**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO BÀI TẬP CÁ NHÂN**

**HỌC PHẦN PHÁT TRIỂN ỨNG DỤNG WEB NÂNG CAO**

**ĐỀ TÀI: TÌM HIỂU NEXTJS VÀ XÂY DỰNG WEBSITE BÁN SÁCH**

**Sinh viên thực hiện**: 2014478 – Nguyễn Xuân Phát - CTK44PM

**Mục lục**

[**I.** **Cơ sở lý thuyết và công nghệ** 4](#_Toc134142894)

[**1.** **NextJs 13** 4](#_Toc134142895)

[**1.1** **Tổng quan Next.js** 4](#_Toc134142896)

[**1.2** **Pages, Dynamic Router, Layout** 4](#_Toc134142897)

[**a)** **Page** 4](#_Toc134142898)

[**b)** **Dynamic router** 5](#_Toc134142899)

[**c)** **Layout** 7](#_Toc134142900)

[**1.3** **Data Fetching** 7](#_Toc134142901)

[**a)** **getInitialProps:** 7](#_Toc134142902)

[**b)** **getServerSideProps** 8](#_Toc134142903)

[**1.4** **API Reference** 9](#_Toc134142904)

[**a)** **Next/router** 9](#_Toc134142905)

[**b)** **Next/link** 9](#_Toc134142906)

[**c)** **Next/image** 10](#_Toc134142907)

[**1.5** **Styling** 11](#_Toc134142908)

[**1.6** **Deployment** 11](#_Toc134142909)

[**2.** **Tailwind** 12](#_Toc134142910)

[**3.** **ASP.Net Core 7** 13](#_Toc134142911)

[**a)** **ASP.Core** 13](#_Toc134142912)

[**b)** **Minimal API** 14](#_Toc134142913)

[**c)** **Jwt Token** 14](#_Toc134142914)

[**d)** **Entity framewok core** 14](#_Toc134142915)

[**II.** **Phân tích và xây dựng website** 15](#_Toc134142916)

[**1.** **Sơ đồ use case** 15](#_Toc134142917)

[**a)** **Use case người dùng:** 15](#_Toc134142918)

[**b)** **Use case quản trị** 16](#_Toc134142919)

[**2.** **Danh sách chức năng của website** 18](#_Toc134142920)

[**3.** **Xây dựng hệ thống API** 19](#_Toc134142921)

[**a)** **Account API:** 19](#_Toc134142922)

[**b)** **Categories API:** 19](#_Toc134142923)

[**c)** **Orders API:** 19](#_Toc134142924)

[**d)** **Product API:** 20](#_Toc134142925)

[**e)** **Dashboard API:** 20](#_Toc134142926)

[**4.** **Xây dựng ứng dụng** 21](#_Toc134142927)

[**a)** **Giao diện trang chủ ứng dụng** 21](#_Toc134142928)

[**b)** **Chức năng xem và tìm kiếm sản phẩm:** 22](#_Toc134142929)

[**c)** **Xem chi tiết sản phẩm** 22](#_Toc134142930)

[**d)** **Chức năng xem giỏ hàng và đặt hàng** 23](#_Toc134142931)

[**e)** **Chức năng đăng nhập:** 24](#_Toc134142932)

[**f)** **Chức năng quản lý sản phẩm:** 25](#_Toc134142933)

[**g)** **Chức năng quản lý danh mục** 25](#_Toc134142934)

[**h)** **Chức năng quản lý đơn hàng** 25](#_Toc134142935)

[**III.** **Kết luận và hướng phát triển** 26](#_Toc134142936)

[**1.** **Kết luận** 26](#_Toc134142937)

[**2.** **Hướng phát triển:** 26](#_Toc134142938)

[**Tài liệu tham khảo** 26](#_Toc134142939)

1. **Cơ sở lý thuyết và công nghệ**
2. **NextJs 13**
   1. **Tổng quan Next.js**

Next.Js, được phát triển bởi Vercel, là một framework mã nguồn mở phổ biến cho việc xây dựng ứng dụng React server-side rendering (SSR). Next được xây dựng để thuận tiện cho việc xây dựng ứng dụng website trở nên dễ dàng nhất. Cho phép phát xậy dựng một ứng dụng React với hiệu xuất cao, có khả năng mở rộng và xây dựng website không cần phải cấu hình. Với cấu trúc của Next các lập trình viên có thể tập trung vào quá trình phát triển, xây dựng tính năng của ứng dụng.

Mặc định, Next khởi tạo HTML trước cho mỗi trang nhưng quá trình này được xử lý ở server side chứ không phải ở client-side. Quá trình này được gọi là pre-rendering. Next đảm bảo các JavaScript để tạo ra trang với đầy đủ các tương tác được liên kết với HTML được tạo. JavaScript được tải sau khi trang web được khởi tạo. Tại thời điểm này, React Js hoạt động một Shadow DOM để đảm bảo nội dung hiển thị khớp với nội dung mà ứng dụng React sẽ hiển thị.

* 1. **Pages, Dynamic Router, Layout**

1. **Page**

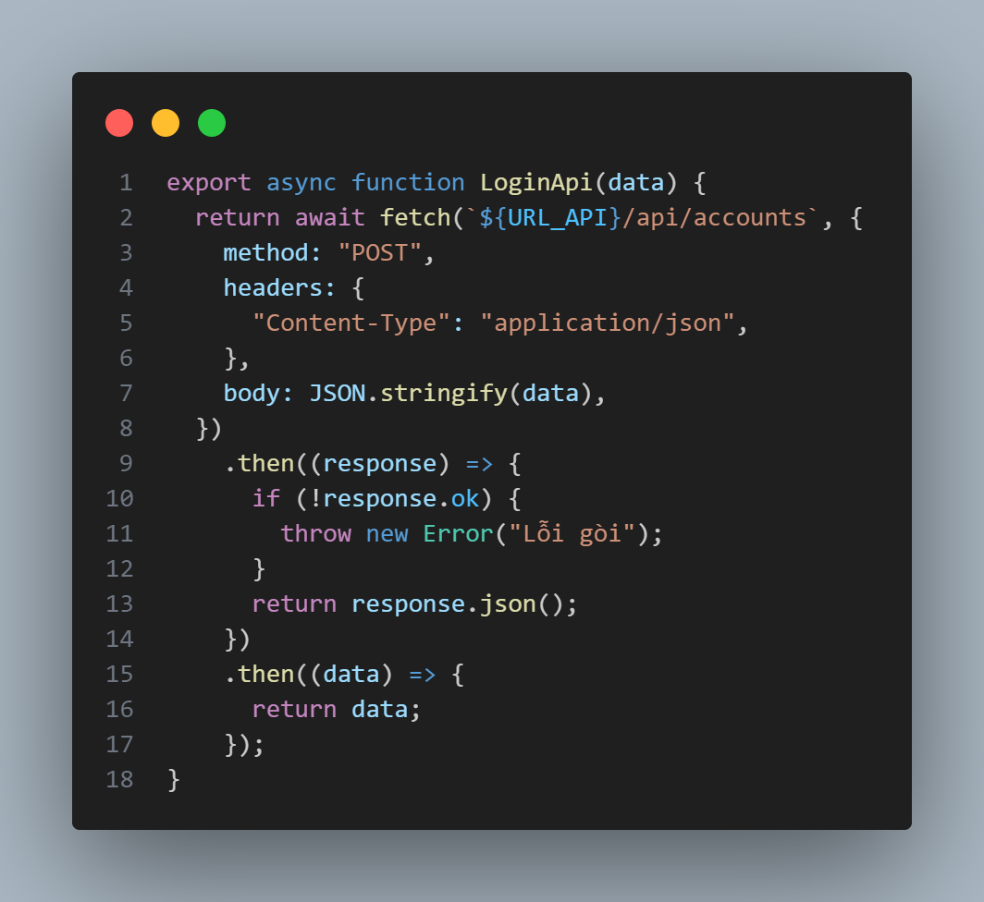
Mỗi trang là một React component được sử dụng với các tệp .js, .jsx, .ts hoặc .tsx, tùy vào cách khởi tạo ứng dụng thư mục lưu trữ page có thể khác nhau đôi chút /app hoặc pages/. Route được xác định với các thư mục lưu trữ file page.jsx.

Ví dụ: trong thư mục pages, nếu bạn tạo một about/page.jsx thì Next sẽ tự tạo cho một bạn một trang với đường dẫn <http://localhost:3000/about> mà không cần phải cấu hình.



Next hổ trợ pre-rendering thông qua server-side rendering và static generation. Có thể kết hợp các phương pháp hiển thị khác nhau trong cùng một ứng dụng trong đó có 1 số trang được tạo bằng SSR và các trang khác được tạo từ static generation. Client-side rendering cũng được sử dụng để hiển thị một số trang cụ thể.

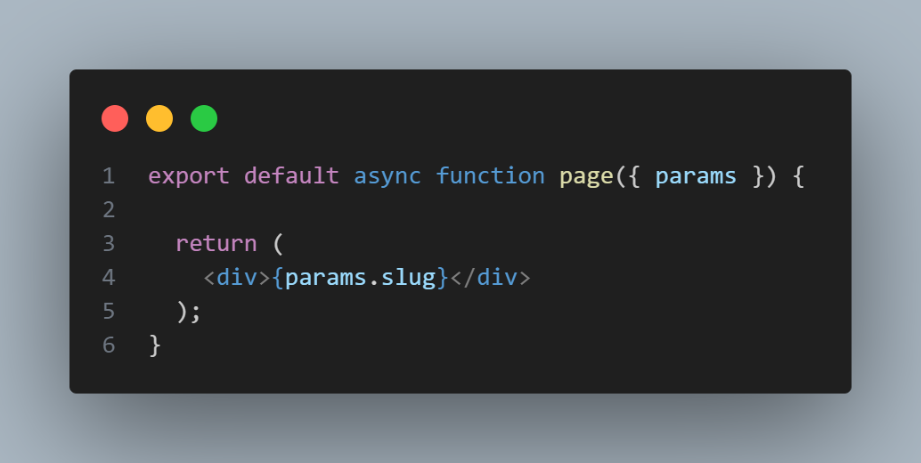
Để sử dụng được server-side rendering, bạn cần phải tạo một một hàm với Async. Ví dụ một hàm async LoginApi:



1. **Dynamic router**

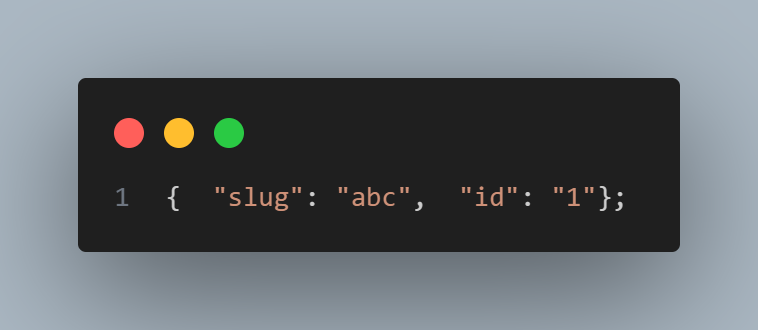
Xác dịnh các Router bằng việc sử dung các đường dẫn được xác định trước không phải lúc nào cũng dễ dàng xây dựng một ứng dụng phức tạp. Trong Next.js, bạn có thể thêm dấu ngoặc vào một trang ([param]) để tạo một Router động (slug, id, …).

Ví dụ, cho một trang pages/post/[slug]/page.jsx, để có thể lấy được giá trị của slug thông qua đoạn mã sau:



Với bất kỳ một Route nào như /post/dynamic-routes hoặc /post/fetch sẽ được lấy về thông qua params với các giá trị tương ứng: dynamic-routes, fetch.

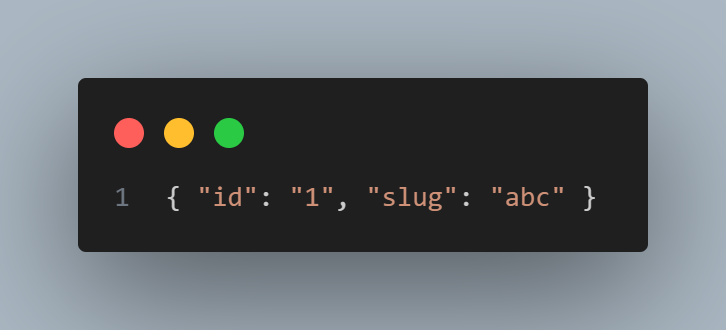
Tương tự, route /post/abc?id=1 sẽ có được đối tượng truy vấn sau:



Tuy nhiên, tham số route sẽ ghi đè những tham số truy vấn với tên giống nhau. Ví dụ, router /post/abc?slug=123 sẽ có đối tượng truy vấn sau:

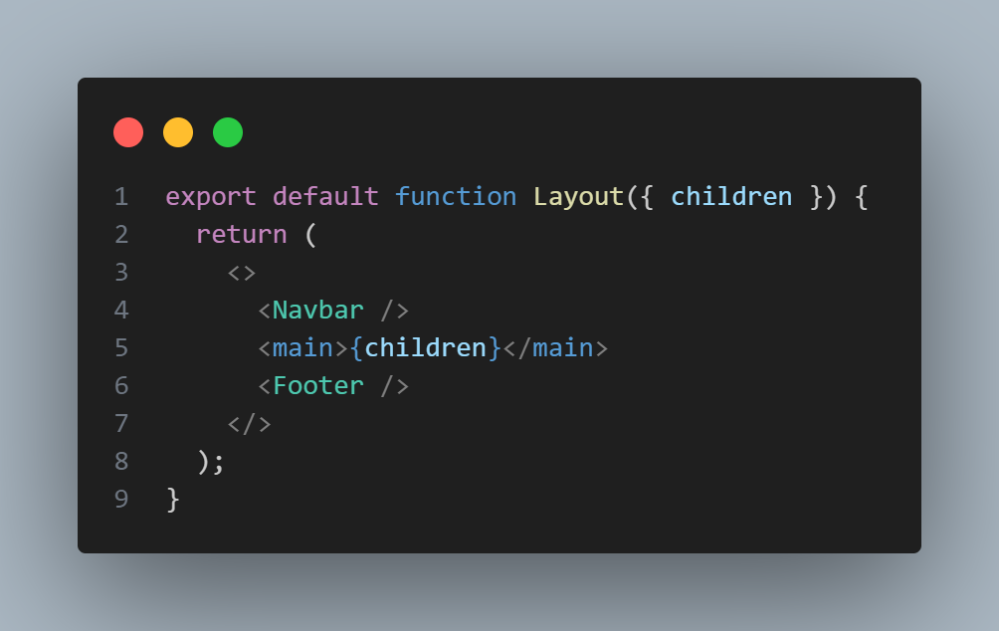


Trong Next, có thể tạo được nhiều dynamic route và phương thức hoạt động có của nó cùng một cách so với một dynamic route. Một trang pages/post/[id]/[slug]/page.jsx sẽ tương ứng với đường dẫn /post/1/abc và giá trị của đối tượng truy vấn như sau:



1. **Layout**

Mô hình React cho phép chúng ta đưa đoạn mã của một trang của các component. Các thành phần này có thể được sửa dụng lại nhiều lần ở các trang khác nhau như: thanh điều hướng, footer, …



Điều hướng các dynamic router ở client-side được xử lý với “next/link”. Nếu bạn muốn di chuyển đến các route khác, sẽ được sử dụng như thế này:



* 1. **Data Fetching**

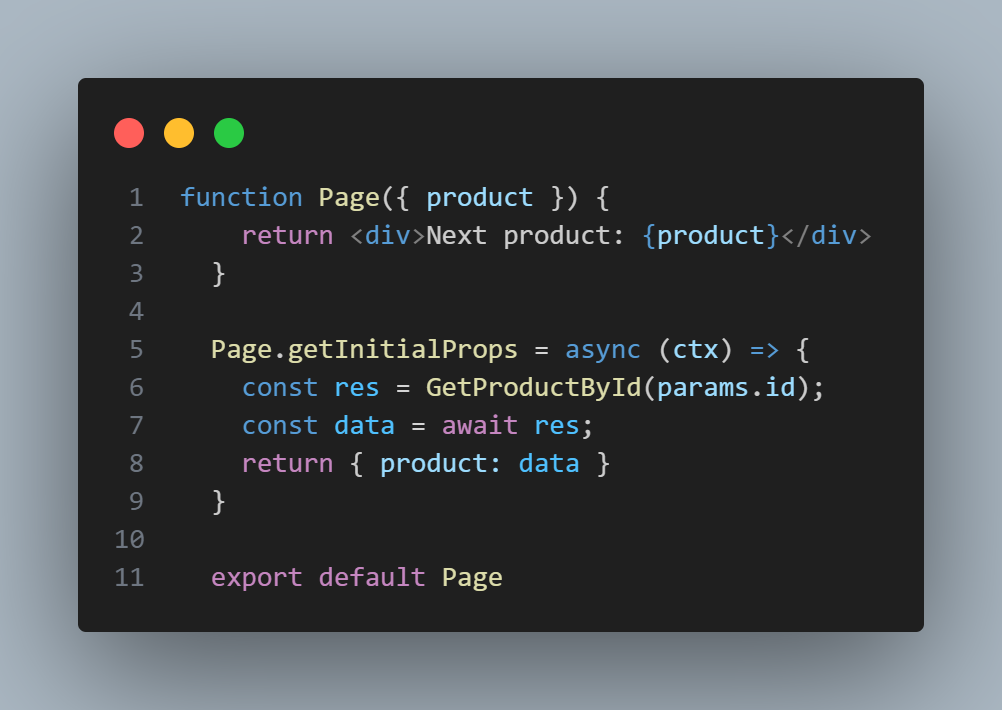
Data fetching trong Next.js cho phép bạn hiển thị nội dung của mình theo nhiều khác nhau, tùy thuộc vào trường hợp sử dụng ứng dụng. Bao gồm pre-rendering với server-side rendering hoặc static generation.

Server-side render sử dụng getInitialProps và getServerSideProps:

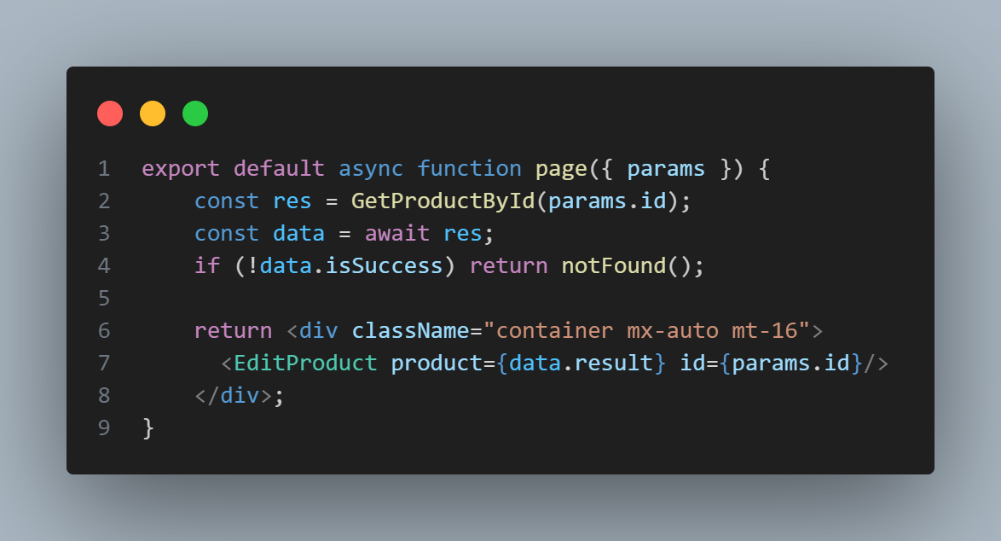
1. **getInitialProps:**

getInitialProps cho phép ở server-side rendering và cho phép thực hiện khởi tạo dữ liệu ban đầu trước khi client-side rendering được khởi chạy. Điều này có nghĩa dữ liệu sẽ được giử trang với dữ liệu đã được lấy từ server. Rất hữu ích cho SEO. getInitialProps sẽ vô hiệu hóa tính năng Automatic Static Optimization

getInitialProps là một async function có thể được thêm vào bất kỳ một phương thức tĩnh:



Hoặc có thể sử dụng một lớp component:



1. **getServerSideProps**

Khi bạn xuất một hàm gòi là getServerSideProps (SSR) từ một trang. Nextjs sẽ khởi tạo trước trang này dự trên mỗi yêu cầu sử dụng mà dữ liệu được trả về bởi getServerSideProps. Điều này rất hữu ích khi bạn muốn fetch dữ liệu khi thay đổi thường xuyên, và có thể cập nhật và hiển thị dữ liệu mới nhất

* 1. **API Reference**

1. **Next/router**

Next router là một module của Next.js, cung cấp chúng ta các công cụ để thực hiện routing (định tuyến) trên client-side của ứng dụng Next.js. Các chức năng chính của next/router bao gồm:

* push (url: string, as?: string, option?: {}): chuyển hướng đến một URL mới và đưa URL đó vào history stack của trình duyệt. Tham số “as” được sử dụng để định nghĩa các URL hiển thị trên trình duyệt, tham số “option” chứa các tùy chọn khác liên quan đến việc chuyển hướng. Ví dụ về một router.push():



* Giả sử user được khởi tạo với giá trị null, khi trang render sẽ kiểm tra thông qua useEffect() nếu user là null sẽ hiểu là chưa được đăng nhập và tự chuyển hướng đến trang /login.
* replace(url: string, as?: string, options?: {}): Thay đổi URL hiện tại trên trình duyệt mà không thêm URL mới vào history stack. Tham số "as" và "options" được sử dụng tương tự như trong hàm push()
* back(): Quay lại trang trước đó trong history stack của trình duyệt.

1. **Next/link**

Next Link là một component của Next.js, cung cấp cho chúng ta các công cụ để thực hiện navigation (điều hướng) giữa các trang trong ứng dụng Next.js mà không cần phải tải lại toàn bộ trang. Các thành phần của next/link bao gồm:

* Href: Định nghĩa URL của trang tiếp theo mà component next/link sẽ đưa người dùng đến
* As: Định nghĩa URL hiển thị trên trình duyệt, thường được sử dụng để định tuyến trang bằng một đường dẫn tự chọn
* Prefetch: Tiền tải trang tiếp theo khi người dùng hover (rê chuột) lên component next/link

Một ví dụ đơn giản với các tệp của trang (pages):

* Pages/page.jsx
* Pages/about/page.jsx
* Pages/blog/[slug]

Chúng ta sẽ có được cách để sử dụng next/link như sau:



1. **Next/image**

Next/image là một component của Next.js, cung cấp cho chúng ta các công cụ để tối ưu hóa tải hình ảnh lên trên trang web. Các chứng năng chính của Image:

* Src: đường dẫn của hình ảnh, đối với hình ảnh không dùng trực tiếp từ client cần phải cấu hình domain để có thể hiển thị được hình ảnh
* Width , height: Kính thước của ảnh khi hiển thị trên trang web
* Alt: Mô tả cho ảnh

Ví dụ khi sử dụng hình ảnh với đường dẫn có domain là localhost:



Để có thể hiển thị được hình ảnh ở server local bạn cần phải cấu hình domain của hình ảnh ở file next.config.js như sau:



* 1. **Styling**

Next.js hỗ trợ nhiều cách thức để áp dụng được CSS trong ứng dụng. Có thể sử dụng qua các modules, styled-jsx,… Next.js có một tính năng gọi là styled-jsx, cho phép bạn viết CSS inline trong file JSX của component. Các rules của CSS được giới hạn trong phạm vi của component đó không gây tác động đến các component khác.

Ngoài ra, Next.js còn hỗ trợ các thư viện styling như Tailwind CSS, Bootstrap, ... Có thể sử dụng một trong các thư viện này để xây dựng giao diện một cách thuận tiện hơn.

Next.js cũng tích hợp các trình biên tập như CSS, SASS và Less giúp cho việc xây dựng website trở nên dễ dàng hơn

* 1. **Deployment**

Vercel là một dịch vụ đám mây được thiết kế đặc biệt dành riêng cho việc phát triển và triển khai ứng dụng Next.js. Nó cho phép bạn triển khai ứng dụng Next.js chỉ trong vài bước đơn giản bằng cách kết nối với GitHub hoặc GitLab. Ngoài ra Vercel còn hỗ trợ deploy ứng dụng web được xây dựng bằng React.js

Sau khi đã kết nối GitHub hoặc GitLab, vercel sẽ tự động triển khai và cập nhật lại mã nguồn khi bạn chỉnh sửa mã nguồn. Ngoài ra, Vercel còn hỗ trợ tính năng pre-rendering cho các ứng dụng Next.js. Giúp thời gian phản hồi của ứng dụng được cải thiện.

Graphical user interface, text, application

Description automatically generated

1. **Tailwind**

Tailwind là một framework css hỗ trợ việc xây dựng và phát triển giao diện (UI) bằng cách sử dụng các lớp CSS được xây dựng sẵn. Với Tailwind, người dùng có thể sử dụng các class utility để kiểm soát một phần lớn các thuộc tính CSS như margin, padding, width, height, border, font-size, font-weight, color, background-color, và nhiều hơn nữa.

Tailwind giúp người dùng tiết kiệm thời gian trong việc phát triển giao diện vì nó cung cấp cho người dùng các lớp CSS, giúp tăng tốc độ viết code.

Để sử dụng được Tailwind trong Next.js cần phải cài các gói thư viện cần thiết.



Sau khi chạy lệnh thành công, cần phải chạy thêm lệnh để khởi tạo file “tailwind.config.js”:



Sau đó cần phải cấu hình để tailwind có thể hoạt động được trong Next.js:



1. **ASP.Net Core 7**
2. **ASP.Core**

API.NET Core là một framework phát triển ứng dụng web động. Đây là một framework mã nguồn mở, miễn phí và có thể chạy trên nhiều nền tảng khác nhau, bao gồm Windows, macOS và Linux.

Routing trong ASP.Net core giúp bạn định tuyến các yêu cầu đến các endpoint phù hợp. Endpoint là một địa chỉ web đại diện cho một tài nguyên cụ thể. Bằng cách sử dụng routing, bạn có thể định tuyến các yêu cầu từ phía client lên server.

Middleware cho phép thêm các xử lý tùy chỉnh vào quá trình xử lý các yêu cầu và phản hồi. Middleware là thành phần trung gian khi xử lý cái request. Có thể áp dụng Middleware vào để bổ sung thêm kết quả xử lý của các Status code. Ví dụ: khi bạn gửi token lên một endpoint yêu cầu token, khi token sai hoặc không có, lúc này middleware sẽ được custom lại để thêm vào một số thông báo message ở phần body của Results với status code 401.

Models là các lớp đại diện cho dữ liệu trong ASP.Net Core, models được sử dụng để biểu diễn các đối tường dữ liệu được lưu trong database, hoặc để truyển qua lại dữ liệu thông qua các endpoint.

1. **Minimal API**

Minimal API là một tính năng được pháp triển từ ASP.Net core 6, cung cấp một cách tiếp cận xây dựng API đơn giản, nhẹ và hiệu quả hơn. Với Minimal Api, lập trình viên có thể tạo ra các API mà không cần phải tạo một số thành phần phức tạp như Controller API, Controller cần phải có một lớp để định nghĩa các router và nhiều thứ khác.

Ưu điểm của Minimal API là cấu trúc đơn giản, dễ đọc, dễ hiểu so với Controller API. Điều này giúp lập trình viên có thể dễ dàng phát triển và sửa lỗi của từng Endpoint một cách đơn giản hơn.

Với Minimal API, lập trình viên có thể dễ dàng thêm các endpoint cho API của họ bằng cách sử dụng phương thức MapGet, MapPost, MapPut, MapDelete tương ứng với phương thức HTTP mà API hỗ trợ.

Một ưu điểm của Minimal Api, có thể dễ dàng kiểm soát hoàn toàn các HTTP responsive thông qua interface IResult, có thể trả được các trạng thái của Http. Ví dụ để trả về một endpoint Not Found 404, có thể trả về Result.NotFound().

1. **Jwt Token**

JWT (JSON Web Tokens) là một tiêu chuẩn mã hóa token định dạng JSON dựa trên web. Nó cung cấp một cách đơn giản để truyền tải thông tin xác thực giữa các bên bằng cách sử dụng một chuỗi chữ ký ký số. Những token này có thể được mã hóa để bảo vệ chúng khỏi sự tấn công của hacker.

Trong .Net Core, chúng ta có thể sử dụng thư viện như: Microsoft.AspNetCore.Authentication.JwtBearer để xác thự Jwt Token

1. **Entity framewok core**

Entity Framework Core là một bộ công cụ ORM (Object-Relational Mapping) được phát triển bởi Microsoft. Nó cung cấp cho các nhà phát triển .NET một cách tiếp cận dễ dàng để tương tác với cơ sở dữ liệu từ code .NET.

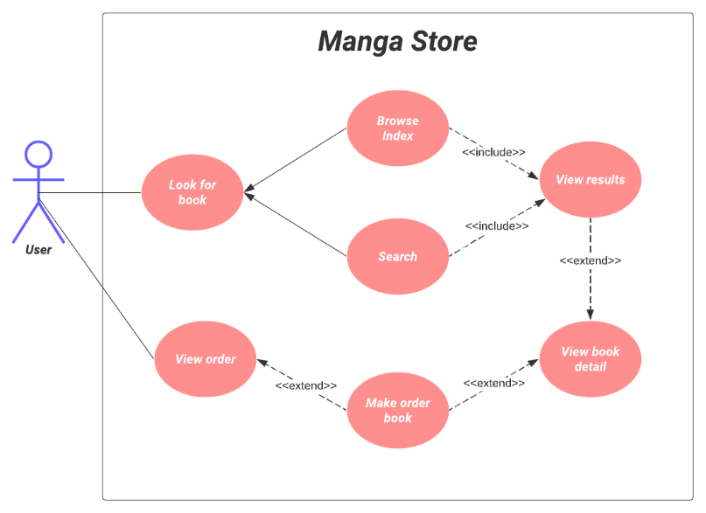
EF core hỗ trợ nhiều loại database như: Sql Server, MySQL, PostgreSQL, SQLite, Oracle, …

EF core dễ cài đặt và sử dụng, cung cấp cho người dùng thông qua Linq để có thể dễ dàng truy vấn dữ liệu trong .Net. Ngoài ra EF core được phát triển để tích hợp với ASP.Net Core tốt nhất.

Một số khái niệm cơ bản khi sử dụng Entity Framework Core:

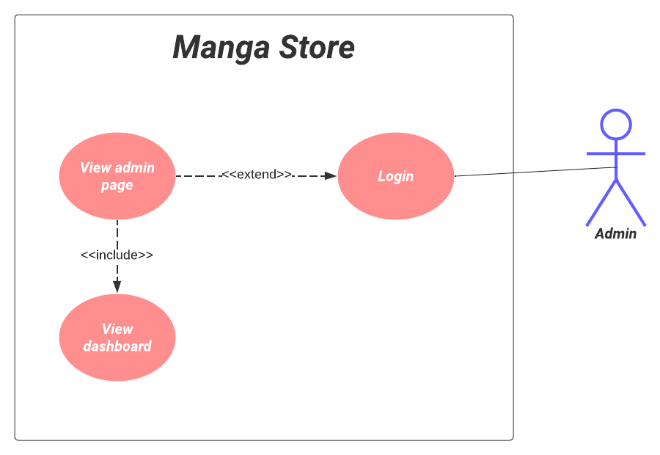
* Model: Model là một tập hợp các đối tượng được định nghĩa trong code để mô tả cấu trúc của cơ sở dữ liệu.
* DbSet: DbSet là một tập hợp các đối tượng trong một model. Nó giúp cho người dùng có thể thao tác với các đối tượng này một cách dễ dàng.
* Migration: Migration là một quá trình để cập nhật cấu trúc của cơ sở dữ liệu. Khi thay đổi cấu trúc của model, người dùng cần phải tạo ra một migration mới để cập nhật cơ sở dữ liệu.
* DbContext: DbContext là một lớp được sử dụng để tạo kết nối tới cơ sở dữ liệu. Nó cũng cung cấp cho người dùng các API để tương tác với cơ sở dữ liệu.

1. **Phân tích và xây dựng website**
2. **Sơ đồ use case**
3. **Use case người dùng:**

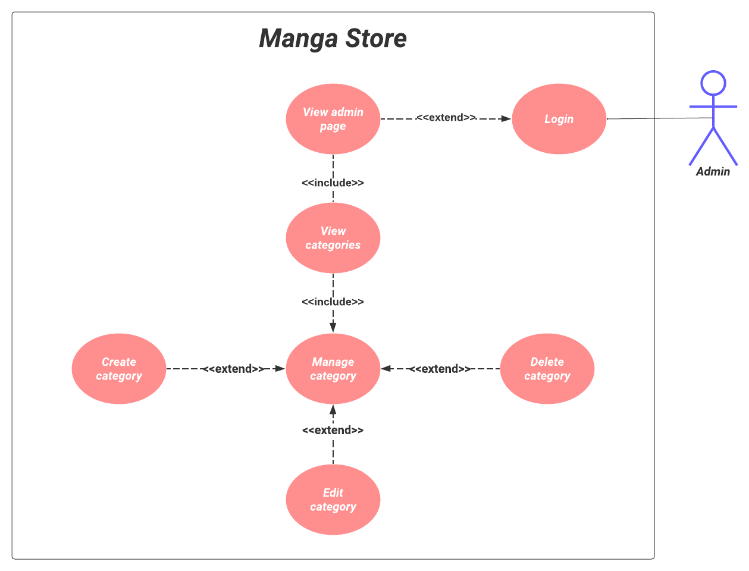
****

1. **Use case quản trị**

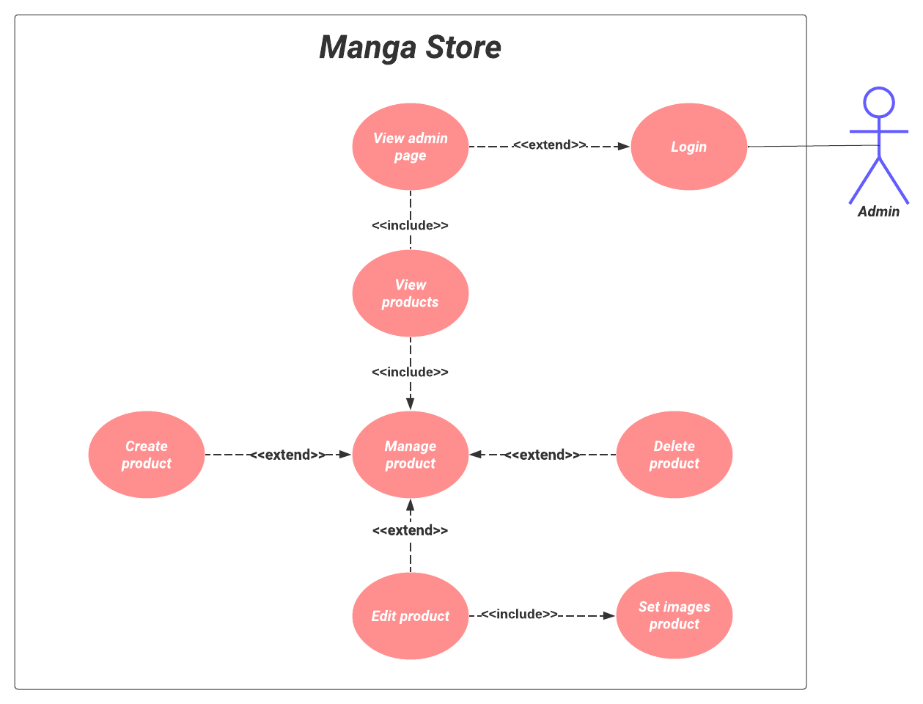
Xem trang thông kê:



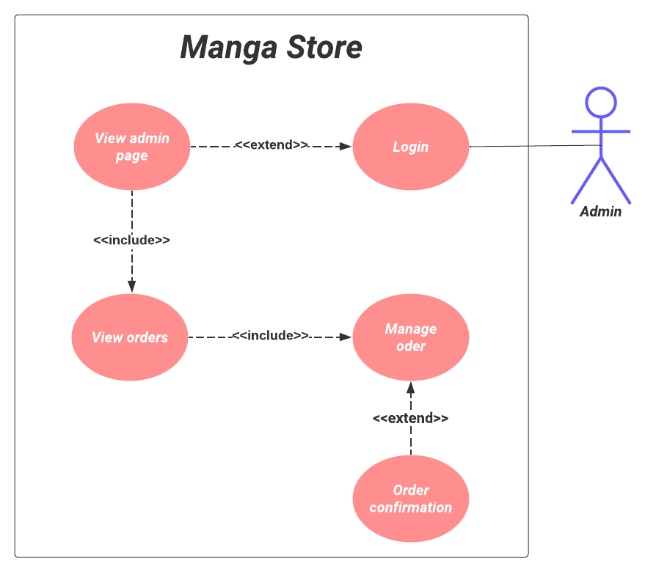
Quản lý danh mục sản phẩm:



Quản lý sản phẩm:



Quản lý hóa đơn:



1. **Danh sách chức năng của website**

|  |  |  |
| --- | --- | --- |
| STT | Loại | Chức năng |
| 1 | Quản lý danh mục | * Hiển thị danh sách danh mục * Tạo danh mục * Chỉnh sửa danh mục * Xóa danh mục |
| 2 | Quản lý sản phẩm | * Hiển thị danh sách sản phẩm * Tạo sản phẩm * Chỉnh sửa sản phẩm * Xóa sản phẩm * Thêm hình cho sản phẩm |
| 3 | Quản lý hóa đơn | * Hiển thị danh sách hóa đơn * Xác nhận đơn hàng |
| 4 | Quản lý thông kê | * Hiển thị thống kê   Thống kê danh thu (ngày, toàn bộ thời gian)  Thống kê số lượng đơn hàng (ngày, toàn bộ thời gian)  Thống kê số lượng (sản phẩm, danh mục) |
| 5 | Danh sách sản phẩm | * Hiển thị sản phẩm bán chạy * Hiển thị sản phẩm liên quan * Hiển thị danh sách sản phẩm |
| 6 | Xem chi tiết sản phẩm | * Hiển thị chi tiết sản phẩm * Hiển thị những sản phẩm có liên qua |
| 7 | Tìm kiếm (admin page) | * Tìm kiếm sản phẩm theo từ khóa * Tìm kiếm danh mục theo từ khóa * Tìm kiếm hóa đơn theo từ khóa |
| 8 | Tìm kiếm | * Tìm kiếm sản phẩm theo từ khóa |
| 9 | Đăng nhập | * Đăng nhập |

1. **Xây dựng hệ thống API**
2. **Account API:**

Graphical user interface, text, application, email

Description automatically generated

1. **Categories API:**

Graphical user interface, application

Description automatically generated

1. **Orders API:**

Graphical user interface, application, email

Description automatically generated

1. **Product API:**

Graphical user interface, application

Description automatically generated

1. **Dashboard API:**

Graphical user interface, application, Teams

Description automatically generated

1. **Xây dựng ứng dụng**
2. **Giao diện trang chủ ứng dụng**

Trang chủ website hiển thị thông tin quảng cáo về cửa hàng, các loại mặt hàng hiện có, hiển thị top 8 sản phẩm bán chạy mà còn hàng tồn của cửa hàng.

Map

Description automatically generated

Text

Description automatically generated

1. **Chức năng xem và tìm kiếm sản phẩm:**

Xem được toàn bộ sản phẩm hiện có của cửa hàng, có thể tìm kiếm sản phẩm với từ khóa theo bất kỳ dạng từ khóa nào (tên sản phẩm, mô tả sản phẩm, slug, …):

A picture containing diagram

Description automatically generated

1. **Xem chi tiết sản phẩm**

Trang chi tiết xem được đầy đủ thông tin của sản phẩm (tên sản phẩm, giá tiền, hình ảnh, số lượng còn lại), sau khi đặt hàng thành công số lượng sản phẩm sẽ được cập nhật lại:

Graphical user interface

Description automatically generated with medium confidence

1. **Chức năng xem giỏ hàng và đặt hàng**

Có thể xem nhanh giỏ hàng từ icon giỏ hàng trên thanh điều hướng góc trên bên phải màn hình:

Graphical user interface, text, application

Description automatically generated

Giỏ hàng chi tiết được hiển thị như sau, đồng thời có thể đặt hàng từ trang này:

Graphical user interface, text, application

Description automatically generated

Sau khi chọn vào nút thanh toán, sẽ chuyển hướng đến trang nhập thông tin để lưu đơn hàng vào hệ thống, mặc định sẽ lưu đơn hàng được tạo mới vào chưa xác nhận:

Graphical user interface, application

Description automatically generated

1. **Chức năng đăng nhập:**

Đăng nhập với dữ liệu đúng sẽ tự động chuyển sang quản trị, đồng thời sẽ thông báo lỗi nếu nhập sai tên đăng nhập hoặc mật khẩu:

Graphical user interface, text, application, chat or text message, email

Description automatically generated

Sau khi đăng nhập thành công, trang sẽ tự động chuyển hướng sang trang quản trị:

Graphical user interface, text, application, email

Description automatically generated

1. **Chức năng quản lý sản phẩm:**

Hiển thị danh sách sản phẩm bao gồm những sản phẩm đã hết hàng, có thể tìm kiếm tương tự như trang ở người dùng. Có thể thêm ,chỉnh sửa và xóa sản phẩm:

A screenshot of a computer

Description automatically generated

Trang chỉnh sửa sản phẩm có thể thay đổi các thông tin của sản phẩm, đồng thời có thể thêm 1 hay nhiều hình cho 1 sản phẩm:

Graphical user interface, text

Description automatically generated

1. **Chức năng quản lý danh mục**

Hiển thị danh sách danh mục, có thể tìm kiếm theo từ khóa (tên, mô tả). Có thể thêm ,chỉnh sửa và xóa danh mục:

Graphical user interface, text, application

Description automatically generated

1. **Chức năng quản lý đơn hàng**

Hiển thị danh sách đơn hàng, có thể tìm kiếm theo từ khóa (thông tin khách hàng, thông tin của đơn hàng, ...). Quản trị có thể quản lý xác nhận đơn hàng từ người dùng gửi lên:

Graphical user interface, text, application, email

Description automatically generated

1. **Kết luận và hướng phát triển**
2. **Kết luận**

Sau thời gian tìm hiểu và xây dựng ứng dụng đạt được thành quả như sau:

* Tìm hiểu tổng quan về Next.js và các cách thức áp dụng vào pháp triển ứng dụng.
* Hiểu được các vận hành của Next.js: Router, Dynamic Router, Fetching data.
* Hiểu được kỹ thuật tương tác dữ liệu thông quan API
* Hiểu và nắm vững xây dựng một web server bằng ASP.Net core 7, sử dụng Minimal API để dựng các endpoint
* Xây dựng được hệ thống bán sách bằng công nghệ Next.js

1. **Hướng phát triển:**

Tiếp tục phát triển những chức năng còn sót của hệ thống để có đa dạng tương tác hơn

Cải thiện và sửa lỗi hệ thống

# **Tài liệu tham khảo**

[1] Reactjs: <https://reactjs.org/>

[2] Entity Framework Core: <https://learn.microsoft.com/en-us/training/>[modules/persist-data-ef-core](https://learn.microsoft.com/en-us/training/modules/persist-data-ef-core/)

[3] ASP.NET Core: <https://learn.microsoft.com/vi-vn/aspnet/core/?view=aspnetcore-6.0>

[4] Minimal APIs: <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/minimal-apis/overview?view=aspnetcore-7.0>

[5] Nextjs: https://nextjs.org/docs

[6] Nextjs tutorial: https://nextjs.org/learn/foundations/about-nextjs?utm\_source=next-site&utm\_medium=nav-cta&utm\_campaign=next-website

[7] Nextjs 13: https://beta.nextjs.org/docs