BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC THĂNG LONG**

****

**BÀI TẬP LỚN**

Môn: Công nghệ Web

**Đề tài 1: Xây dựng website bán sách**

**SINH VIÊN: A41842 NGUYỄN VĂN THANH**

**A42908 ĐÀM THỊ LAN ANH**

**GIẢNG VIÊN:** **NGUYỄN HÙNG CƯỜNG**

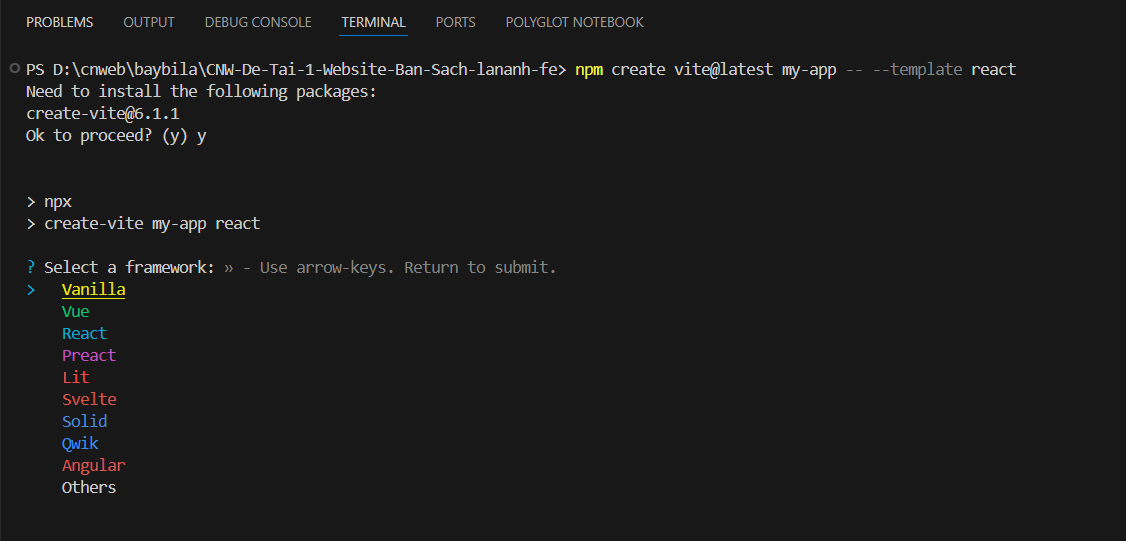
1. **Frontend**
2. **Khởi tạo ứng dụng**

## Khởi tạo dự án react js

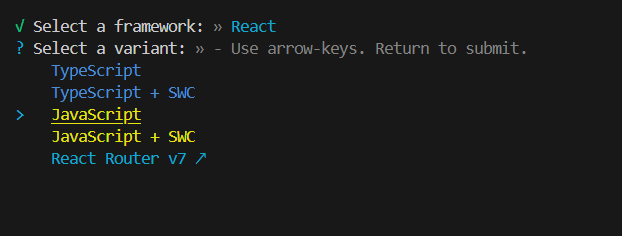
Cài đặt Vite: Vite là một công cụ phát triển hiện đại cho các dự án web, nổi bật với tốc độ khởi động nhanh và khả năng phản hồi tức thì. Để khởi tạo một dự án React JS với Vite, chúng ta cần sử dụng lệnh sau:

npm create vite@latest my-app -- --template react

Sau khi chạy lệnh trên, Vite sẽ tải về các gói cần thiết và bắt đầu quá trình tạo dự án.

****

Chọn framework và biến thể: Trong quá trình cài đặt, Vite sẽ yêu cầu người dùng lựa chọn framework và biến thể muốn sử dụng. Chúng ta sẽ chọn React làm framework, như hình dưới đây. Sau đó chọn JavaScript để phát triển ứng dụng React với JavaScript tiêu chuẩn.



Sau khi quá trình tạo dự án hoàn tất, chúng ta cần di chuyển vào thư mục dự án và cài đặt các phụ thuộc bằng lệnh:

cd my-app

npm install

Chạy Ứng Dụng: Để khởi động server phát triển và chạy ứng dụng React, chúng ta sử dụng lệnh:

npm run dev

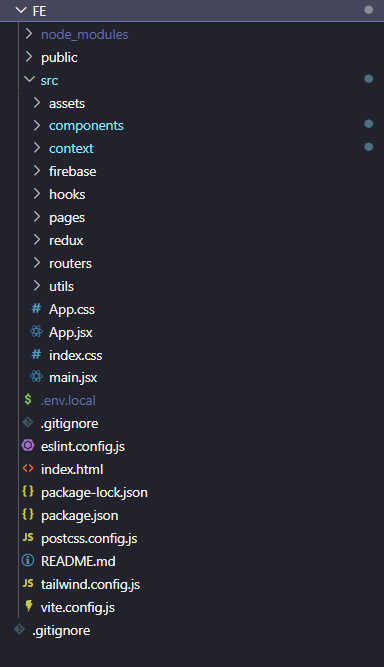
Server sẽ được khởi động và ứng dụng sẽ mở trong trình duyệt web với địa chỉ mặc định là <http://localhost:3000>.

## Thư viện



* Reduxjs / toolkit:thư viện quản lí state
* React-hook-form: thư viện quản lí form
* React-icons: thư viện cung cấp icons
* React-redux: thư viện hỗ trợ cho react làm việc với redux
* React-router, react-router-dom: thư viện cấu hình định tuyến cho react
* Sweetalert2: thư viện hiển thị popup
* Swiper: thư viện tạo slider
* Autoprefixer, postcss, tailwindcss: thư viện css

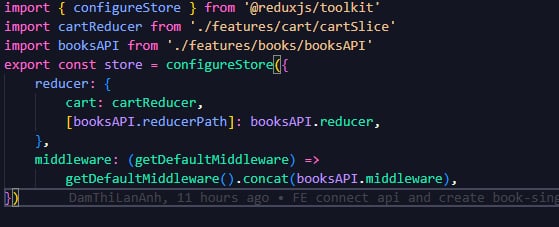
## Cấu trúc thư mục dự án



|  |  |
| --- | --- |
| Tên folder | Chức năng |
| Assets | Chứa tài nguyên ảnh |
| Components | Chứa các thành phần tái sử dụng |
| Hooks | Chứa các custom hook |
| Pages | Các trang của trang web |
| Redux | Cấu hình quản lí state |
| Router | Cấu hình định tuyến cho trang web |
| Utils | Chứa các tiện ích |
| Firebase | Chứa thư viện và framework |
| Context | Chứa các tệp liên quan đến việc quản lý **ngữ cảnh** hoặc **trạng thái toàn cục** |

## II.Xây dựng các hàm gọi api và xử lí trạng thái

* 1. **Tạo và cấu hình store**



Sử dụng hàm configureStore của redux để tạo store

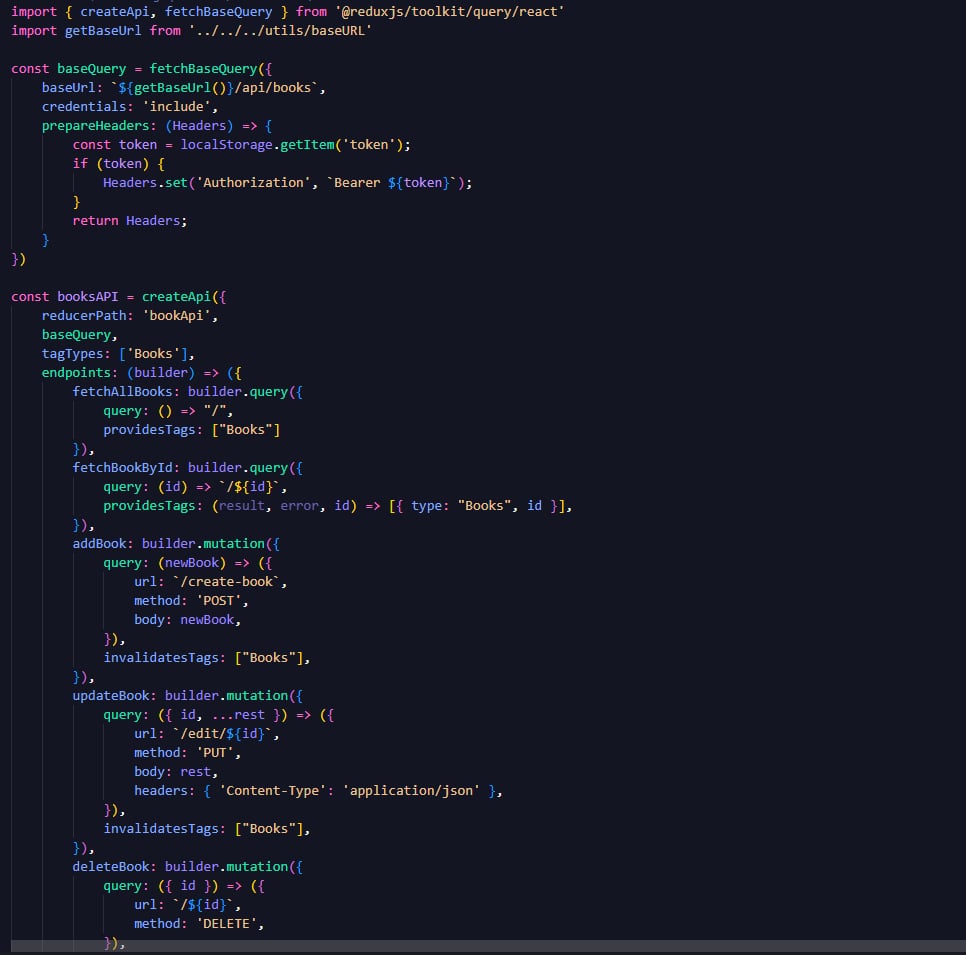
Store bao gồm các Reducer và Middleware

### 2.1.1.Tạo slice

Cấu trúc các slice chia theo chức năng như:

-sản phẩm

-giỏ hàng

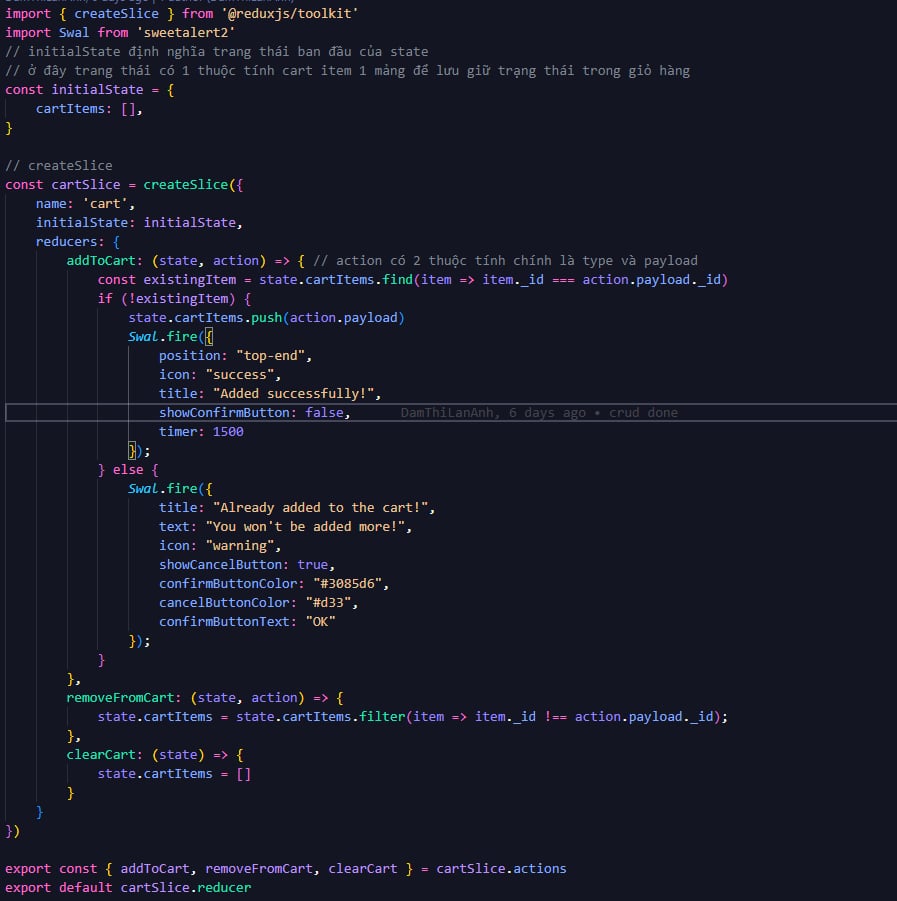


### 2.1.2. Tạo hàm gọi api cho sản phẩm:

-Tạo baseQuery gồm base endpoint của api, và 1 số config cho header

-Cấu hình hàm gọi api thông qua rtk query

### 2.1.3. Tạo hàm quản lí state cho giỏ hàng



Initial State của danh sách các sản phẩm trong giỏ hàng là 1 mảng dữ liệu

Cấu hình slice cho giỏ hàng bao gồm các reducers tượng trưng cho các hành động như: thêm vào giỏ hàng, xóa khỏi giỏ hàng,…

* 1. **Các hàm gọi Api**
* Màn hình trang chủ User

A screenshot of a computer program

Description automatically generated

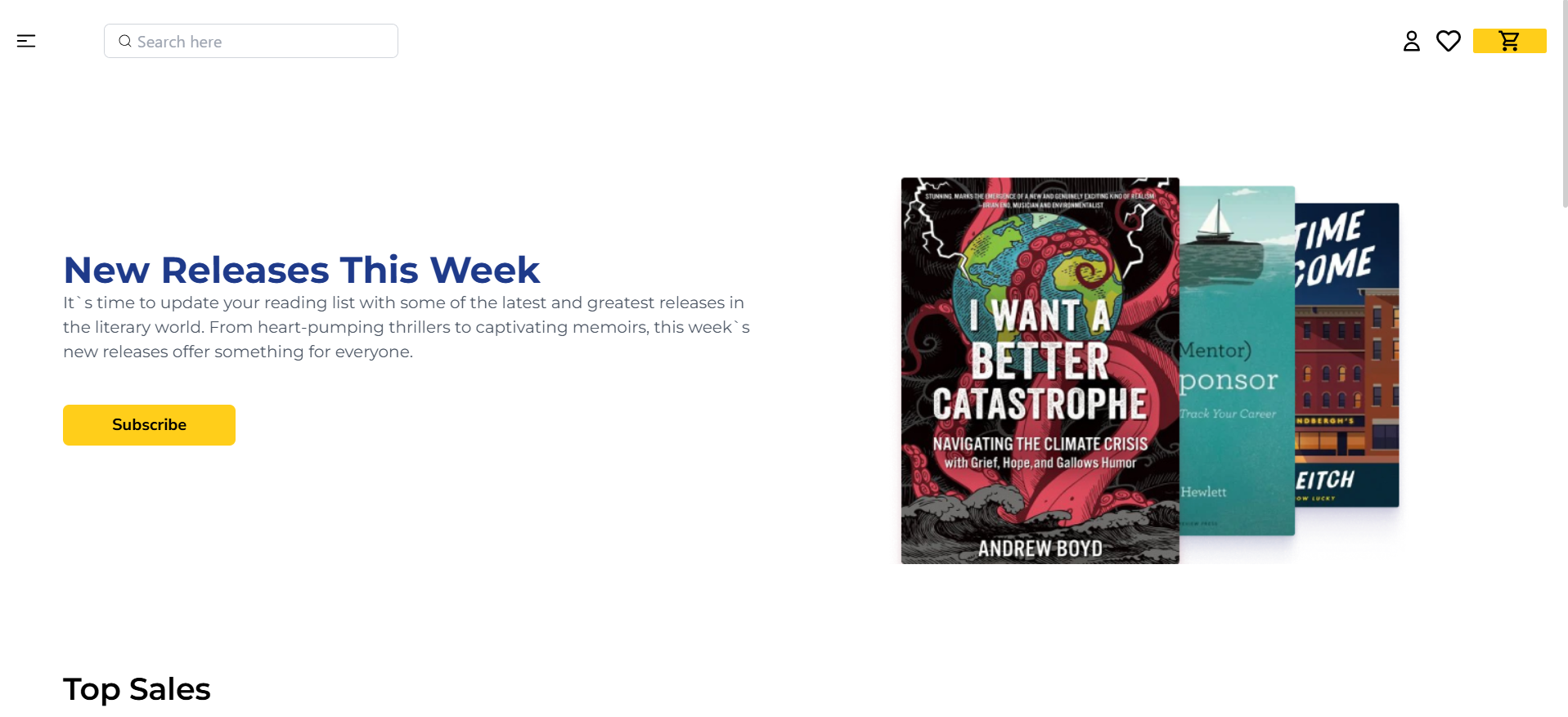
* Màn hình detail của sản phẩm

A computer screen shot of a program code

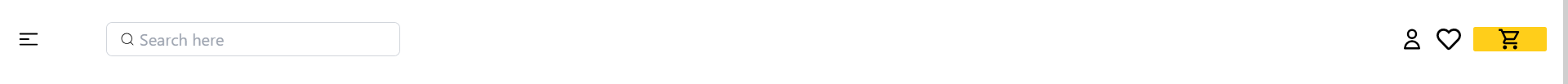
Description automatically generated

## III. Xây dựng FrontEnd các trang web

## 3.1. Màn hình home:

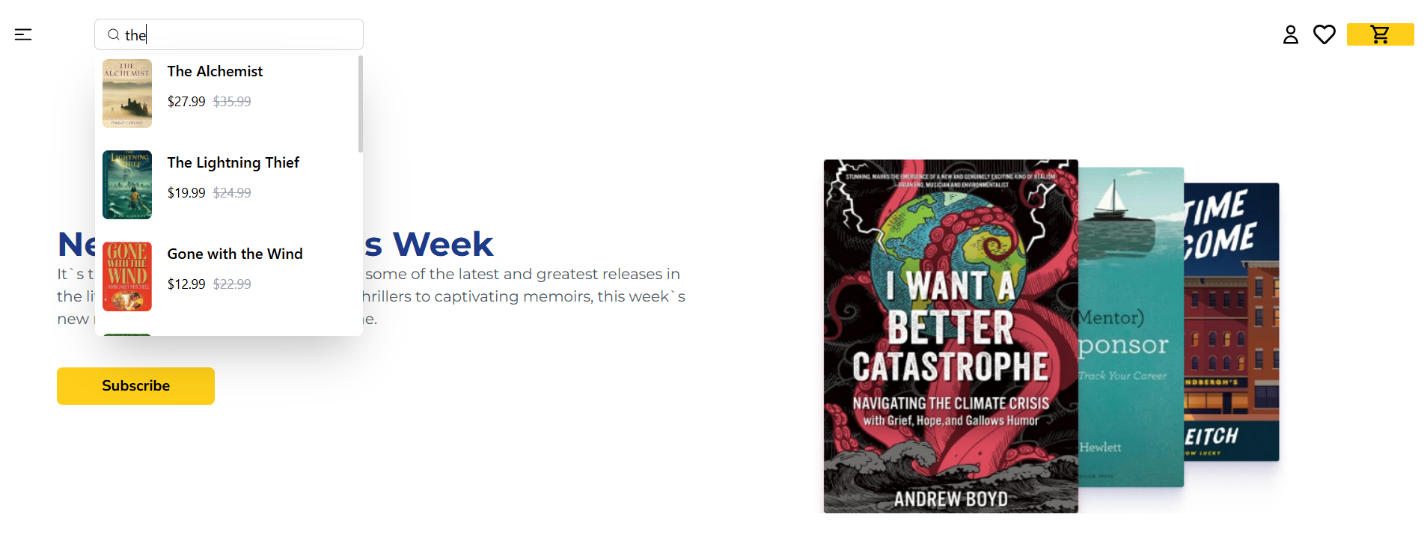


## 3.2. Header:

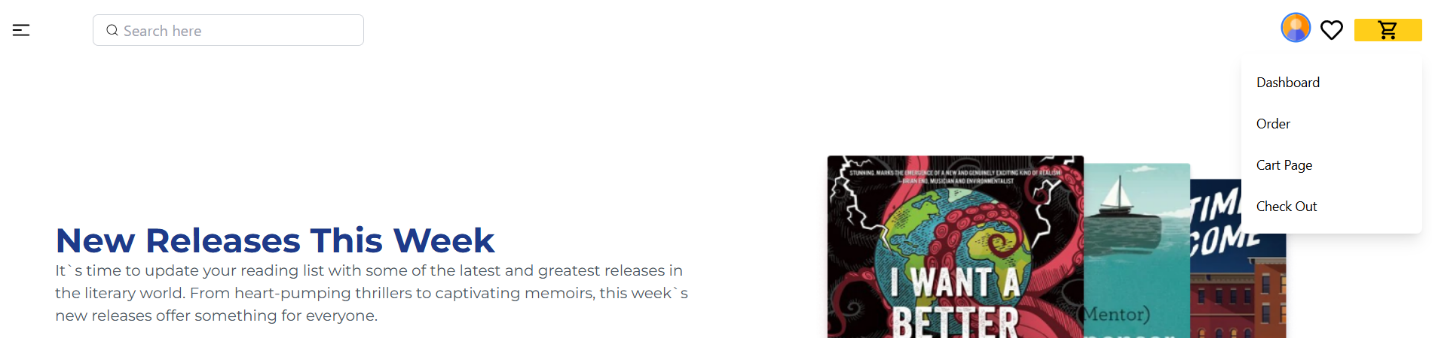


-Icon đầu trang: để quay về trang home

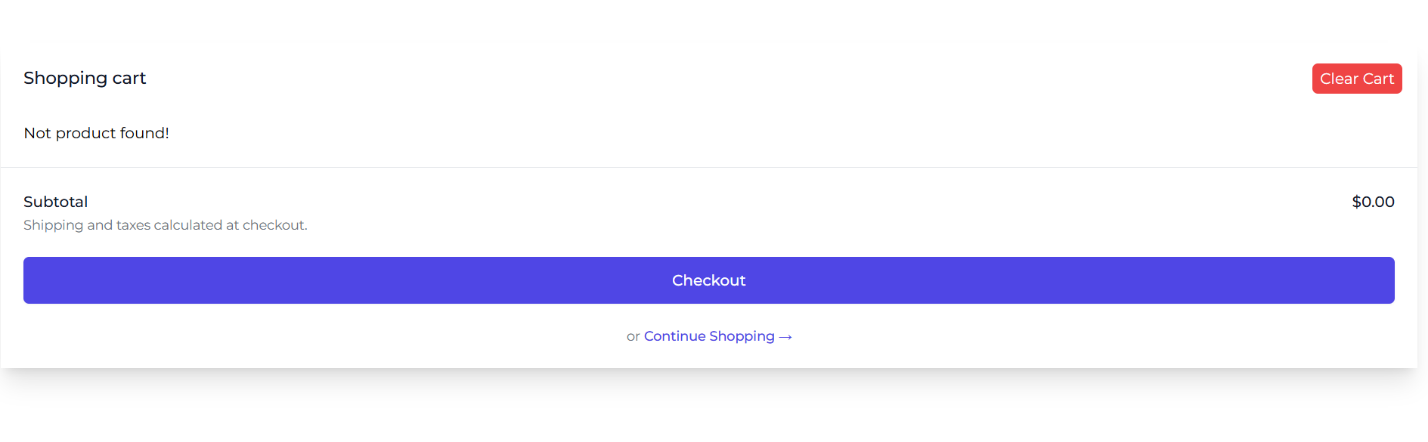
-Input search : dùng để search sản phẩm theo tên, 1 giây sau khi người dùng dừng gõ sẽ bắt đầu search sản phẩm



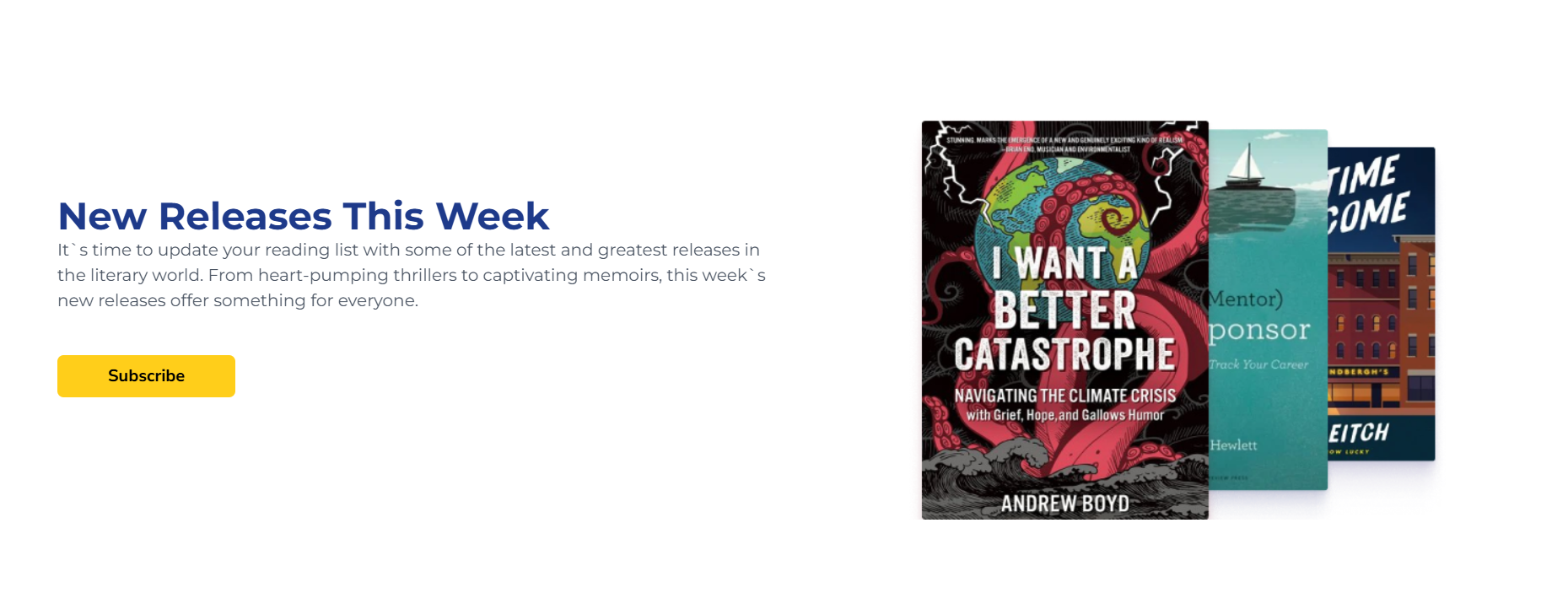
-Icon User: hiển thị dropdown login, logout. Sau khi đăng nhập icon sẽ đổi màu và có thêm các chức năng khác



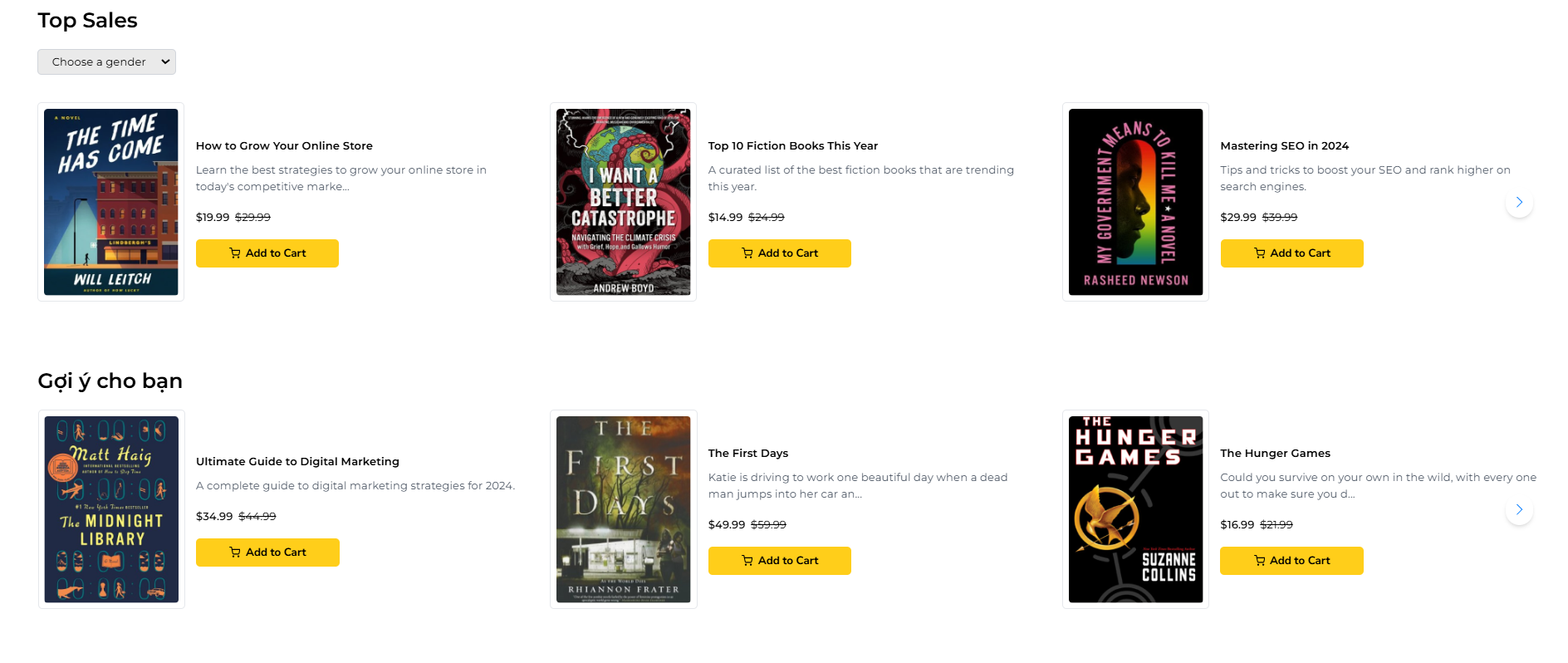
-Icon giỏ hàng: Hiển thị số sản phẩm trong giỏ hàng (sử dụng reducer trong rtk)



## 3.3. Hero banner: Hiển thị ảnh Sản phẩm nổi bật



## 3.4. Top Sale:



- Sử dụng rtk query để GET danh sách sản phẩm qua api từ server

- Sử dụng Swiper để tạo slider cho các sản phẩm

- Dùng dropdown để lọc các sản phẩm theo danh mục.

- Click từng sản phẩm sẽ chuyển sang trang chi tiết qua react router-dom

## 3.5. Màn hình chi tiết sản phẩm

- Dùng rtk query để GET dữ liệu chỉ tiết sản phẩm từ id của sản phẩm trên URL qua api

- Click vào nút “add to cart” để thêm sản phẩm vào giỏ hàng, sử dụng reducer của rtk

- Dùng rtk query để GET sản phẩm có cùng danh mục, sử dụng swiper để tạo slider, có thể click vào các sản phẩm để xem hoặc thêm vào giỏ hàng.

A screenshot of a book

Description automatically generated

## 3.6. Màn hình check out

Sử dụng react hook form để quản lí form check out

A screen shot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

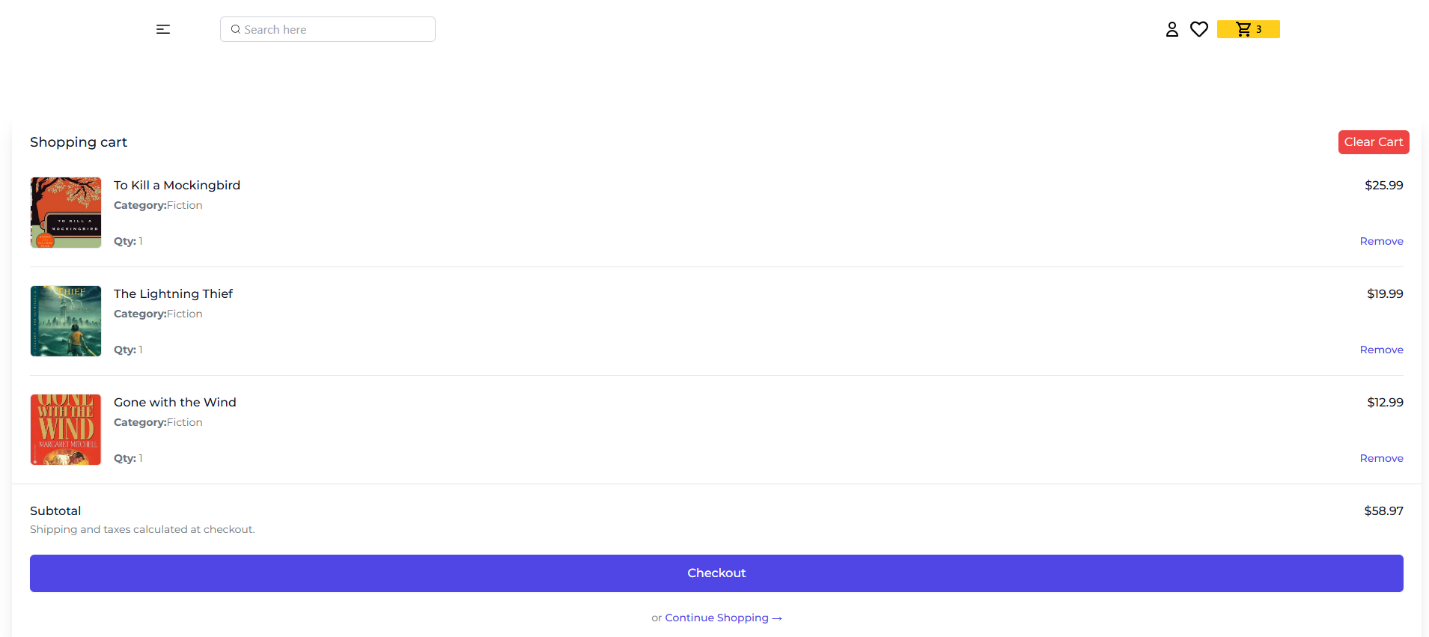
## 3.7. Màn hình shopping cart

Thêm, xóa sản phẩm và cập nhật lại total price:

* addToCart: Thêm sản phẩm vào giỏ nếu chưa có, logic: Kiểm tra sản phẩm đã tồn tại trong giỏ hàng, xử lý dựa trên kết quả kiểm tra (chi tiết trong 3.10)
* removeFromCart: Xóa một sản phẩm khỏi giỏ hàng, logic: Sử dụng filter để tạo một mảng mới chứa các sản phẩm không trùng với \_id của sản phẩm được truyền trong action.payload. Gán lại mảng đã lọc vào state.cartItems
* clearCart: Xóa toàn bộ giỏ hàng, logic: Đặt state.cartItems thành một mảng rỗng

A computer screen shot of a program code

Description automatically generated



## 3.8. Màn hình login và register

- Login bằng tài khoản email đã đăng ký hoặc login bằng tài khoản google, sử dụng API bên thứ 3 (firebase) để thực hiện chức năng.

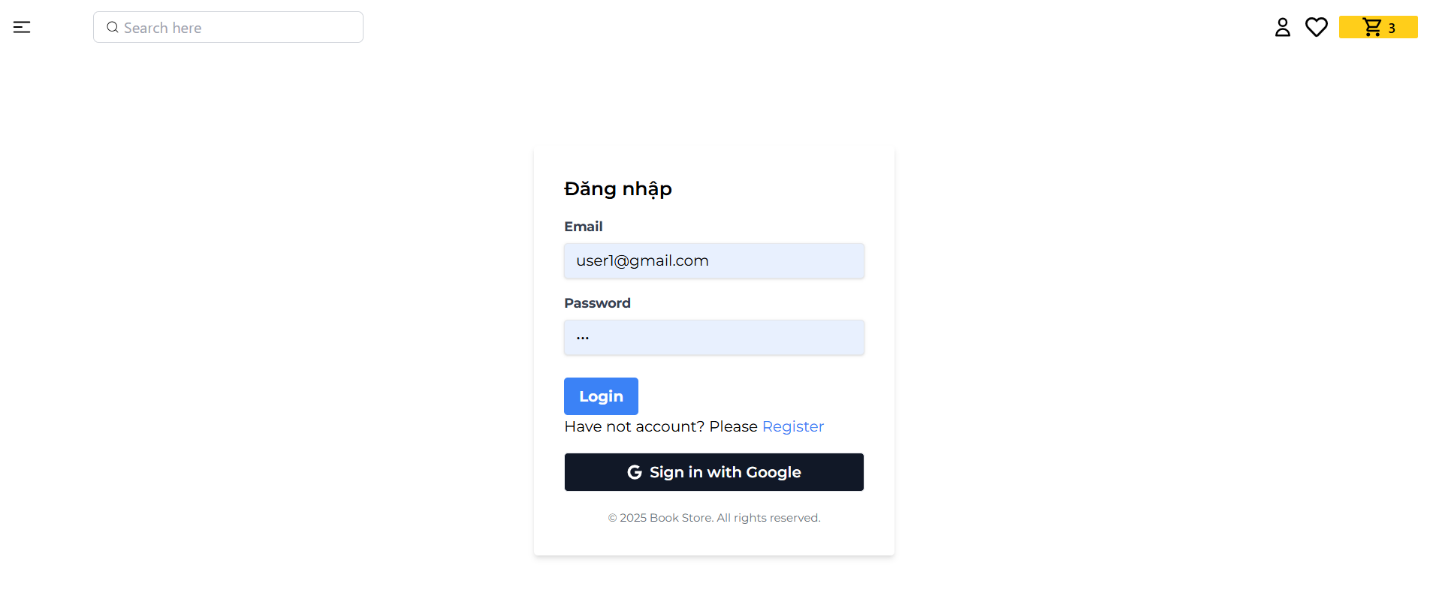
- Tài khoản quản lý trong fỉrebase

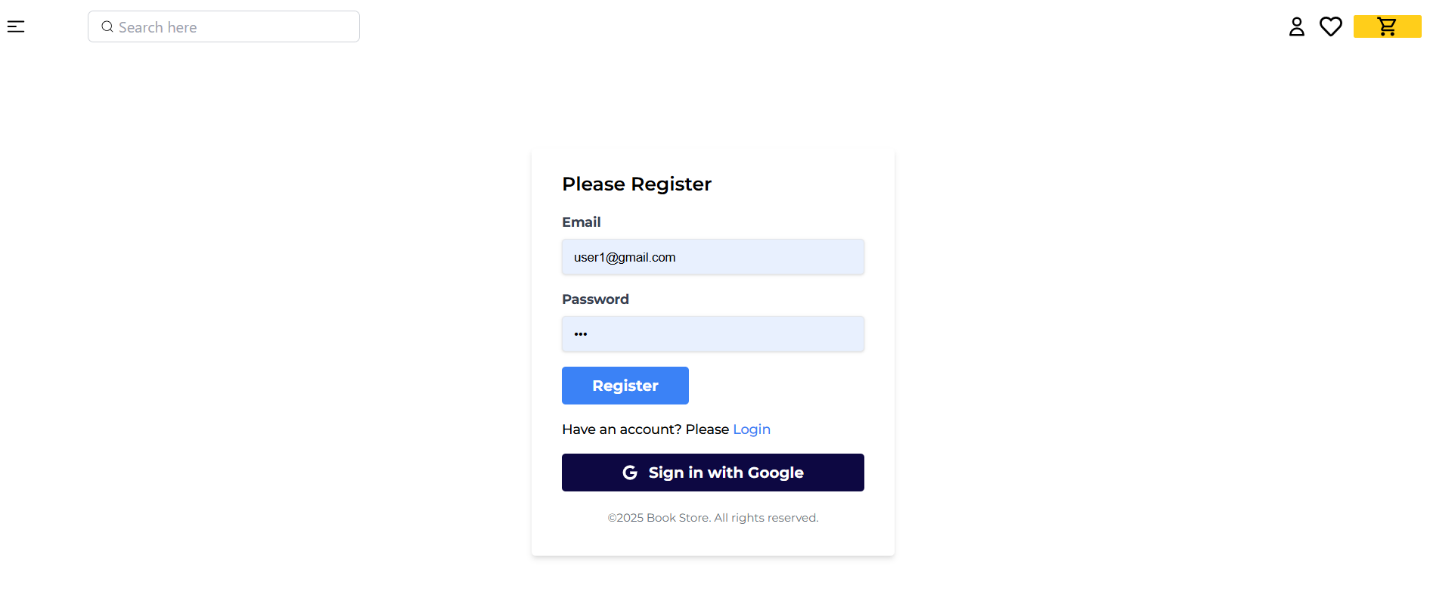
- Chi tiết src code tại CNW/FE/src/AuthContext.jsx

A screenshot of a computer

Description automatically generated

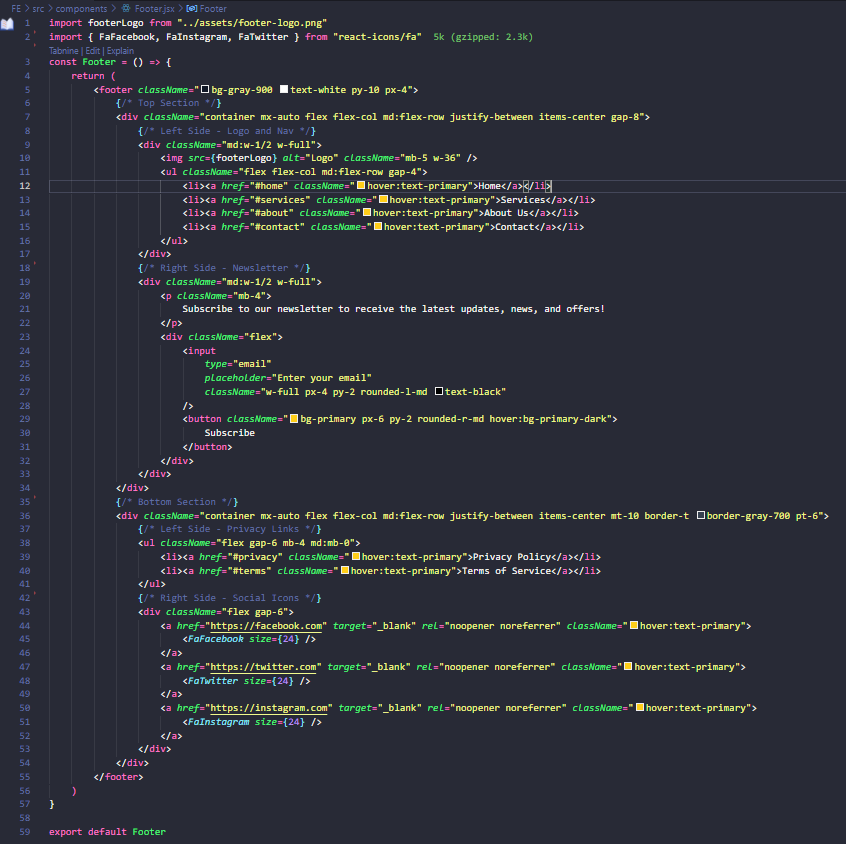
- Tài liệu tham khảo: <https://firebase.google.com/docs/auth?hl=vi>

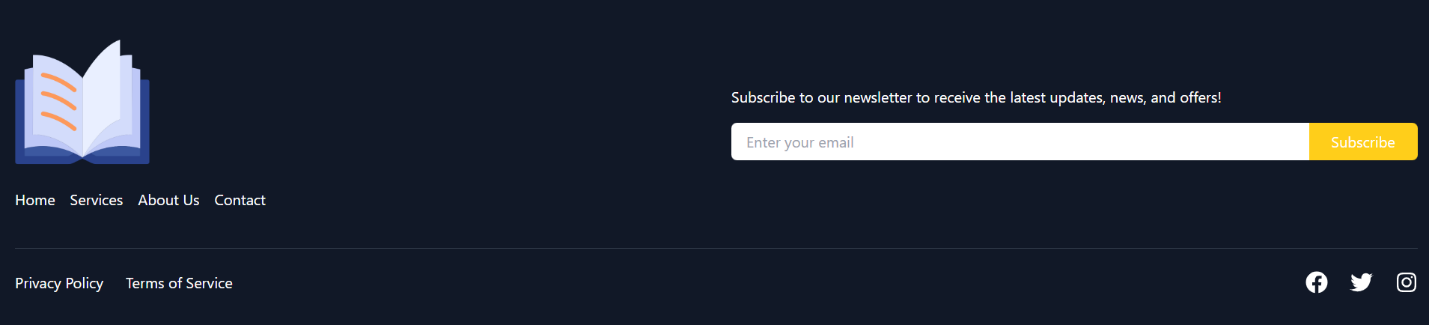




## 3.9. Màn hình footer

- Thiết kế giao diện footer





## 3.10. Thông báo sau khi add sản phẩm thành công và không thành công

- Sử dụng thư viện sweetalert2 để hiển thị các thông báo (alert) thân thiện với người dùng.

B1: cài đặt: npm install sweetalert2

B2: tại file main.jsx: import 'sweetalert2/dist/sweetalert2.js'

B3: tại file cartSlice.js

* Nếu sản phẩm chưa có trong giỏ hàng (!existingItem):

+ Thêm sản phẩm mới vào mảng state.cartItems bằng push.

+ Hiển thị thông báo "Added successfully!" bằng Swal.fire.

* Nếu sản phẩm đã tồn tại trong giỏ hàng (existingItem):

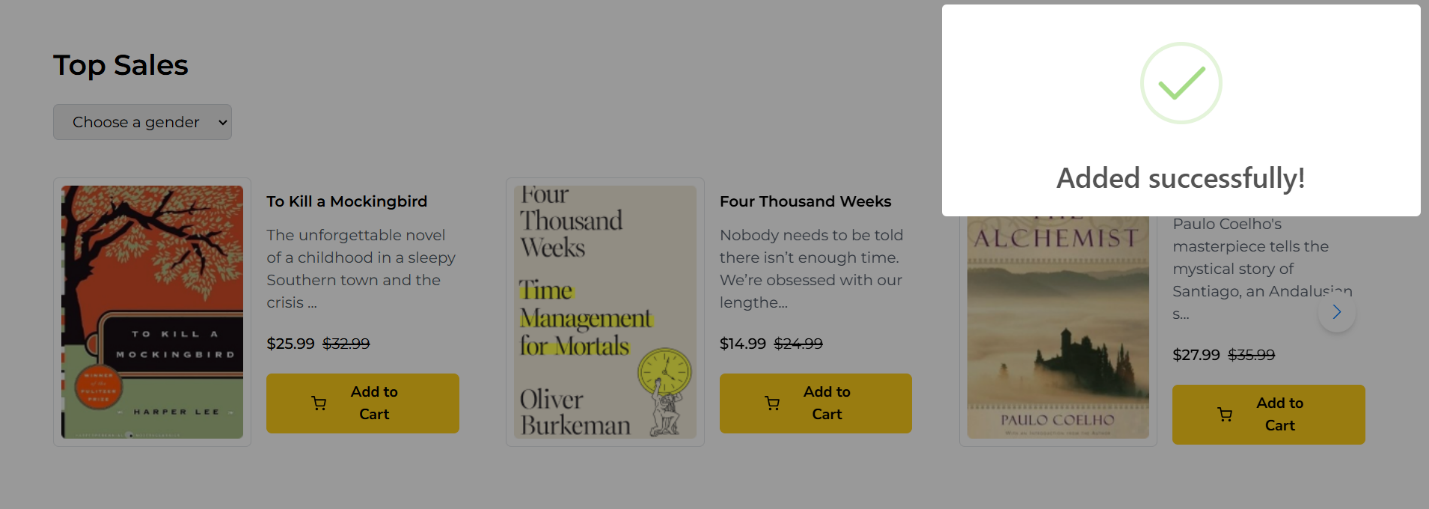
+ Không thêm sản phẩm vào giỏ.

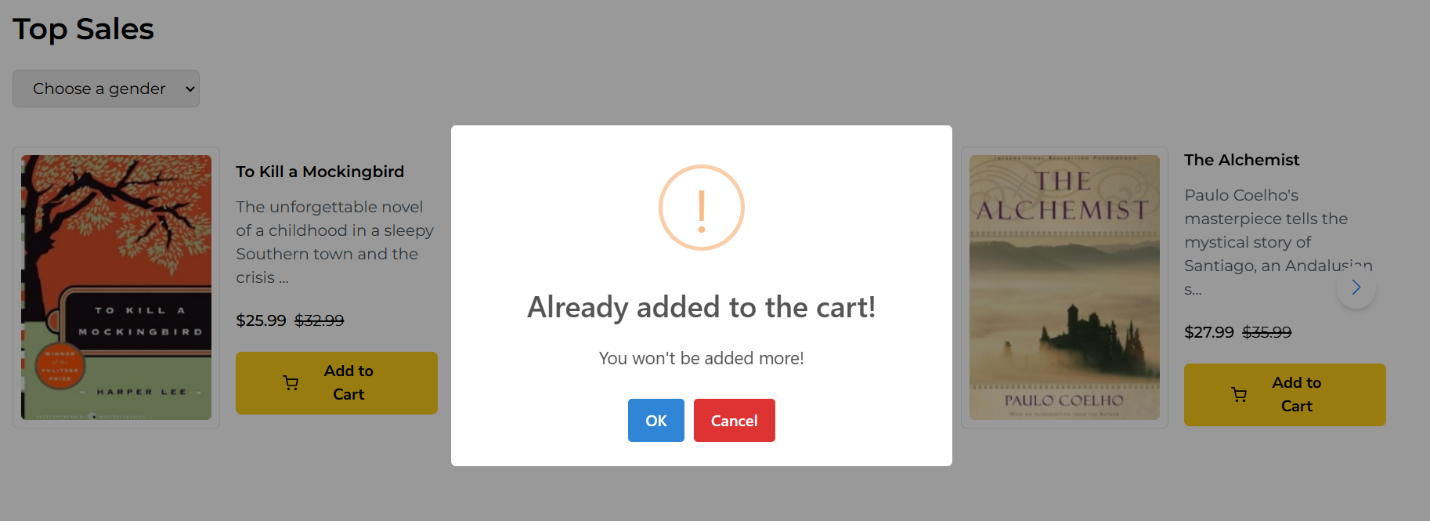
+ Hiển thị thông báo "Already added to the cart!" để cảnh báo người dùng.

A computer screen shot of a program

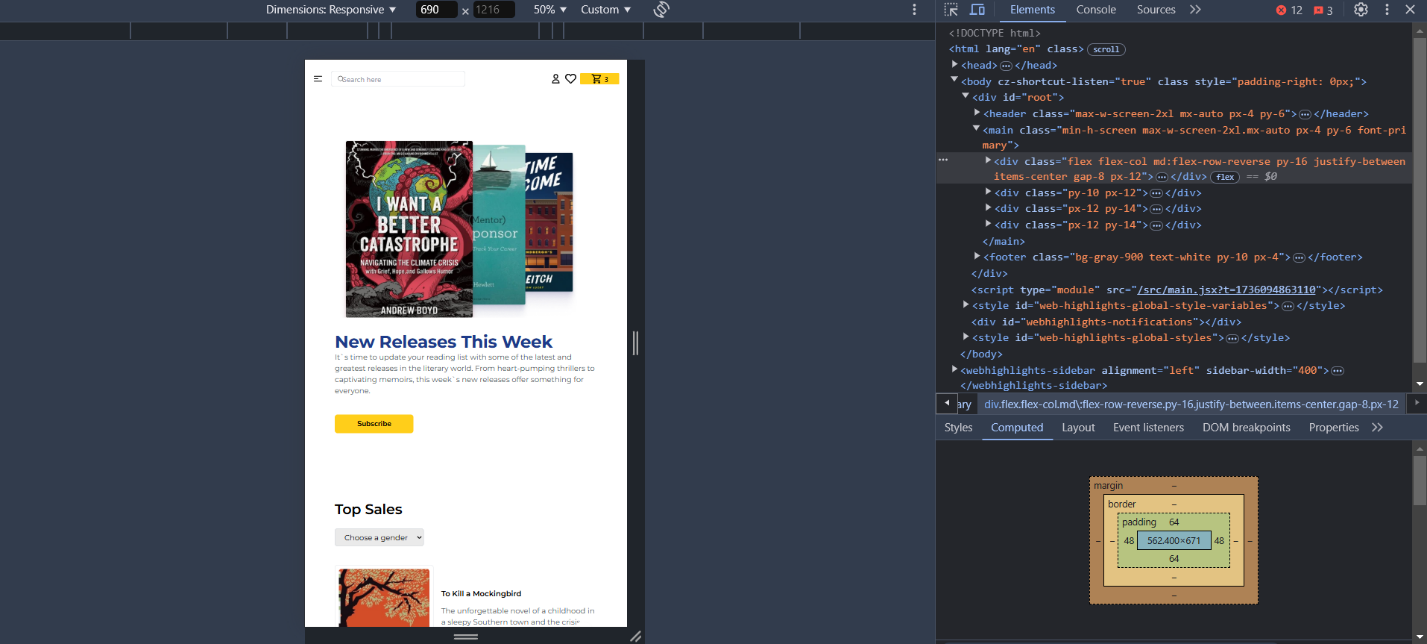
Description automatically generated

* Tài liệu tham khảo: https://sweetalert2.github.io/





## 3.11. Dimensions responsive



Sử dụng các lớp tiện ích từ Tailwind CSS để tạo kiểu dáng và định dạng trang để phù hợp với các thiết bị khác nhau như điện thoại, máy tính bảng hoặc máy tính bàn