Chapter 1: Introduction Nuxt: (slide 1 - 2)

* Slide 1:

Nuxt là gì ?

* + Nuxt là một framework JavaScript mã nguồn mở dựa trên Vue.js. Phiên bản mới nhất hiện tại là Nuxt 3. Nuxt 3 là bản nâng cấp lại từ framework Nuxt dựa trên Vite, Vue 3 và Nitro với sự hỗ trợ từ Typescript.
  + Nuxt giúp đơn giản hóa việc triển khai website theo phương pháp SSR và STG, ngoài ra còn có CSR, ISR, ESR và SWR. Nuxt cung cấp cách tiếp cận có cấu trúc và dựa trên quy ước xây dựng ứng dụng của VueJS, cho phép developer chỉ tập trung develop tính năng mà không phải config phức tạp.

Thêm:  
**SSR (Server-Side Rendering): Kết xuất trang trên server giúp cải thiện SEO và tốc độ tải trang.**

* + Bạn có thể nói: "Một trong những điểm mạnh của Nuxt 3 là khả năng kết xuất phía server, hay SSR. Điều này có nghĩa là trang web của chúng ta được render trên server trước khi gửi tới người dùng, giúp cải thiện tốc độ tải trang và tối ưu hóa cho các công cụ tìm kiếm, đặc biệt là SEO."
* **SSG (Static Site Generation): Tạo ra các trang web tĩnh nhanh chóng và bảo mật.**
  + Bạn có thể nói: "Nuxt 3 cũng hỗ trợ tạo ra các trang web tĩnh thông qua SSG. Điều này rất hữu ích cho việc xây dựng các trang web nhanh chóng, bảo mật và dễ dàng triển khai trên các nền tảng tĩnh."

Chapter 2: Advantages and Features of Nuxt(slide 3 - 4):

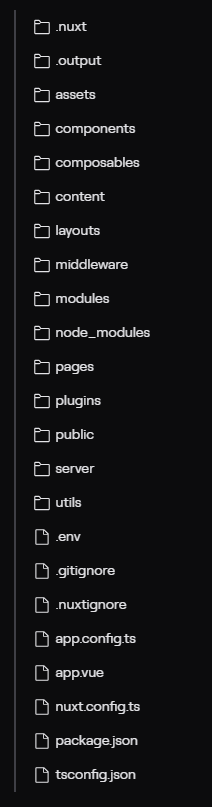
* Lợi ích của Nuxt:
  + **Hiệu suất cao:** Cải thiện tốc độ tải trang và SEO với SSR.
    - **Nuxt được xây dựng dựa trên phương pháp SSR:**
      * Faster initial page load time(Introduction Nuxt web đọc thêm)
      * Improved SEO:
      * Better performance on low-powered devices
      * Better accessibility
      * Easier caching

🡪 Overall, server-side rendering can provide a faster and more efficient user experience, as well as improve search engine optimization and accessibility.

* + **Tối ưu hóa tự động:** Tối ưu hóa mã nguồn, bundle splitting và lazy loading.
    - Nuxt 3 tự động tối ưu hóa mã nguồn bằng cách phân chia bundle và sử dụng lazy loading. Điều này đảm bảo rằng chỉ những phần cần thiết của trang web mới được tải khi người dùng cần, giúp giảm thời gian tải và tăng hiệu suất tổng thể.
  + **Trải nghiệm phát triển tuyệt vời:** Hot module replacement (HMR), cấu trúc thư mục rõ ràng và hỗ trợ TypeScript.
    - Nuxt 3 cho phép chúng ta thấy ngay các thay đổi trong mã nguồn mà không cần phải tải lại trang
    - Cấu trúc thư mục rõ ràng và hỗ trợ TypeScript cũng giúp quá trình phát triển trở nên dễ dàng và hiệu quả hơn
  + **Cộng đồng và tài liệu phong phú:** Hỗ trợ từ cộng đồng và tài liệu phong phú giúp dễ dàng học và phát triển.
* Feature Nuxt3:
  + File-based routing:
    - define routes based on the structure of your pages/ directory. This can make it easier to organize your application and avoid the need for manual route configuration
  + Code splitting
    - Nuxt automatically splits your code into smaller chunks, which can help reduce the initial load time of your application
  + Server-side rendering out of the box:
    - Nuxt comes with built-in SSR capabilities, so you don't have to set up a separate server yourself.
  + Auto-imports
    - write Vue composables and components in their respective directories and use them without having to import them with the benefits of tree-shaking and optimized JS bundles
  + Data-fetching utilities
    - Nuxt provides composables to handle SSR-compatible data fetching as well as different strategies.
  + Zero-config TypeScript support
    - write type-safe code without having to learn TypeScript with our auto-generated types and tsconfig.json
  + Configured build tools
    - we use Vite by default to support hot module replacement (HMR) in development and bundling your code for production with best-practices baked-in.

Chapter 3: Structure folder Nuxt: (Slide 5 - 6)

* Cấu trúc thư mục mẫu:



* Cái mà sẽ đụng và gặp phải nhiều nhất đó là:
  + **assets:** Thư mục **“assets/”** được sử dụng để thêm tất cả nội dung của trang web mà công cụ xây dựng (webpack hoặc Vite) sẽ xử lý. Thư mục thường chứa các loại file sau: Stylesheets (CSS, SASS, v/v.), Fonts, hình ảnh sẽ không được phục vụ từ thư mục **“public/”**.
  + **components:** Thư mục **“components/”** là nơi bạn đặt tất cả các Vue components của mình, sau đó có thể import vào bên trong các trang của bạn hoặc các components khác.
  + **composables:** Nuxt 3 sử dụng thư mục **“composables/”** để tự động import các Vue composables của bạn vào ứng dụng của bạn bằng cách sử dụng tính năng auto-imports.
  + **layouts:** Được sử dụng để thay đổi giao diện ứng dụng của bạn. Một ứng dụng có thể có nhiều layouts ví dụ: admin layout, guest layout và registered clients layout. Các bố cục này sẽ được sử dụng lại trong các trang khác nhau để xử lý giao diện của chúng (sidebar, menu, footer, v.v.). Trong khi cài đặt, Nuxt CLI cung cấp mặc định **layouts/default.vue** layout và được sử dụng trong tất cả các trang.
  + **middleware:** Được sử dụng để xử lý các yêu cầu trước khi trang được hiển thị cho người dùng. Middleware cho phép bạn thực hiện các tác vụ như xác thực, kiểm tra quyền truy cập, ghi lại hoạt động và các xử lý tùy chỉnh khác trước khi trang được hiển thị.
  + **pages:** Nuxt cung cấp routing dựa trên file để tạo các routes trong ứng dụng web của bạn bằng Vue Router.
  + **plugins:** Nuxt tự động đọc các file trong thư mục **“plugins/”** của bạn và tải chúng khi tạo ứng dụng Vue. Bạn có thể sử dụng hậu tố .server hoặc .client trong tên file để chỉ tải plugin ở server hoặc client.
  + **stores: Đây sẽ là thư mục khởi tạo ra store để quản lý state.**
  + **app.vue:** Component chính trong các ứng dụng Nuxt 3.
  + **nuxt.config.ts:** File nuxt.config.ts chứa cấu hình tùy chỉnh Nuxt của bạn và cho phép bạn cấu hình ứng dụng của mình, những cấu hình này bao gồm head title và associated styles và scripts, middlewares, plugins, authentication, modules và thậm chí cả các API.
  + **package.json:** Chứa tất cả các dependencies và scripts cho ứng dụng của bạn.
  + **tsconfig.json:** Nuxt tự động tạo file .nuxt/tsconfig.json với các giá trị mặc định hợp lý khác. Bạn chỉ cần việc tạo 1 file tsconfig.json bên ngoài root và import từ file Nuxt đã khởi tạo.

Chapter 4: Apply and expand (slide 7 -8)

* Apply:
  + Nuxt sẽ được tạo thông qua lệnh: npx nuxi@latest init <project-name>
  + Hình ảnh sau khi chạy lệnh
* Expand:
  + Modules:
    - Dễ dàng tích hợp và có rất nhiều modules nhưng ở đây sẽ giới thiệu về 2 modules được ứng dụng trong bài:
      * Pinia:
        + Pinia là một thư viện quản lý trạng thái mới cho Vue 3. Được thiết kế để thay thế Vuex, Pinia cung cấp một API đơn giản hơn, hiệu suất cao hơn và tích hợp tốt hơn với Vue 3
        + Đưa hình ảnh về cấu hình Pinia
      * Axios:
        + Axios là một thư viện HTTP client mạnh mẽ và linh hoạt, dựa trên Promise, giúp chúng ta dễ dàng thực hiện các yêu cầu HTTP từ trình duyệt và Node.js
        + Đưa hình ảnh về Axios

Chapter 5: Demo:

* Đưa hình ảnh về bài làm