**Chương 2: NỘI DUNG THỰC TẬP**

2.1 Next JS

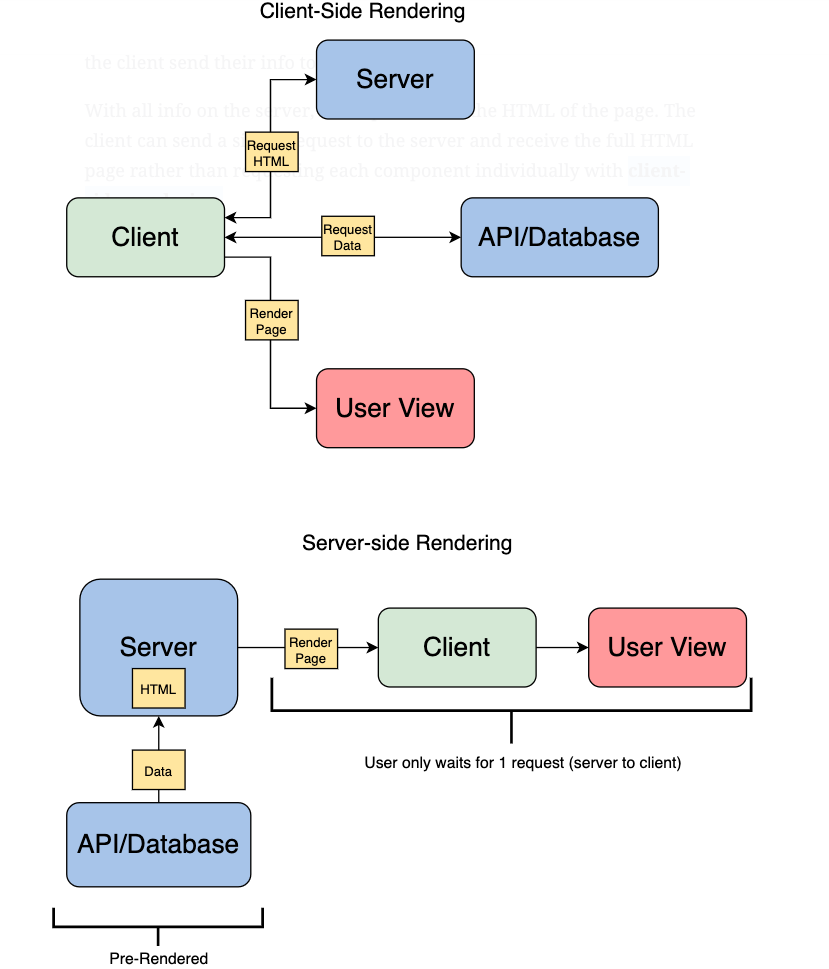
2.1.1. Khái niệm

- Next.js là một framework front-end React được phát triển dưới dạng open-source bổ sung các khả năng tối ưu hóa như render phía máy chủ (SSR) và tạo trang web static. Next.js xây dựng dựa trên thư viện React, có nghĩa là các ứng dụng Next.js sử dụng core của React và chỉ thêm các tính năng bổ sung. Việc triển khai ứng dụng SSR cho phép máy chủ truy cập tất cả dữ liệu được yêu cầu và xử lý JavaScript cùng nhau để hiển thị trang. Sau đó, trang được gửi lại toàn bộ cho trình duyệt và ngay lập tức được hiển thị. SSR cho phép các trang web load trong thời gian nhỏ nhất và tăng trải nghiệm người dùng với khả năng phản hồi nhanh hơn.

- Ngoài ra, sử dụng SSR cũng mang lại cho bạn lợi thế về SEO, giúp trang web của bạn hiển thị cao hơn trên các trang kết quả của công cụ tìm kiếm. SSR làm cho các trang web xếp hạng tốt hơn cho SEO vì chúng tải nhanh hơn và nhiều nội dung trang web có thể được quét bởi các SEO trackers. Thẻ <head> trong Next.js cũng cho phép bạn chỉnh sửa thẻ <head> của một trang web, điều mà bạn không thể thực hiện trong React. Thẻ <head> là một phần cốt lõi trong metadata của trang web và góp phần vào xếp hạng SEO của trang web.

2.1.2. Tại sao sử dụng NextJS?

- Ưu điểm chính của Next.js là hỗ trợ SSR tích hợp để tăng hiệu suất và SEO. Với tất cả thông tin trên server, nó sẽ xử lý để generate ra thông tin HTML của trang/ Sau đó Client có thể gửi một yêu cầu đến Server và nhận toàn bộ trang HTML thay vì yêu cầu từng thành phần riêng lẻ với Client Render.



* Điểm mạnh:

+ Ứng dụng Next.js tải nhanh hơn đáng kể so với ứng dụng React do được render phía Server.

+ Hỗ trợ các tính năng cho static web.

+ Đối với những ai đã có kinh nghiệm làm việc với React thì việc tiếp tập NextJS sẽ là một việc dễ dàng.

+ Tự động code splitting cho các page nhằm tối ưu hoá performance khi load trang.

+ Dễ dàng xây dựng các API internal thông qua các API routes tích hợp sẵn và tạo các endpoit API.

+ Hỗ trợ tích hợp cho route cho page, CSS, JSX và TypeScript.

+ Nhanh chóng thêm các plugin để tùy chỉnh Next.js theo nhu cầu của trang cụ thể của bạn.

* Điểm yếu: Nhược điểm thực sự duy nhất của Next.js là nó là một framework được cố định, có nghĩa là nó có một phương pháp và bộ công cụ cụ thể mà nó muốn bạn sử dụng để xây dựng các ứng dụng của mình. Tuy nhiên, các tùy chọn của Next.js sẽ phù hợp với phạm vi của hầu hết các dự án.

2.1.3. Cách cài đặt

a. Khởi tạo ứng dụng NextJs

- Tương tự với create-react-app của ReactJS thì với NextJS ta cũng có công cụ tương tự là create-next-app. Sau khi cài việc cài đặt thành công ta có thể khởi tạo một ứng dụng NextJS mới với lệnh:

$ create-react-app learn-nextjs

b. Cấu trúc

- Next.js sử dụng hệ thống tệp tối giản để tránh tệp dự án lộn xộn, có nghĩa là điểm bắt đầu chỉ chứa mức tối thiểu cần thiết để chạy ứng dụng. Hiểu cấu trúc thư mục này sẽ giúp bạn thao tác nó để phù hợp với các dự án của riêng bạn.

Mỗi dự án Next.js bắt đầu với 3 thư mục: pages, public và styles.

* Dưới đây là một ví dụ về những gì bạn sẽ tìm thấy trong một dự án Next.js mới:

- pages

- api

- hello.js

- \_app.js

- index.js

- public

- favicon.ico

- vercel.svg

- styles

- globals.css

- Home.module.css

* Pages

+ Thư mục pages chứa các tệp trang của bạn. Mỗi tệp pages là một React Component với một route duy nhất được tạo tự động từ tên tệp. Ví dụ, trang hello.js Next.js sẽ được tìm thấy tại page/hello.js.

+ Một số trang như \_app.js ở trên, bao gồm tiền tố gạch dưới trong tên của chúng để đánh dấu chúng là component tùy chỉnh. Các component này được Next.js sử dụng để làm việc với các component khác.

+ Ví dụ: \_app.js được sử dụng để bắt đầu mỗi trang và không được sử dụng làm trang web của riêng nó.

* Public

+ Thư mục này dành cho việc phân phát tệp tĩnh, có nghĩa là các tệp này không thay đổi và chỉ có thể được tham chiếu.

+ Thư mục này thường chứa hình ảnh hoặc biểu tượng mà trang web sử dụng cũng như thông tin nội bộ như Xác minh trang web của Google.

+ Ví dụ: Trong thư mục Public hiện có favicon.ico là một biểu tượng nhỏ để sử dụng trên các tab của trình duyệt và vercel.svg hiển thị biểu tượng của công ty.

* Styles

+ Thư mục này chứa các tệp CSS của toàn bộ ứng dụng. Tệp Gloals.css thiết lập tiêu chuẩn chung mà tất cả các trang trong dự án sẽ sử dụng.

+ Bạn cũng có thể thêm style dành riêng cho component bằng cách sử dụng các tệp module được đặt tên với hậu tố module, <componentName> .module.css.

+ Routing/Navigation in your App.

+ Navigation cho phép user của bạn có thể điều hướng giữa các trang web với nhau trong Next.js. Routes và Links là hai phương pháp chính bạn có thể sử dụng để xác định cách điều hướng trang web. Các Route trong Next.js có thể tiếp cận được do Next đã tích hợp chức năng định tuyến vào của từng component trong thư mục page. Để tối ưu hóa định tuyến ứng dụng của bạn, điều quan trọng là phải hiểu index, nested routes và dynamic routes. Cùng mình tìm hiểu các khái niệm đó dưới đây nhé.

* Index

+ Các tệp như index.js được chuyển đến root point của ứng dụng của bạn /, thay vì /index. Bạn có thể sử dụng điều này để làm lợi thế của mình bằng cách tạo nhiều tệp chỉ mục hoạt động như trang đích hoặc điểm bắt đầu của các đường dẫn điều hướng khác nhau trong trang web của bạn. Ví dụ như:

- pages

- index.js # found at `/`

- users

- index.js # found at `/users`

- account.js # `/users/account`

+ Ở ví dụ trên, trang index.js bên dưới các trang là trang chủ của trang web được truy cập nếu không có route bổ sung nào được nhập. Index.js thứ hai dưới users là trang đích cho đường dẫn user, được bằng cách nhập <siteName> / users.

* Nested

+ Nested routes là các route chỉ có thể truy cập thông qua một route mẹ được chia sẻ, chẳng hạn như /user/account. Bạn có thể nghĩ về các route lồng nhau giống như các tệp lồng nhau trên máy tính của bạn, trong đó bạn phải điều hướng qua tất cả các component cao hơn để đến được component lồng nhau.

* Dynamic Routes

+ Chức năng route trong NextJS cũng giúp chúng ta có thể tạo ra các trang web với đường dẫn động. Chẳng hạn như một trang web có chức năng detail của user khi click vào một user trên list user thì chúng ta sẽ định nghĩa file như sau:

- users

- index.js

- [account.js] # `/users/[accountName]`

+ Lúc này chúng ta đã có thể tạo ra một trang web detail của 1 user nào đó và có thể truy cập chúng thông qua URL: /users/[accountName]

* Link

+Link là thành phần được Next phát triển để cho phép người dùng có thể chuyến hướng giữa các route với nhau. Ví dụ: hãy xem xét một thư mục trang có các tệp sau:

pages/index.js

pages/about.js

pages/blog/[slug].js

2.2 Spring boot

2.2.1 Giới thiệu

-**Spring Boot** là một Project nằm trên tầng **IO Execution** (Tầng thực thi) của **Spring IO Framework**.  
  
**Spring Boot** là một bước tiếp theo của **Spring**, để làm cho **Spring** dễ dàng hơn trong việc thiết lập và phát triển ứng dụng. Với **Spring Boot** các cấu hình **Spring** được giảm thiểu tối đa. **Spring Boot** hỗ trợ các bộ chứa nhúng (embedded containers) điều này cho phép các ứng dụng web có thể chạy độc lập mà không cần phải trên khai lên các **Web Server**.  
  
Bạn có thể dùng spring boot để tạo ứng dụng Java Web chạy bằng command line **'java -jar'** hoặc xuất khẩu ra file war để triển khai lên **Web Server** như thông thường. **Spring Boot** cung cấp cho bạn một **"CLI Tool"** để chạy các kịch bản Spring (spring scripts).

- Spring Boot chính là một Java framework siêu to và khổng lồ và có nhiều khả năng hữu ích vì nó có thể giúp lập trình viên giải quyết rất nhiều vấn đề. So với framework Spring thông thường, Spring Boot tỏ ra những lợi thế vượt trội. Khi sử dụng Spring Boot, rất nhiều thứ được cải tiến hỗ trợ lập trình viên như:

* Auto config: tự động cấu hình thay lập trình viên, bạn chỉ cần viết code và tiến hành chạy hệ thống là được.
* Dựa trên các Annotation để tạo lập các bean thay vì XML.
* Server Tomcat có thể được nhúng ngay trong file JAR build ra và có thể chạy ở bất kì đâu mà java chạy được.

Khi sử dụng Spring Boot, lập trình viên chỉ cần:

* Sử dụng Spring Initializr: nhập các thông tin của dự án (project), chọn thư viện (Library) rồi tải code về máy.
* Mở mã nguồn (source code) và bắt đầu viết code.
* Có thể chạy ngay trong IDE, hoặc build thành file JAR mà không cần cấu hình config cho server nữa.



2.2.2. Tại sao nên học springboot?

- Việc ứng dụng Spring Boot giúp các lập trình viên tập trung tốt hơn đến mục tiêu viết code của mình thay vì phải lo nghĩ nhiều đến các kỹ thuật code như với [Node.js](https://itnavi.com.vn/blog/nodejs-la-gi/" \t "_blank).Bản thân Spring Boot chứa đựng nhiều các library bên trong, cấu trúc code cũng trở nên chuẩn mực hơn, bạn không cần phải quá bận tâm đến việc code làm sao cho thật tốt.

Do đó, bạn có nhiều thời gian hơn để tập trung vào sự logic của sản phẩm.Spring Boot là sự cải tiến tuyệt vời của Spring, cải thiện được rất nhiều khâu rườm rà đến từ Spring. Hơn thế, quá trình học hỏi và tiếp thu Spring Boot cũng trở nên dễ dàng hơn rất nhiều.

Đừng cố giữ những quan điểm cổ hủ cho rằng muốn biết về Spring Boot trước tiên phải học Spring. Thực chất đây là điều thực sự không cần thiết, bạn hoàn toàn có thể lựa chọn học Spring Boot ngay từ đầu vì nó giúp bạn hiểu và ứng dụng các vấn đề nhanh hơn rất nhiều.

2.2.3. Đặc điểm

- Những tính năng nổi bật của Spring Boot mà bạn không thể bỏ qua như:

* Phát triển web một cách đơn giản và nhanh chóng.
* Config an toàn
* Có thể hỗ trợ YAML
* Có tính quản trị cao
* Dễ dàng ứng dụng Spring và các sự kiện.
* Có thể cấu hình ở bên ngoài và tạo ra những tệp thuộc tính.
* Tính bảo mật cao.
* Ghi log
* SpringApplication: Đây là class được dùng để khởi chạy ứng dụng từ hàm main(). Vì thế, khi khởi chạy ứng dụng, các lập trình viên chỉ cần gọi method run() là được.
* Externalized Configuration: Spring Boot cho phép người dùng có thể sử dụng cấu hình (config) từ bên ngoài. Chính vì vậy mà mỗi ứng dụng được tạo ra có thể chạy được trên nhiều môi trường khác nhau.
* Profiles: Dùng để phân chia các loại cấu hình cho các môi trường khác nhau.
* Loggin: Sử dụng phục vụ cho toàn bộ chức năng log trong phạm vi nội bộ. Những logging này sẽ được quản lý một cách mặc định.