замовлення за його ідентифікатором. Якщо замовлення знайдено, метод повертає інформацію про нього зі статусом 200 (OK). У випадку відсутності замовлення повертається статус 404 (Not Found).

CreateOrder - цей метод призначений для створення нового замовлення на основі даних, отриманих у форматі OrderHeaderDTO. Метод валідує вхідні дані та створює замовлення в базі даних, повертаючи інформацію про створене замовлення зі статусом 201 (Created).

‘

UpdateOrder - метод дозволяє оновити дані існуючого замовлення. Якщо замовлення з заданим ідентифікатором існує, його дані оновлюються відповідно до наданих значень. Успішне оновлення підтверджується поверненням статусу 200 (OK).

Контроллер PaymentController, відповідає за обробку платежів в системі. Використовуючи Stripe як платіжний шлюз, цей контролер дозволяє здійснювати безпечні транзакції безпосередньо через веб-сервіс.

Опис методу MakePayment :

MakePayment - цей метод призначений для ініціювання процесу оплати за допомогою Stripe. Він приймає ідентифікатор користувача (UserId) та перевіряє, чи існує користувацька кошик (ShopingCard) з товаром для цього користувача. У разі, якщо кошик відсутній або порожній, метод повертає помилку з відповідним повідомленням.

Після перевірки наявності та наповнення кошика, MakePayment використовує API Stripe для створення наміру оплати (PaymentIntent). В цьому процесі вказується загальна сума покупок, валюта транзакції, та методи оплати, які будуть прийнятні для цієї операції. Після створення наміру оплати, ідентифікатор та секретний ключ наміру зберігаються в об'єкті кошика для подальшої обробки платежу на клієнтській стороні.

У разі успішного створення наміру оплати, метод повертає дані про кошик із включеним ідентифікатором та секретом оплати зі статусом 200 (OK), що дозволяє користувачу продовжити процес оплати на клієнтській стороні.

Контроллер ShopingCardController, що відповідає за управління користувацьким кошиком. Цей контролер дозволяє виконувати основні операції з кошиком, такі як додавання та оновлення кількості товарів, а також отримання інформації про поточний стан кошика

Опис методів ShopingCardController:

AddAndUpdateItem - цей метод призначений для додавання нового товару до кошика або оновлення кількості існуючого товару. Він перевіряє існування користувача та кошика, додає товар або оновлює кількість товару в кошику. У

‘
випадку електронних книг (eBooks) кількість завжди залишається одиницею. Метод обробляє випадки, коли товар додається вперше або коли необхідно оновити кількість існуючого товару в кошику, і повертає відповідний статус операції.

GetCart - цей метод дозволяє отримати детальну інформацію про кошик користувача, включаючи список товарів у кошику, їх кількість та загальну вартість. Він забезпечує візуалізацію кошика для користувачів, дозволяючи їм переглядати обрані товари та їх загальну вартість перед оформленням замовлення.

Контроллер SubjectController, відповідає за управління предметами (категоріями) книг. Цей контролер дозволяє виконувати базові операції CRUD для предметів.

Опис методів SubjectController:

GetAllSubject - цей метод дозволяє отримати список усіх предметів, доступних у магазині. У випадку, коли предмети відсутні, метод повертає статус 404 (Not Found), інформуючи про відсутність доступних категорій.

GetOneSubjects - за допомогою цього методу можна отримати детальну інформацію про конкретний предмет за його ідентифікатором. Якщо предмет з заданим ідентифікатором не знайдено, повертається статус 404 (Not Found).

AddNewSubject - цей метод призначений для додавання нового предмета до каталогу магазину. Він приймає дані предмета і, після валідації, додає предмет до бази даних, повертаючи інформацію про створений предмет зі статусом 201 (Created).

DeleteSubject - відповідає за видалення предмета з бази даних. Успішне видалення предмета підтверджується поверненням статусу 200 (OK) з відповідним повідомленням.

3.4. Розробка користувацького інтерфейсу веб-сервісу

На основі вимог до інтернет-магазину, ідентифіковано наступні сторінки, які розроблені у моєму веб-сервісі:

1. Головна сторінка:

- Огляд основних категорій наукової літератури.

‘

- Рекомендовані та нові книги.

2. Каталог продукції:
    - Сторінка з повним списком книг, поділених за категоріями та жанрами.
    - Фільтри для сортування за авторами, мовою публікації, жанром тощо.
3. Сторінка книги:
    - Детальний опис книги, включаючи авторів, зміст, ISBN, відгуки користувачів.
    - Можливість додавати книгу до корзини.
    - Можливість залишити відгук
4. Корзина:
    - Перегляд вибраних книг з можливістю зміни кількості або видалення.
    - Оформлення замовлення.
5. Сторінка реєстрації/авторизації
    - Форми для створення нового акаунта або входу в існуючий.
6. Система оплати:
    - Інтеграція з платіжними системами.
7. Особистий кабінет користувача
    - Управління профілем.
    - Історія замовлень.
    - Відстеження статусу активних замовлень.
8. Контактна інформація.
9. FAQ
10. Про магазин

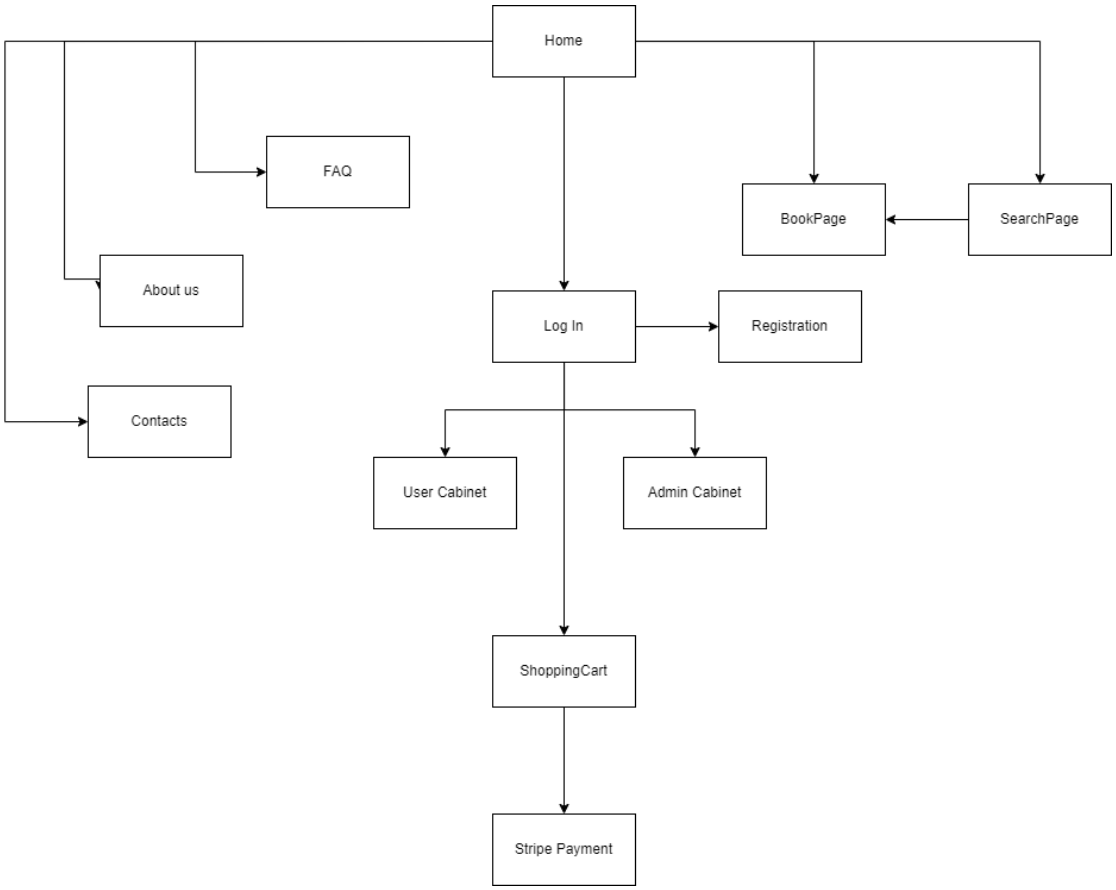

Рис. 3.7. Структурна схема сайту

В рамках розробки веб-сервісу інтернет-магазину наукової літератури, я обрав Redux Toolkit Query (RTK Query) для взаємодії з backend частиною. RTK Query спрощує процес роботи з асинхронними даними через API, автоматизуючи багато аспектів запитів і мутацій.

RTK Query використовує концепцію "API слайсів", які організовують код запитів до конкретного API в одному місці. Кожен API слайс визначає ендпойнти, доступні для взаємодії, які включають операції отримання (queries) та оновлення (mutations) даних. За допомогою цих ендпойнтів, RTK Query генерує хуки, які можна безпосередньо використовувати у компонентах React для виконання запитів або мутацій, забезпечуючи автоматичне управління станом запиту, кешування відповідей та повторне використання даних.

Коли компонент використовує хук ендпойнта для виконання запиту, RTK Query автоматично виконує запит до сервера, обробляє відповідь і зберігає отримані дані в глобальному стані Redux. Це забезпечує централізоване управління даними і

‘
їх легкий доступ з будь-якого місця додатку. Вбудовані механізми кешування дозволяють мінімізувати звернення до сервера, автоматично використовуючи кешовані дані для однакових запитів, що підвищує продуктивність і зменшує навантаження на сервер.

Основні розроблені слайси:

- Слайс BookApi:
  - getBooks:

    Запит: Виконує GET запит до Book для отримання списку всіх книг. Використання: Цей ендпойнт дозволяє отримати актуальний список книг без необхідності передачі додаткових параметрів.
  - getBookById:

    Запит: Виконує GET запит до Book/{id}, де id - це ідентифікатор конкретної книги.

    Використання: Використовується для отримання детальної інформації про певну книгу за її ID.
  - AddBook:

    Мутація: Виконує POST запит до Book для додавання нової книги. В тілі запиту передається об'єкт NewBook, який відповідає моделі NewBookModel.
  - UpdateBook:

    Мутація: Виконує PUT запит до Book/{id}, де id - це ідентифікатор книги, що оновлюється. У тілі запиту передається об'єкт UpdateBook, який відповідає моделі UpdateBookModel.
- Слайс ShopingCardApi:
  - getShopingCard:

    Запит: Виконує GET запит для отримання даних кошика покупок користувача. Приймає UserId як параметр для ідентифікації кошика.
  - AddItemInCard:

    Мутація: Виконує POST запит до ShopingCard/AddItemInCard для додавання або оновлення кількості певної книги в кошику користувача.
- Слайс authApi:

‘

- registerUser:

  Мутація: Виконує POST запит для реєстрації нового користувача.

- loginUser:

  Мутація: Виконує POST запит для входу користувача.

- Слайс authorApi Методи:
  - getAllAuthors: Запит для отримання списку всіх авторів.
  - getOneAuthor: Запит для отримання деталей про конкретного автора за ідентифікатором.
  - addNewAuthor: Мутація для додавання нового автора до бази даних.
  - updateAuthor: Мутація для оновлення даних існуючого автора.
  - deleteAuthor: Мутація для видалення автора з бази даних.
- Слайс orderApi Методи:
  - getOrdersQuery: Запит для отримання списку замовлень.
  - getOrderQuery: Запит для отримання деталей конкретного замовлення.
  - createOrderMutation: Мутація для створення нового замовлення.
  - updateOrderMutation: Мутація для оновлення деталей замовлення.
- Слайс paymentApi Методи:
  - makePaymentMutation: Мутація для ініціації процесу оплати, включаючи створення наміру оплати та обробку платежу.
- Слайс subjectApi Методи:
  - getAllSubjectsQuery: Запит для отримання списку всіх категорій (предметів).
  - getOneSubjectQuery: Запит для отримання деталей про конкретний предмет.
  - addNewSubjectMutation: Мутація для додавання нового предмета.
  - deleteSubjectMutation: Мутація для видалення предмета.

Одним з ключових аспектів є управління станом додатку. Для ефективної реалізації цієї задачі було обрано Redux . Redux суттєво спрощує процес управління станом, забезпечуючи потужний набір утиліт, які допомагають зменшити кількість повторюваного коду та підвищити читабельність. У межах дипломної роботи було реалізовано декілька ключових слайсів:

- authSlice:

‘

Відповідає за управління станом авторизації користувачів. Цей слайс містить інформацію про поточного авторизованого користувача, включаючи його ідентифікатор, електронну пошту, роль та ім'я користувача. Функція SetLogginUser використовується для оновлення цих даних у стані.

- BookSlice:

  Управляє списком книг у додатку. Цей слайс включає в себе масив книг і дозволяє оновлювати цей список за допомогою дії SetBookList.

- ShoppingCartSlice:

  Слайс призначений для управління кошиком покупок користувача. Він містить масив обраних товарів і надає можливості для додавання товарів до кошика, оновлення кількості товару та видалення товарів з кошика.

В процесі розробки веб-сервісу мого інтернет-магазину наукової літератури було створено кілька ключових сторінок. Ці сторінки розроблені з метою забезпечити легкий доступ до наукових книг, зручність управління кошиком покупок і простоту в процесі оформлення замовлення.

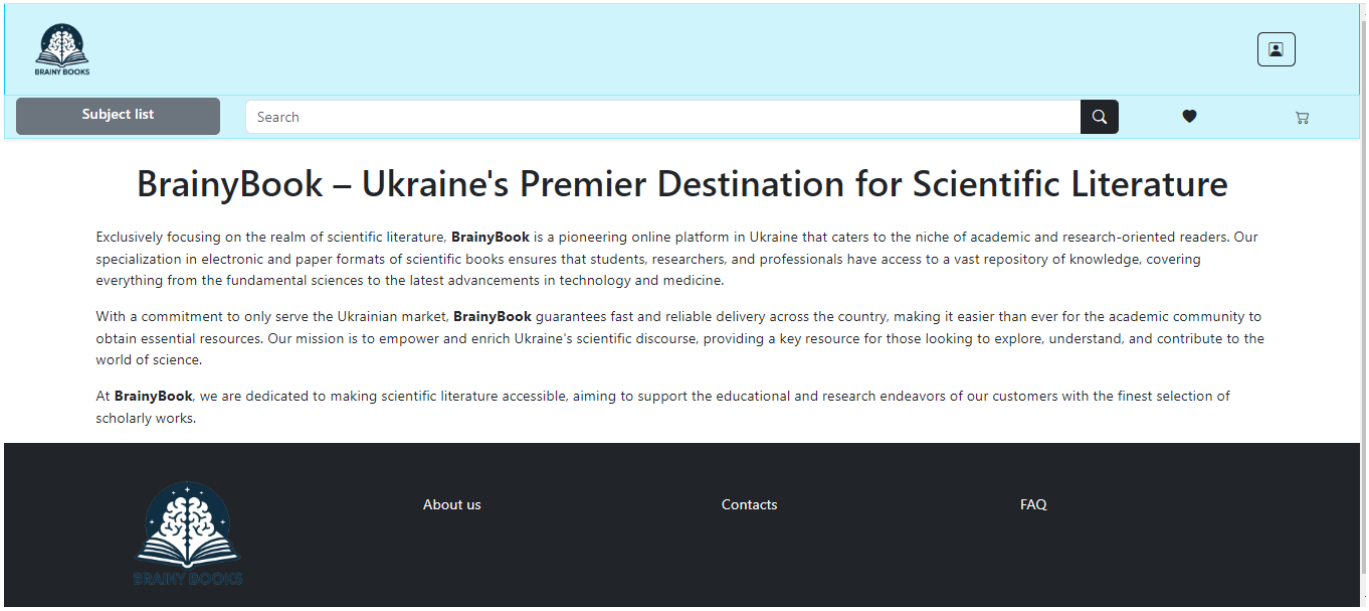

Рис. 3.8. Сторінка Про нас

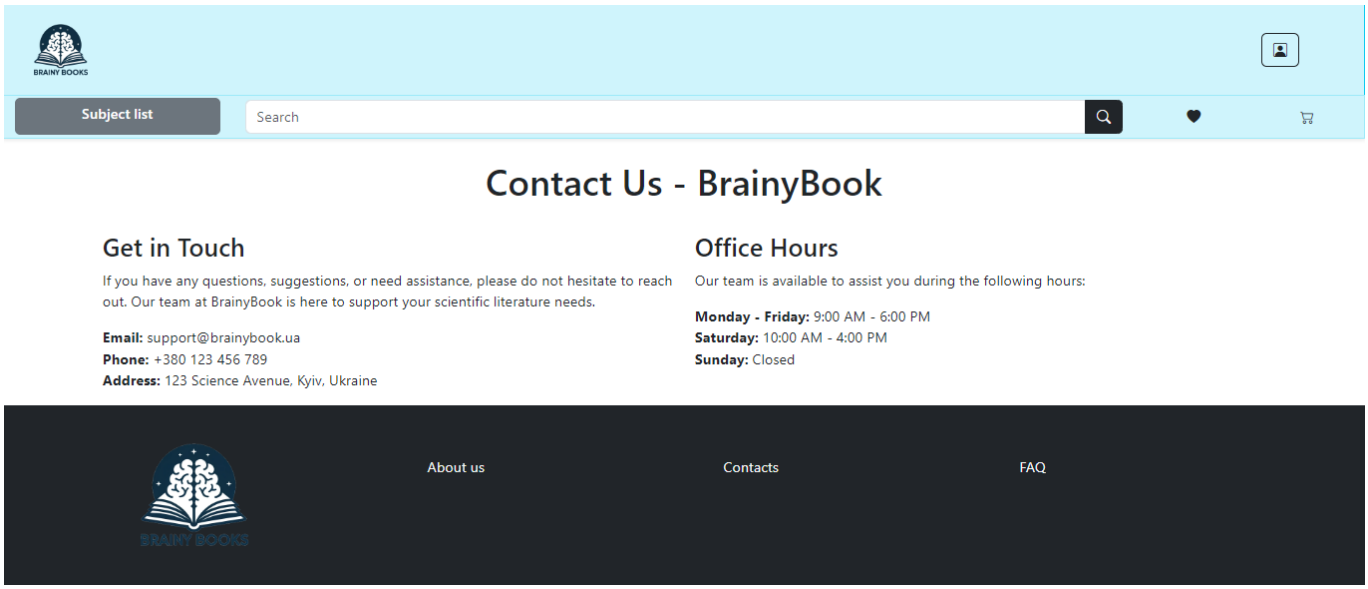

Рис. 3.9. Сторінка Контакти

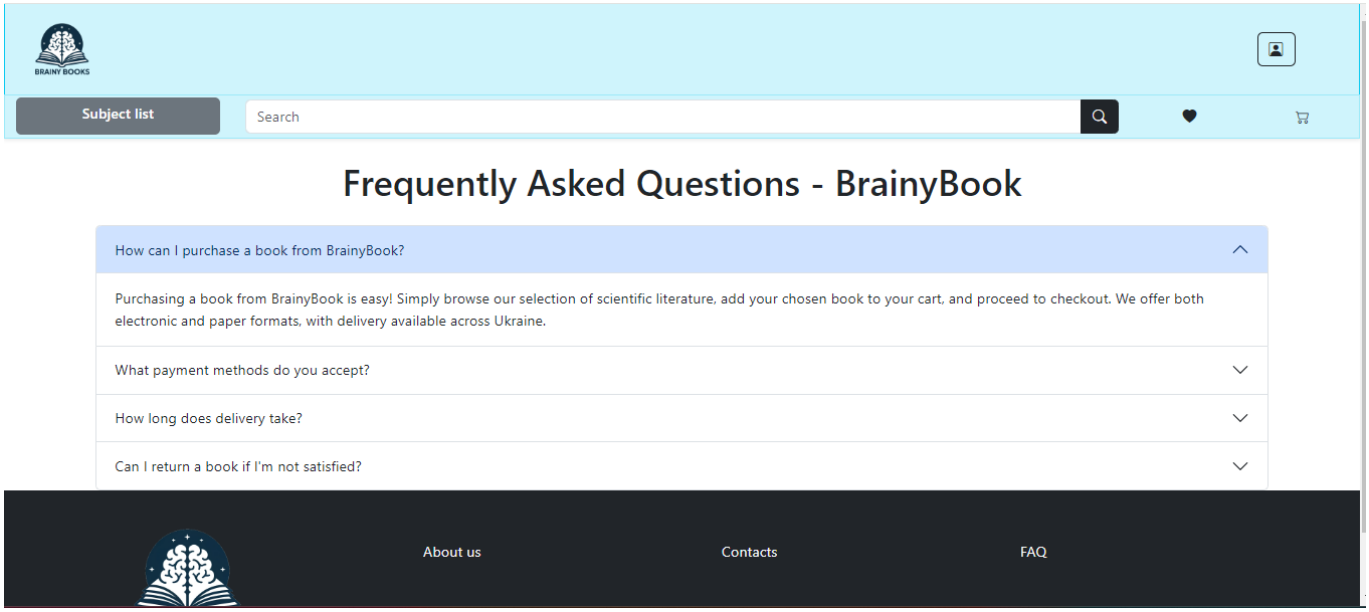

Рис. 3.10. Сторінка FAQ

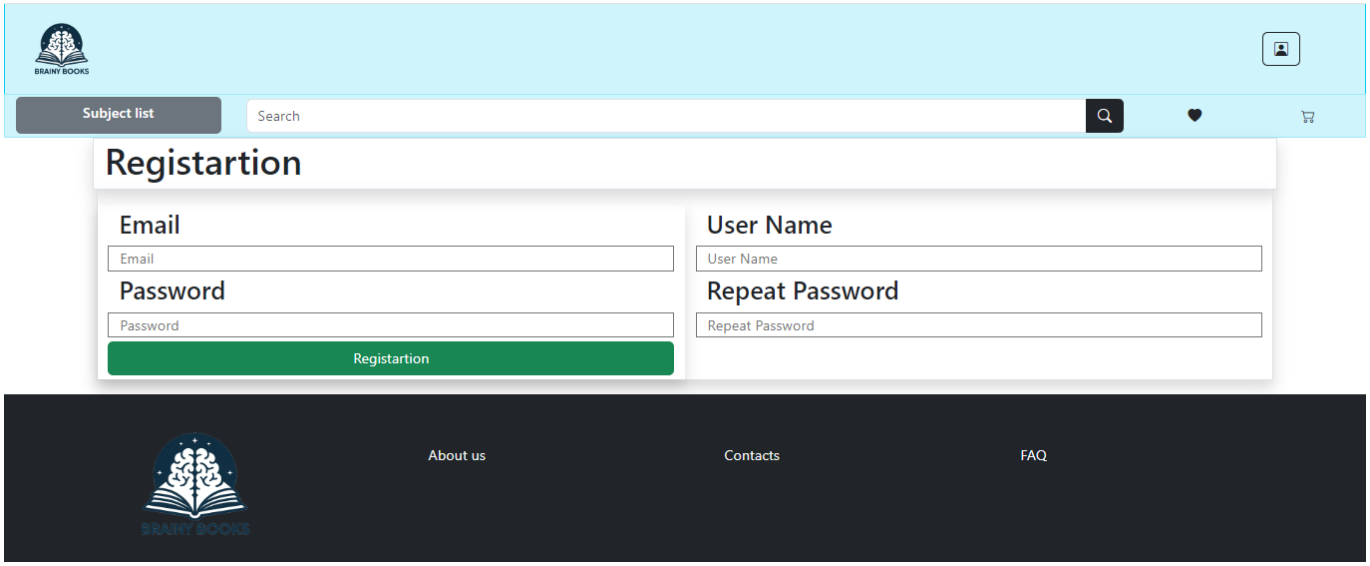

Рис. 3.11. Сторінка регістарації

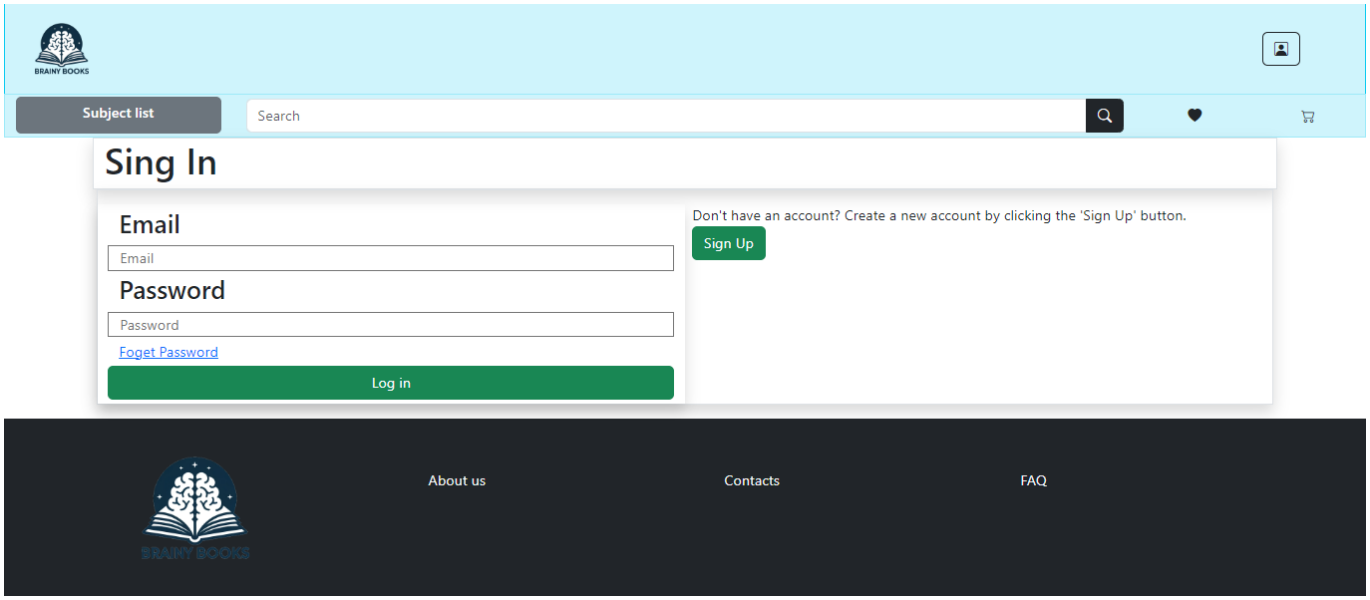

Рис. 3.12. Сторінка Авторизації

‘

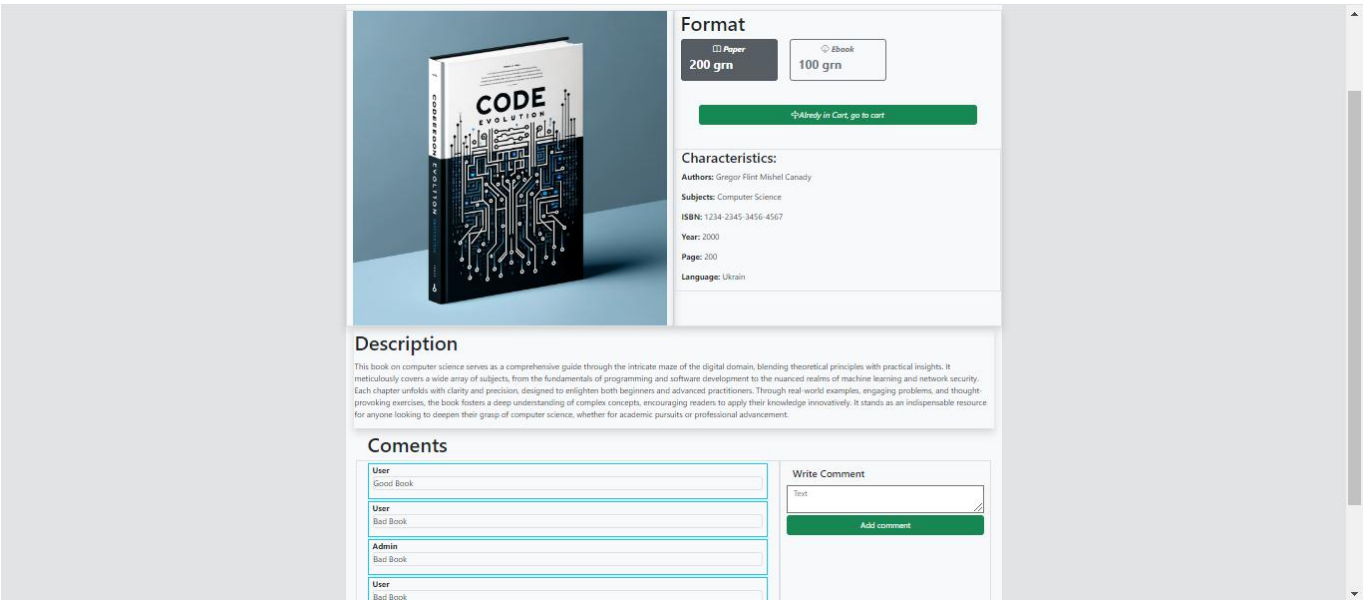

Рис. 3.13. Сторінка Книги

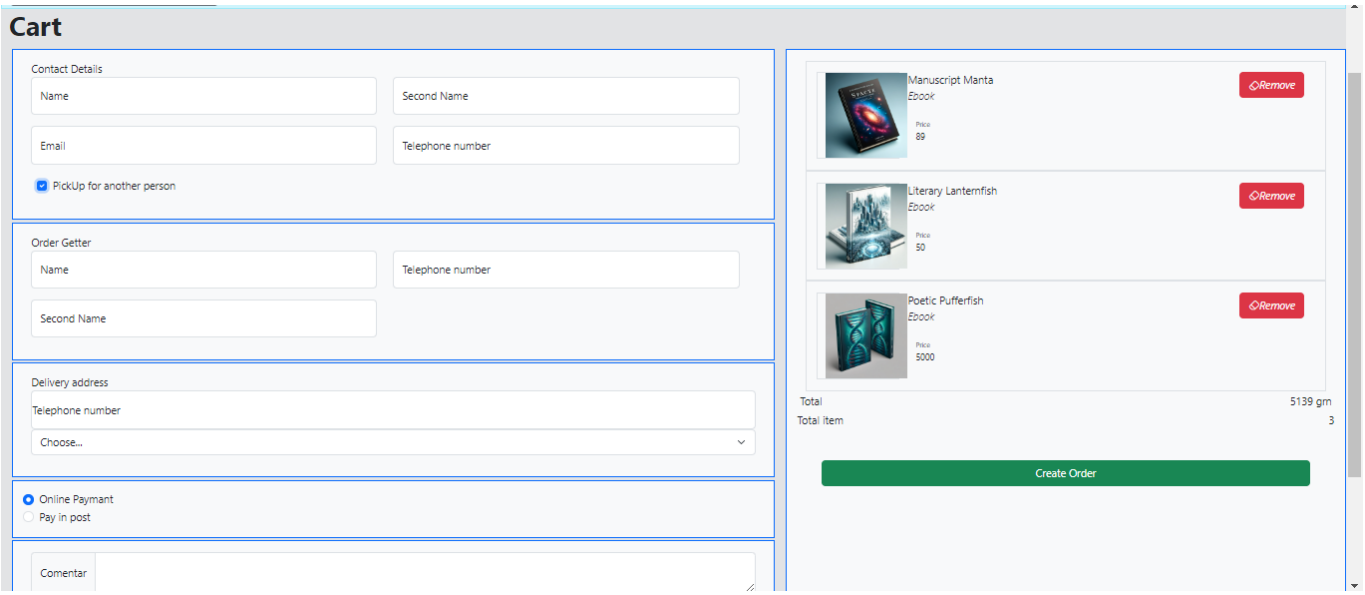

Рис. 3.14. Сторінка Книги

Висновок до 3 розділу

В рамках розділу було створено інтернет-магазин для продажу наукової літератури. Проект включає ретельно продуману структуру бази даних, ефективні веб-сервіси для обробки даних, інтуїтивно зрозумілий користувацький інтерфейс та надійні методи авторизації та оплати. Особливу увагу було приділено забезпеченню високої продуктивності, безпеки та зручності користування. Реалізація такого проекту сприятиме популяризації наукових знань, забезпечуючи легкий доступ до

‘
великої кількості наукової літератури. Окрім того, інтеграція з сучасними платіжними системами та впровадження передових технологій для роботи з даними та управління користувацьким інтерфейсом відкриває нові перспективи для розвитку електронної комерції в освітній сфері. Завдяки цьому проекту було демонстровано, як технологічні інновації можуть бути застосовані для вирішення актуальних завдань у сфері доступу до наукових ресурсів, підвищення ефективності освітнього процесу та розширення можливостей для наукових досліджень.

‘

ВИСНОВКИ

У процесі розробки моєї дипломної роботи, яка зосереджена на створенні інтернет-магазину для наукової літератури, для вирішення проблем, які стоять перед сучасною академічною спільнотою у доступі до наукових ресурсів. Використання передових технологій, таких як React, ASP.NET Core та PostgreSQL, дало можливість створити ефективний інструмент для пошуку, перегляду та придбання наукової літератури, спрямований на задоволення потреб широкого кола користувачів. Мною було розроблено інтуїтивно зрозумілий інтерфейс, що включає розширені функції пошуку та персоналізовані рекомендації, забезпечуючи користувачам легкий доступ до широкого спектру наукових матеріалів.

Значна увага приділена не лише технологічній складовій, але й аналізу потреб користувачів, що дозволило виявити та ефективно вирішити ключові виклики, які вони зазнають при пошуку наукової літератури. Результатом цієї роботи став інтернет-магазин, який не тільки спрощує доступ до наукових ресурсів, але й сприяє підвищенню ефективності наукових досліджень та освітнього процесу.

Впровадження проекту відкриває нові перспективи для розвитку академічної спільноти, забезпечуючи швидкий обмін знаннями та сприяючи глобальному доступу до наукової інформації. Завдяки цьому, дипломна робота не просто вирішує актуальну проблему, але й надає міцний фундамент для подальших досліджень та розвитку в галузі цифрової дистрибуції наукових ресурсів.

‘

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1."How to Start a Bookstore Business." Publishing State. URL: https://publishingstate.com/how-to-start-a-bookstore-business/2023/ (Дата звернення: 15.01.2024)

2."Procurement in Retail." DataWiz Blog. URL: https://datawiz.io/uk/blog/procurement-in-retail (Дата звернення:15.01.2024)

3."How to Make Money Selling Books." BookScouter Blog. URL: https://bookscouter.com/blog/how-to-make-money-selling-books/ (Дата звернення: 15.01.2024)

4."Ecommerce UX." Duda Blog. URL: https://blog.duda.co/ecommerce-ux (Дата звернення: 15.01.2024)

5."Ecommerce Product Pages." Nielsen Norman Group. URL: https://www.nngroup.com/articles/ecommerce-product-pages/ (Дата звернення: 15.01.2024)

6.JavaScript language overview. URL: https://developer.mozilla.org/enUS/docs/Web/JavaScript/Language_overview (Дата звернення 15.01.2024)

7."React Documentation (Legacy)." React. URL: https://uk.legacy.reactjs.org/ (Дата звернення: 15.01.2024)

8. "React Router Tutorial." React Router Documentation. URL: https://reactrouter.com/en/main/start/tutorial (Дата звернення: 15.01.2024)

9. "Redux Toolkit Documentation." Redux Toolkit. URL: https://redux-toolkit.js.org/ (Дата звернення: 15.01.2024)

10."ASP.NET Documentation." Microsoft .NET. URL: https://dotnet.microsoft.com/en-us/apps/aspnet (Дата звернення: 15.01.2024)

11."PostgreSQL Documentation." PostgreSQL. URL: https://www.postgresql.org/(Дата звернення: 15.01.2024)

12."Entity Framework Documentation." Microsoft Learn. URL: https://learn.microsoft.com/en-us/ef/ (Дата звернення: 15.01.2024)

‘