



slington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS6004NI Application Development

Assessment Weightage & Type

Milestone 1

Semester

2024 Spring

PROJECT TITLE: Readify

Student Name: Divya Shrestha

London Met ID: 22085527

College ID: NP01CPS230022

Milestone Due Date: 17th March

Milestone Submission Date: 17th March

Word Count: 693

I confirm that I understand my coursework must be submitted online via MySecondTeacher under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Milestone1.docx

Islington College,Nepal

Document Details

Submission ID
trn:oid::3618:91608723

Submission Date
Apr 17, 2025, 8:12 AM GMT+5:45

Download Date
Apr 17, 2025, 8:13 AM GMT+5:45

File Name
Milestone1.docx

File Size
5.0 KB

12 Pages

674 Words





4,407 Characters




11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

-  7 Not Cited or Quoted 11%
Matches with neither in-text citation nor quotation marks
-  0 Missing Quotations 0%
Matches that are still very similar to source material
-  0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
-  0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 1%  Internet sources
- 0%  Publications
- 11%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Table of Contents

1	GitHub Repository	1
2	Project Report	2
2.1	Project Title	2
2.2	UI Designs (Wireframes).....	2
2.2.1	Home Page	2
2.2.2	Books Listing Page	3
2.2.3	Book Description Page	4
2.2.4	Cart Page.....	5
2.2.5	Order Confirmation Page	6
2.2.6	Order History Page	7
2.2.7	Admin Dashboard Page.....	8
2.2.8	Booking Management Page.....	9
2.3	Database Modeling (ERD)	10
2.4	API Endpoints	11
2.5	Technology Stack.....	13
2.5.1	Framework (ASP.NET Core Web API)	13
2.5.2	External Libraries (React)	13
2.5.3	Entity Framework Core (ORM)	13
2.5.4	Database (Postgres).....	14

Table of Figures

Figure 1:GitHub Repository	1
Figure 2: Home Page	2
Figure 3: Books Listing Page	3
Figure 4: Book Description	4
Figure 5: Cart Page	5
Figure 6: Order Confirmation Page	6
Figure 7: Order History	7
Figure 8: Admin Dashboard.....	8
Figure 9: Booking Management Page	9
Figure 10: Initial Database ERD	10

Table of Tables

Table 1 : API Endpoint	12
------------------------------	----

1 GitHub Repository

Repository Link: <https://github.com/divyaSht04/Readify>

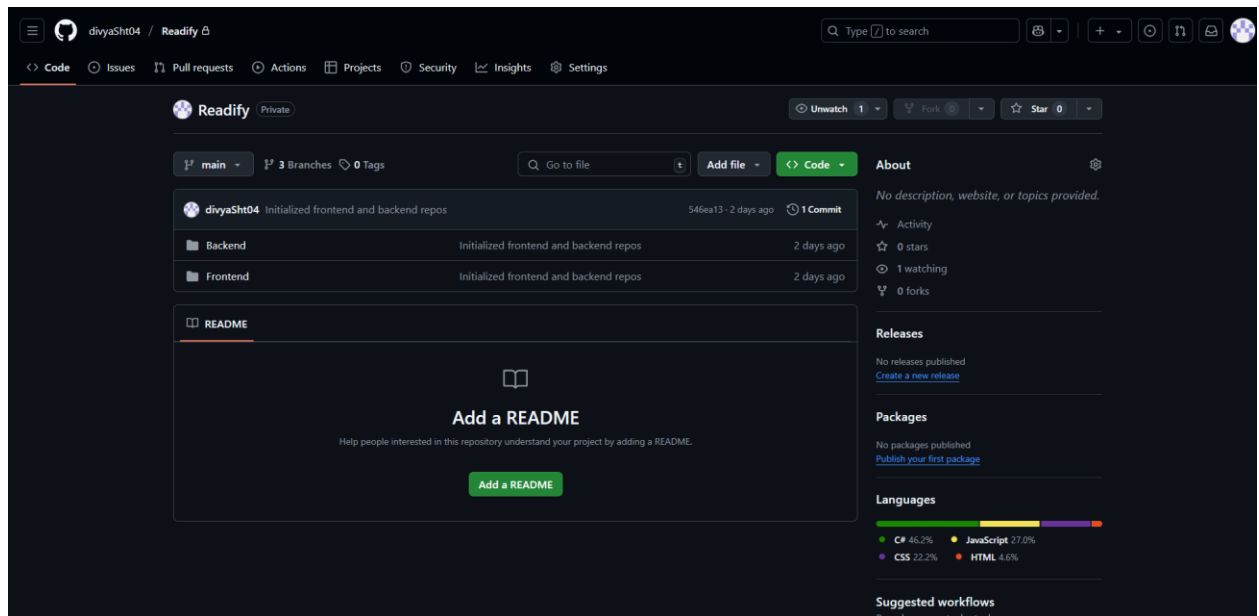


Figure 1:GitHub Repository

2 Project Report

2.1 Project Title

Readify

2.2 UI Designs (Wireframes)

2.2.1 Home Page

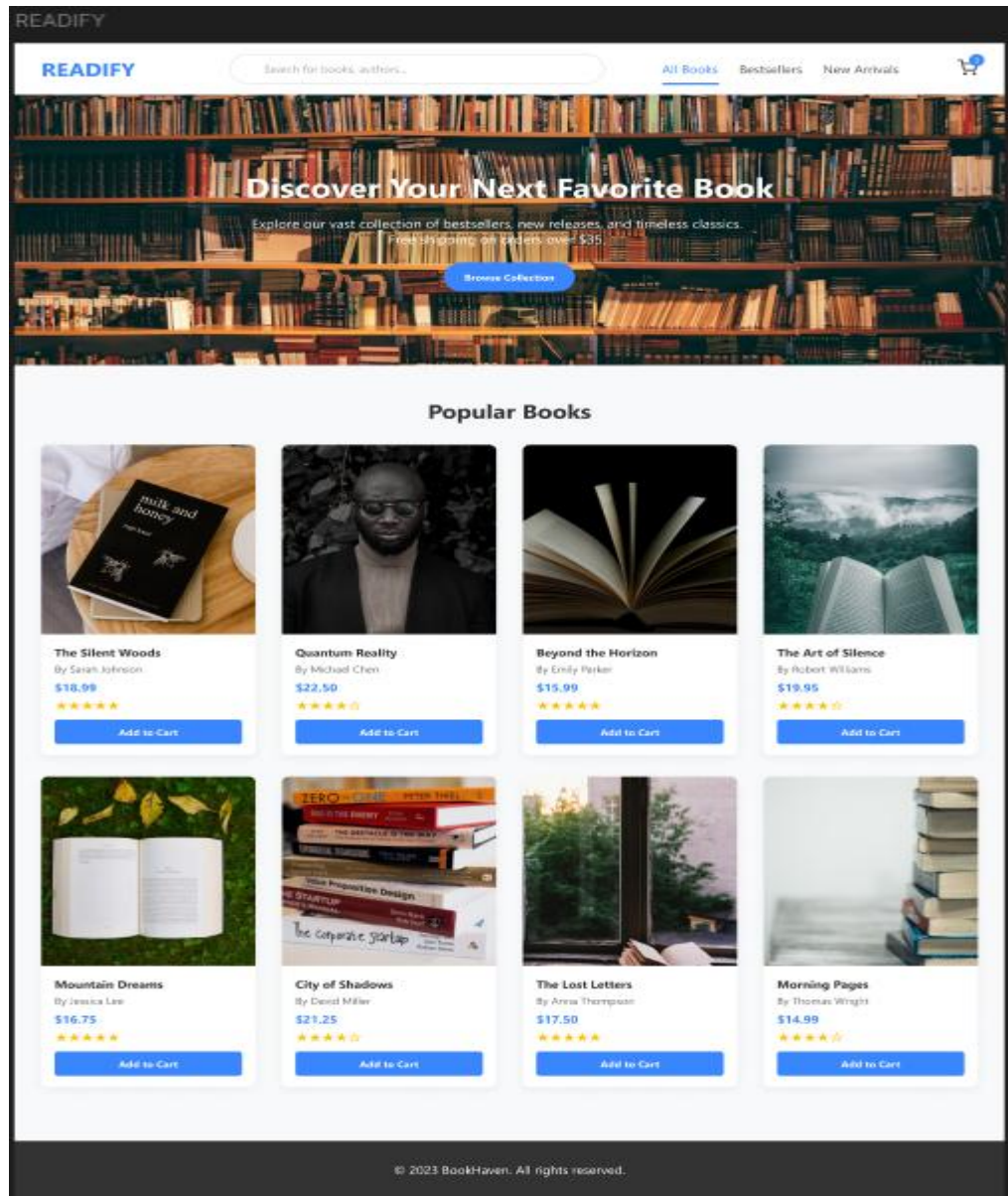


Figure 2: Home Page

2.2.2 Books Listing Page

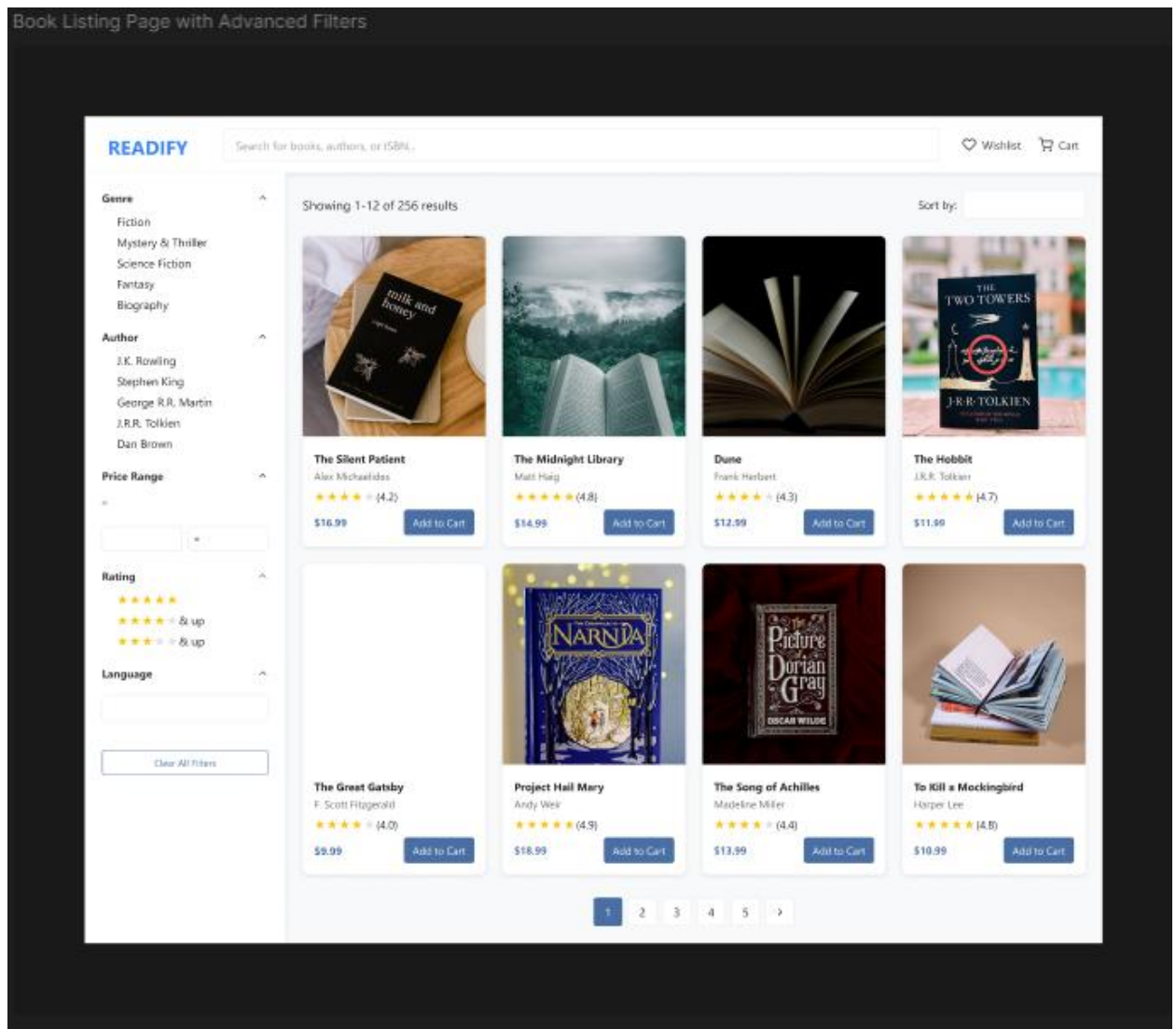



Figure 3: Books Listing Page

2.2.3 Book Description Page

Book Detail Page

READIFY



The Silent Echo

by Eleanor Winters

In this captivating novel, Eleanor Winters weaves a tale of mystery and intrigue set against the backdrop of a small coastal town. When protagonist Olivia returns to her childhood home after twenty years, she discovers long-buried secrets that threaten to upend her understanding of her family's history. Beautifully written with rich character development and atmospheric prose, The Silent Echo explores themes of memory, identity, and the power of truth.

\$24.99

Format

Paperback

Hardcover (+\$8.00)

Add to Cart

Bookmark

Customer Reviews

4.7 ★★★★★ based on 128 reviews

Michael Thompson

★★★★★

This book completely captivated me from the first page. The author's descriptive writing style made me feel like I was right there in the coastal town with Olivia. The mystery unfolded at the perfect pace, with just enough clues to keep me guessing until the very end. Highly recommend!

October 15, 2023

Sarah Johnson

★★★★☆

Eleanor Winters has done it again! The character development in this novel is outstanding, particularly the protagonist's journey. I found the middle section a bit slow, but the ending more than made up for it with several unexpected twists. A great read for fans of literary fiction with elements of mystery.

October 8, 2023

David Chen

★★★★★

I couldn't put this book down! The way the author explores the themes of memory and identity resonated deeply with me. The descriptions of the coastal town were so vivid I could almost smell the salt air. This would make an excellent book club selection as there's so much to discuss.

September 30, 2023

Figure 4: Book Description

2.2.4 Cart Page

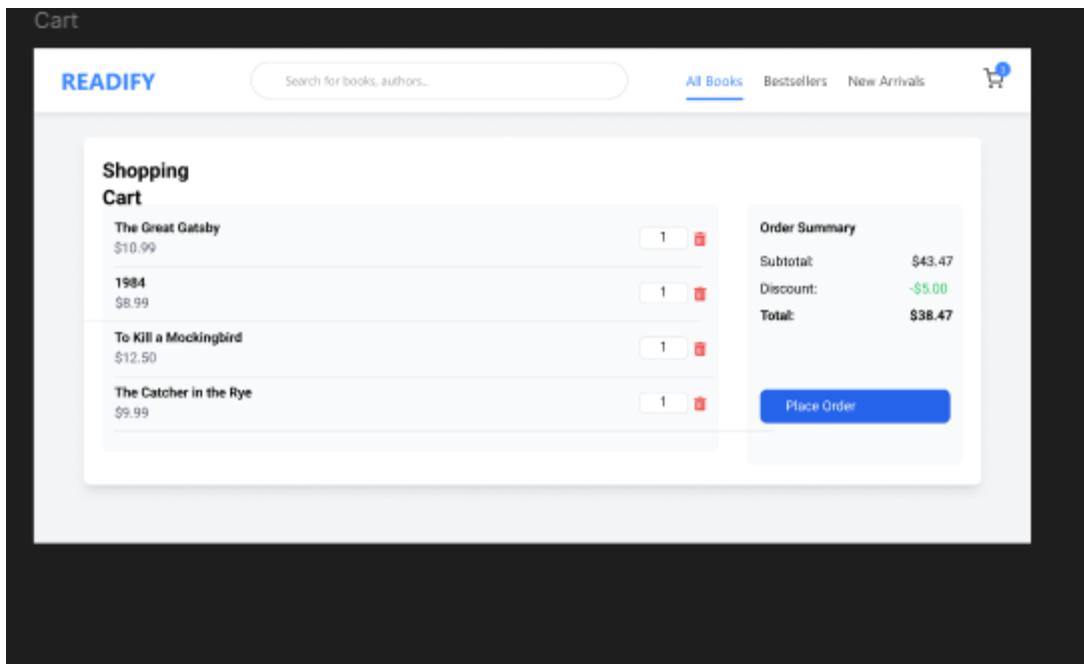


Figure 5: Cart Page

2.2.5 Order Confirmation Page

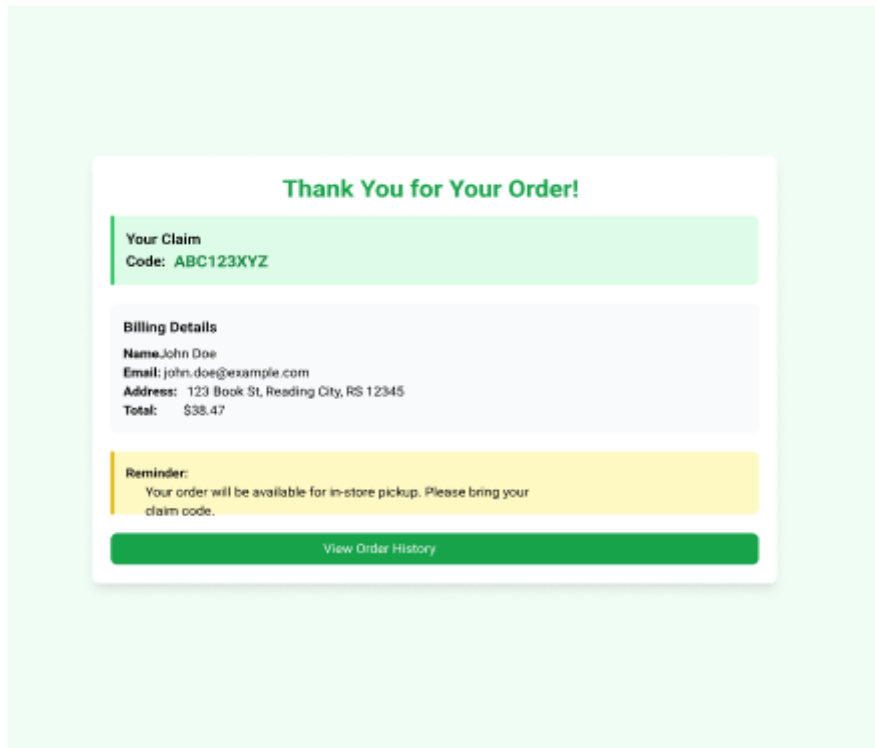


Figure 6: Order Confirmation Page

2.2.6 Order History Page

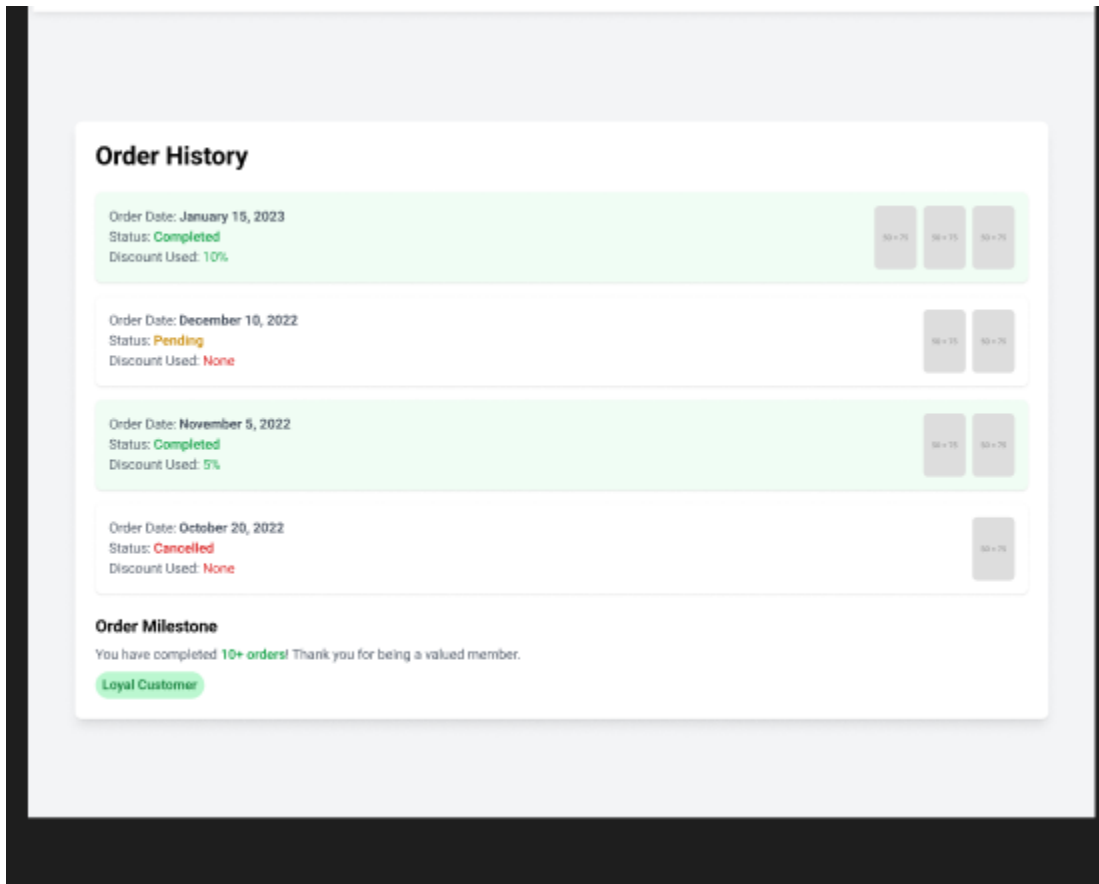


Figure 7: Order History

2.2.7 Admin Dashboard Page

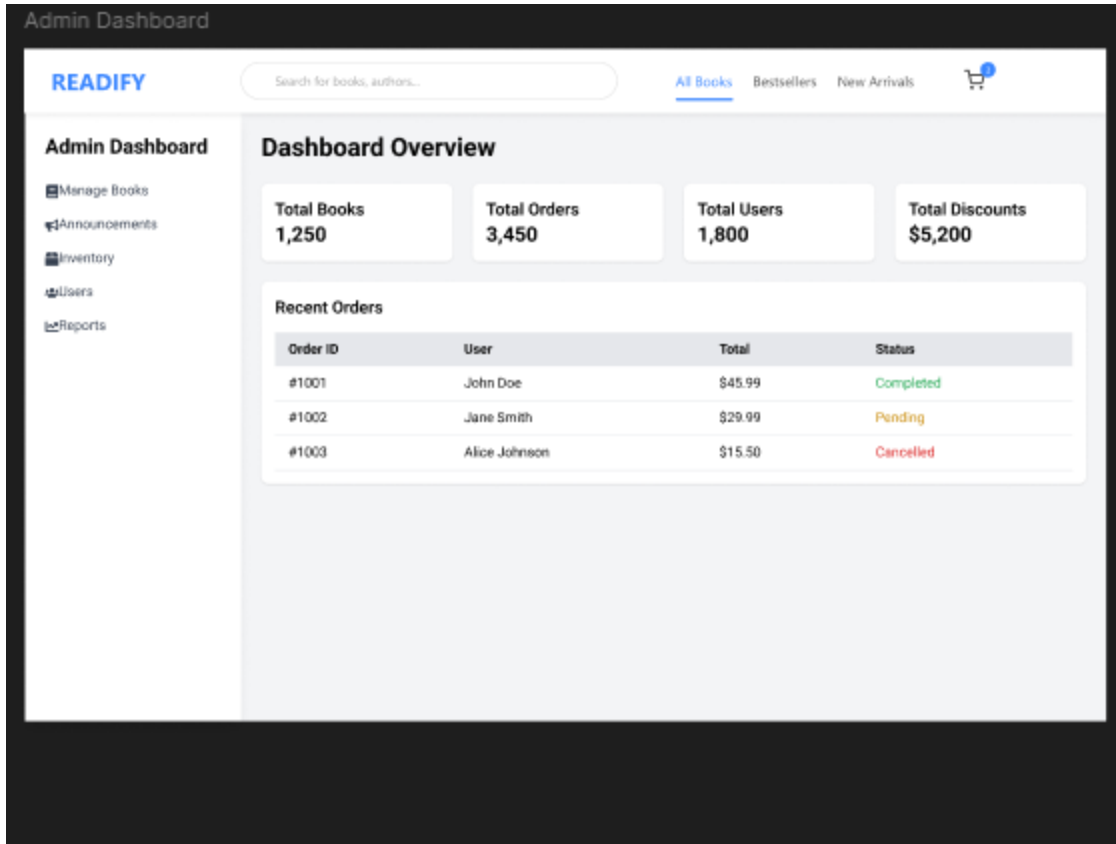


Figure 8: Admin Dashboard

2.2.8 Booking Management Page

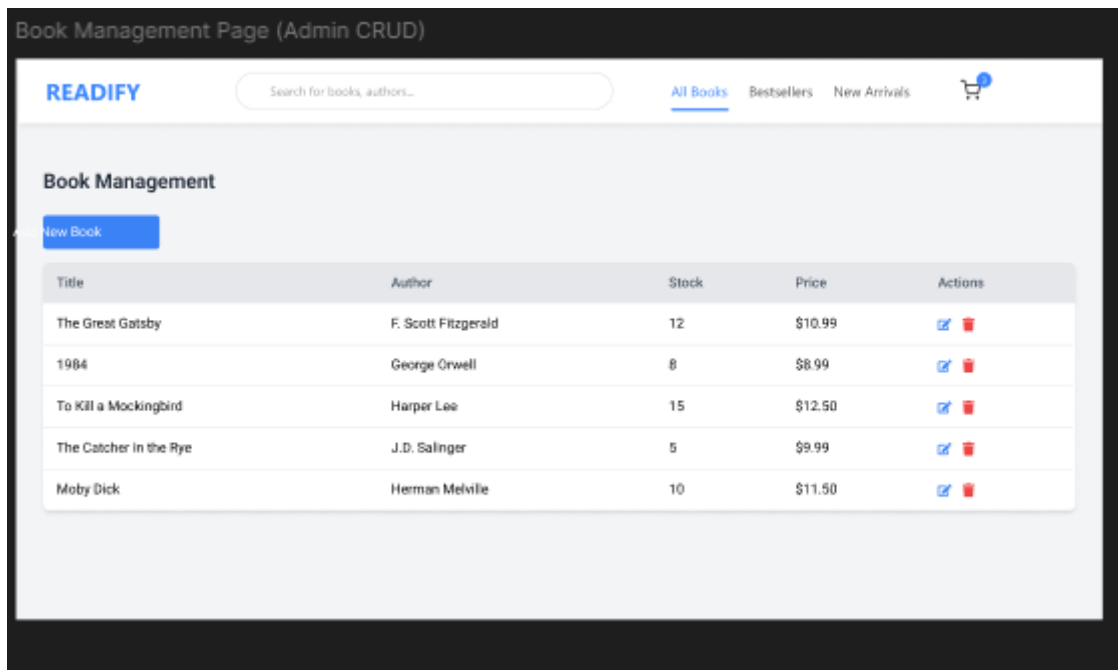


Figure 9: Booking Management Page

2.3 Database Modeling (ERD)

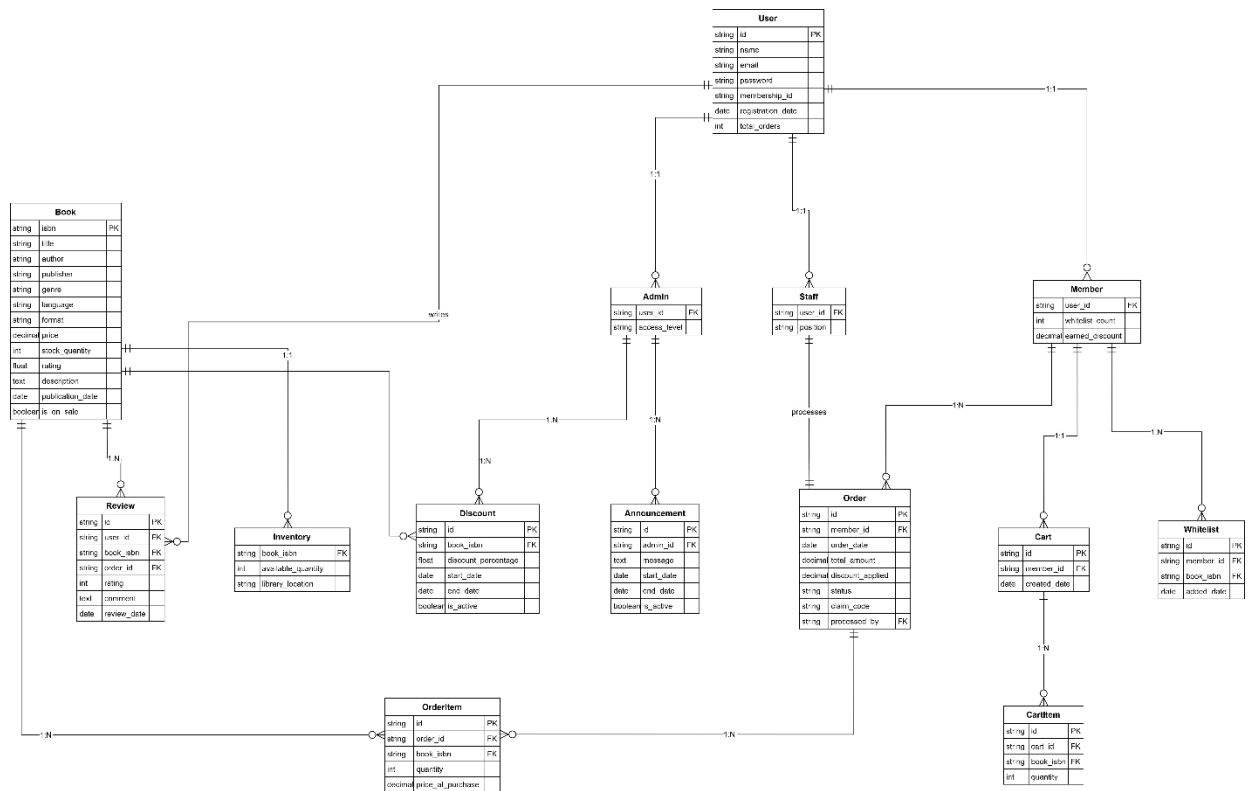


Figure 10: Initial Database ERD

2.4 API Endpoints

Endpoint	HTTP Method	Description
/books	GET	Retrieve a list of books.
/books/{bookId}	GET	Retrieve detailed information about a specific book by its ID.
/users/register	POST	Register a new user (member) with the system, returning user details and possibly a token.
/users/{userId}/bookmarks/{bookId}	POST	Add a book to the user's bookmarks (whitelist). Requires user authentication.
/users/{userId}/bookmarks	GET	Retrieve the list of books bookmarked by the user. Requires user authentication.
/users/{userId}/cart/{bookId}	POST	Add a book to the user's shopping cart. Requires user authentication.
/users/{userId}/cart	GET	Retrieve the contents of the user's shopping cart. Requires user authentication
/users/{userId}/orders	POST	Place a new order. Requires user authentication and cart contents. Automatically generates a claim code and sends an email with the bill.
/users/{userId}/orders	GET	Retrieve a list of the user's past orders. Requires user authentication.
/users/{userId}/orders/{orderId}/cancel	POST	Cancel an existing order. Requires user authentication and ownership of the order.

/users/{userId}/reviews/{bookId}	POST	Submit a review (rating and comment) for a book. Requires user authentication and proof of purchase.
/staff/orders/fulfill/{claimCode}	POST	Fulfill an order using the claim code. Requires staff authentication.
/admin/books	POST	Add a new book to the catalog. Requires admin authentication.
/admin/books	GET	Retrieve the list of all books in the catalog. Requires admin authentication.
/admin/books/{bookId}	PUT	Update the details of a specific book. Requires admin authentication.
/admin/books/{bookId}	DELETE	Remove a book from the catalog. Requires admin authentication.
/admin/discounts	POST	Set a new discount on time with an optional "On Sale" flag. Requires admin authentication.
/admin/discounts	GET	Retrieve the list of current discounts. Requires admin authentication.
/admin/announcements	POST	Create a new timed banner announcement (e.g., for deals or new arrivals). Requires admin authentication.
/admin/announcements	GET	Retrieve the list of current announcements. Requires admin authentication.

Table 1 : API Endpoint

2.5 Technology Stack

2.5.1 Framework (ASP.NET Core Web API)

ASP.NET Web API serves this work because it enables RESTful service development, which supports connection with various clients including applications through different platforms. Because of its lightweight construction and support for high-performance operations and modern web standards ASP.NET Web API provides optimal conditions for scalable backend system development. This framework serves as the primary selection for API development because it includes built-in functionality for routing alongside serialization and authentication systems.

2.5.2 External Libraries (React)

The project development will use React because it enables developers to create dynamic web interfaces that specifically work well with single-page applications. The component-based structure of React enables scalability by promoting code reuse and better management capabilities. A virtual DOM provides super-fast rendering capabilities in addition to a modern smooth user experience through its implementation. The React ecosystem offers both community backing and various tools to enable developers to create front-end interfaces that are both modernized and highly interactive and easily maintainable.

2.5.3 Entity Framework Core (ORM)

The project utilizes Entity Framework Core as its Object-Relational Mapper (ORM) because the tool provides functionality to link object-oriented C# elements to relational database structures. The database interactions become straightforward with EF Core because developers require only clean and readable code without needing to handle complicated SQL statements unless they decide to explicitly use SQL. As an ORM it enables LINQ queries, implements automatic migration features and functions with .NET to boost development velocity and deliver robust data operations.

2.5.4 Database (Postgres)

The project selects PostgreSQL as its database choice because it stands among the top open-source relational database systems. PostgreSQL delivers exceptional reliability combined with rock-solid robustness for handling challenging queries and vast datasets. PostgreSQL includes advanced capabilities in JSON processing together with full-text search functionality and customized functions which make it adaptable to work with both classic and contemporary applications.