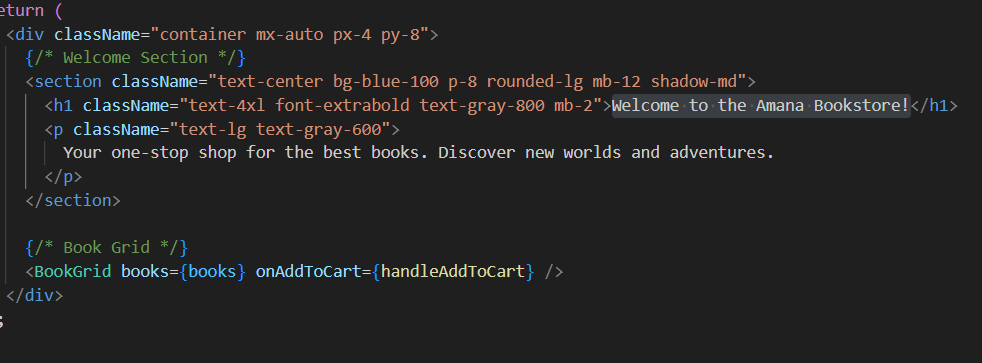
1. **Main Welcome:** Which file contains the main welcome message and hero section that users see when they first visit the website?

**Answer: page.tsx**



1. **Data:** Where is all the book data stored in this application? What file would you look at to see the list of available books?

**Answer: in data folder**

1. Find the navigation bar at the top of the website. Which file controls what appears in this navigation?

**Answer: in components floder , Navbar.tsx control the navbar**

1. Looking inside of the components folder, what do you think each component does in the context of this application and which files reference each component?

 **BookCard.tsx** → Displays a single book card (used inside **BookGrid**).

 **BookGrid.tsx** → Renders a grid layout of books (used in **page.tsx**).

 **BookListItem.tsx** → Displays a single book in list format (used inside **BookGrid**).

 **CartItem.tsx** → Shows a single cart item (used in **cart/page.tsx**).

 **Navbar.tsx** → The top navigation bar (used in **layout.tsx**).

 **Pagination.tsx** → Provides pagination controls (used inside **BookGrid**).

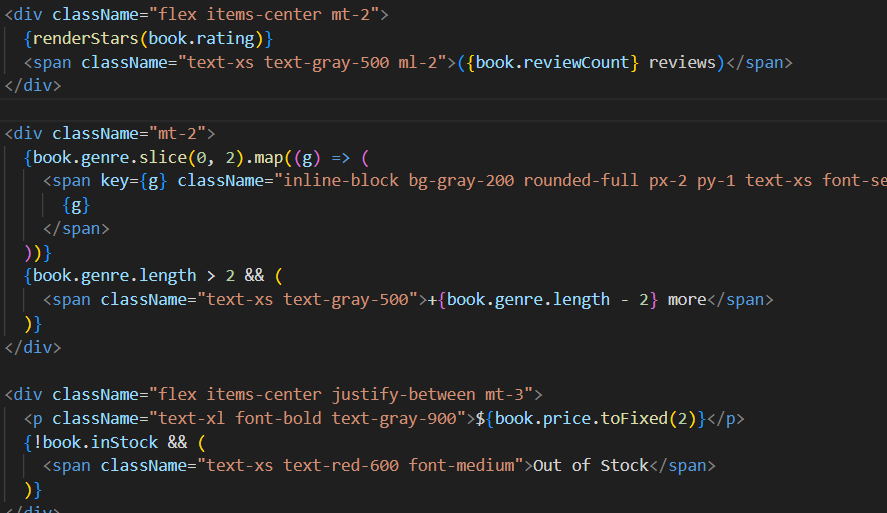
1. Which file controls how Books get displayed in the All Books section? Which file controls how Books get displayed in the Featured Books Section?

 **All Books section** → uses **BookListItem.tsx** to display each book.

 **Featured Books section** → uses **BookCard.tsx** to display each featured book.

1. How are the properties (like price, review count, and genre) related to the books being retrieved from the data? Show the snippet of code that controls the display of these properties in the Featured Book section.

**BookCard.tsx** displays each featured book using properties from the book object (price, review count, genre).



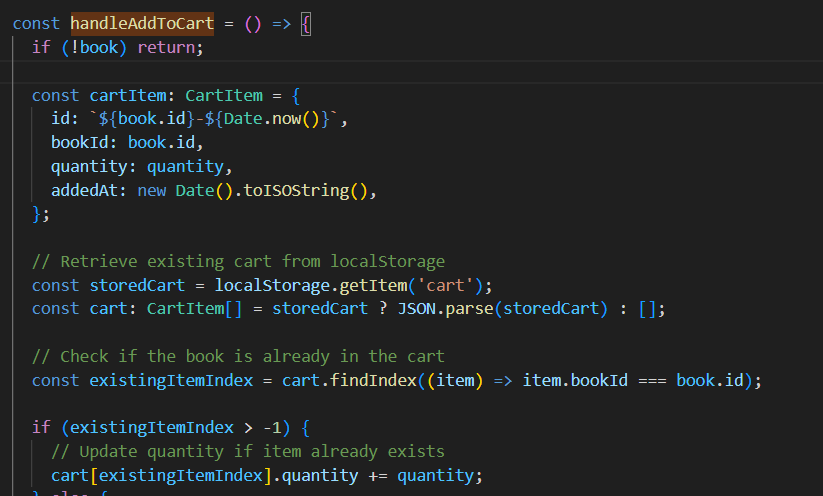
1. How does this application generate unique URLs for each individual book (like /book/1, /book/2, etc.)? Which folder structure makes this possible?

 **Dynamic routing**: The [id] folder creates unique URLs for books (/book/1, /book/2).

 **page.tsx** inside [id] renders the individual book page based on its ID.

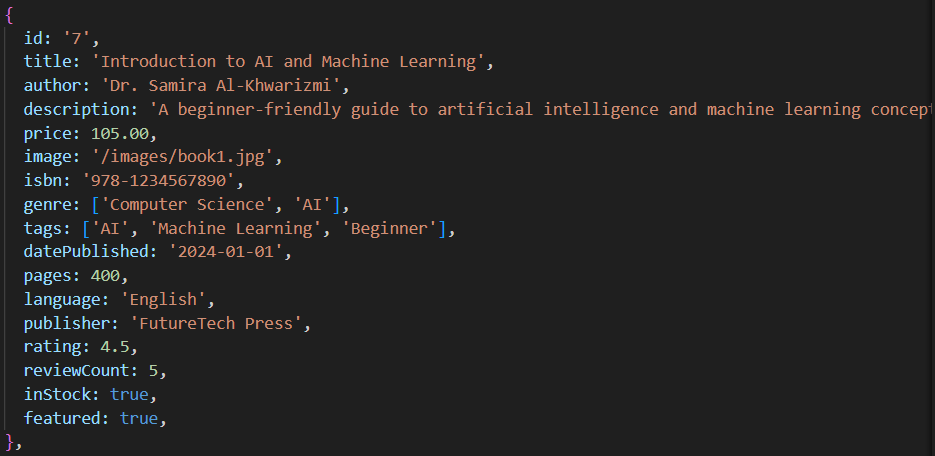
1. Where in the codebase is the logic for keeping track of the items in the cart? Briefly, how does it work? Show screenshots of the codebase.

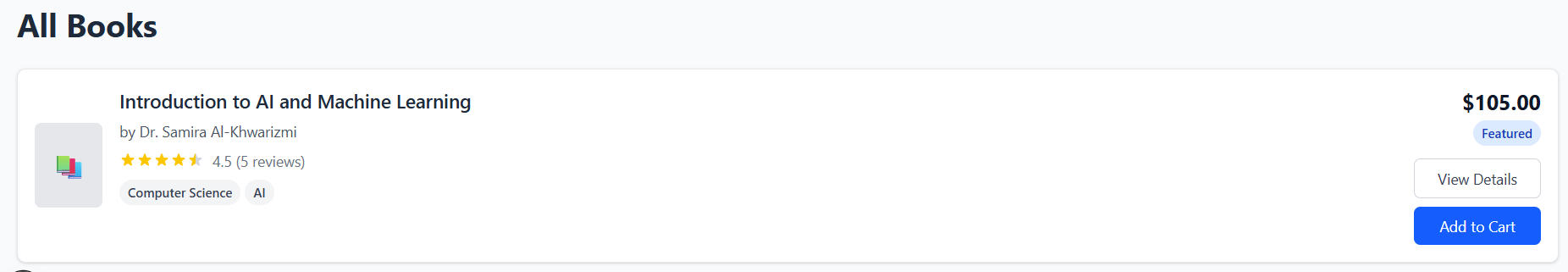
**Answer:** A cart item is created for the selected book, the cart is retrieved from localStorage, and if the book exists its quantity is increased, otherwise it’s added as new. Finally, the updated cart is saved back to localStorage.



1. How might we add an additional book to the collection? For this question, add a book yourself and include a screenshot showing that it worked.

**Answer:**  by adding to book file in data folder



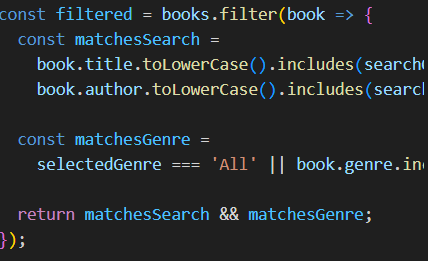


1. How does the code “filter” the books that are considered the “Featured Books”? How does it “filter” books based on category? Show screenshots or code snippets from the code base.

**Answer:** The code filters “Featured Books” and by category in BookGrid.tsx using these snippets:

****

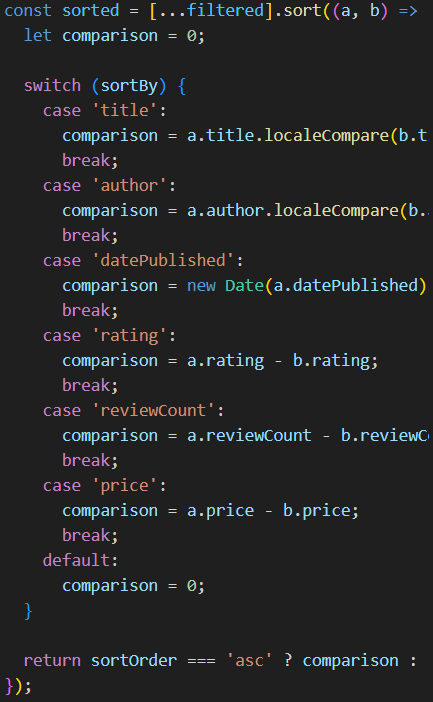
This line creates a list of books where book.featured is true.

****

This filters books by the selected genre (category) and search query.

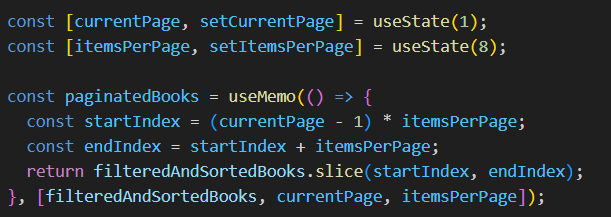
These code snippets are from BookGrid.tsx.

1. How does the code “sort” books based on the user selection? Show screenshots or code snippets from the code base



1. How does the code manage “pagination”? Where can we change the number of items being displayed per page? Show screenshots or code snippets from the code base.?

Pagination is managed in BookGrid.tsx using these lines:



You can change the number of items displayed per page by updating the value in:



1. How are the styling of the page elements being defined? How might I change the blue colors used on the buttons? How might I change the text height of the hero section?

The styling of page elements is defined using Tailwind CSS utility classes (e.g., bg-blue-600, text-4xl, py-8) directly in the JSX code.

To change the blue colors on buttons:  
Find classes like bg-blue-600, hover:bg-blue-700, or focus:ring-blue-500 in your components (e.g., BookCard.tsx, BookGrid.tsx). Replace them with another Tailwind color (e.g., bg-green-600) or customize your Tailwind config for new color values.

To change the text height of the hero section:  
In page.tsx, look for the hero section’s heading, which uses classes like text-4xl (font size) and possibly leading-tight (line height). Adjust these classes to change the text height, e.g., use leading-loose for more spacing or text-5xl for larger text.

1. What is the index.ts file found in the types folder? How does it get used in the applicaton?

The index.ts file in the types folder defines TypeScript types and interfaces (such as Book, CartItem, Review) used throughout your application for type safety and code completion.

It gets used by importing these types into other files, for example:



This allows components and data files to use consistent type definitions for books, cart items, and reviews, helping prevent bugs and making the code easier to maintain.

1. What is the packages.json file? Why is it significant?

The package.json file is a configuration file for Node.js projects. It lists your project's dependencies (libraries and frameworks), scripts, metadata, and configuration.

**Significance:**

* It defines which packages your project needs to run and develop.
* It allows others to install all dependencies with a single command (npm install).
* It manages scripts for building, testing, and running your app.
* It helps ensure consistency across different environments and developers.

1. What is held inside of the public folder? Why do we have a separate public and app folder?

**Answer:**

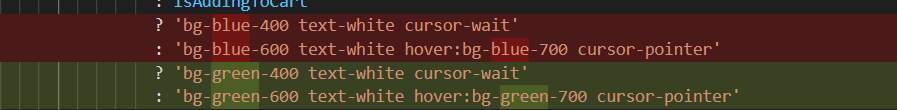
The public folder holds static files like images, icons, and other assets that are directly accessible by the browser (e.g., /images/book1.jpg). These files are not processed by the build system and can be referenced by URL.

We have a separate public and app folder because:

* **public** is for static assets served as-is.
* **app** contains your application’s source code (components, pages, logic) that gets compiled and bundled by the framework.

This separation keeps static resources organized and ensures efficient asset delivery.

changes



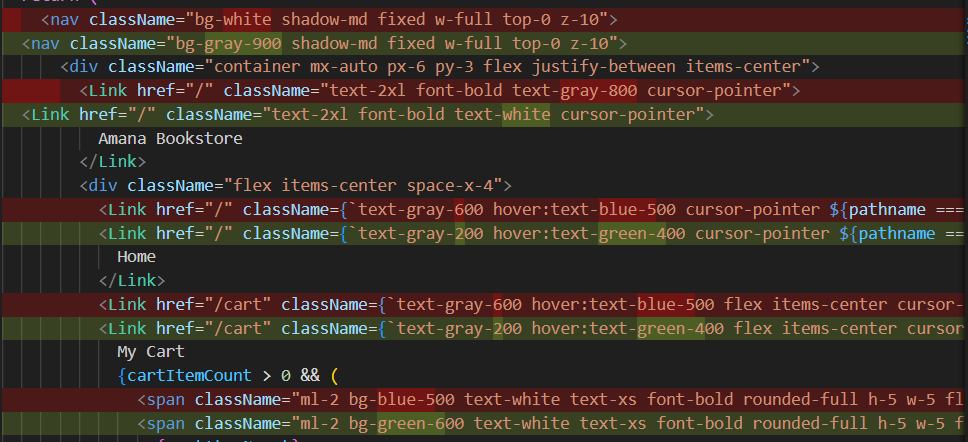
* 1. The primary button color in BookCard.tsx has been changed from blue to green



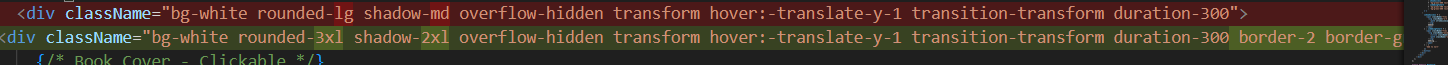
* 1. This increases the heading size and line height for a more prominent look. By page.txs



* 1. new tagline has been added under the main hero



* 1. The navbar background color has been changed to dark gray (bg-gray-900)



* 1. The book cards now have a larger border radius, a stronger shadow, and a subtle green border for a more modern look.