TÀI LIỆU KHÓA HỌC “React Ultimate với Javascript”

Tác giả: Hỏi Dân IT & Eric

Version: 1.0

Note cập nhật:

- Update tài liệu

Chapter 1: Bắt buộc xem
Hướng dẫn sử dụng khóa học hiệu quả

#1. Hướng Dẫn Sử Dụng Khóa Học Hiệu Quả

Bạn vui lòng "xem video lần lượt" theo trình tự. Vì khóa học như 1 dòng chảy, video sau sẽ kế thừa lại kết quả của video trước đó.

1. Dành cho học viên "có ít thời gian"
Nếu bạn vội, cần học nhanh, hoặc "bạn đã biết rồi", thì "vẫn xem video, cơ mà không cần code theo".

Lưu ý: vẫn xem qua tài liệu khóa học để biết "video hướng dẫn gì".
Đã "Không xem video", thì cần "đọc giáo án".

Có như vậy mới biết khóa học nó làm cái gì.

2. Dành cho học viên "thông thường"
Nguyên tắc:
- Xem video lần lượt
- Xem video kết hợp với giáo án. Bạn không cần take note, vì những điều quan trọng đã có trong giáo án

- **Bạn vui lòng code theo video.**
Nếu bạn "code theo ý bạn", vui lòng "không hỏi khi có bugs".

Câu chuyện này giống như việc bạn đi khám bệnh, nhưng không tin lời bác sĩ
=> Nếu bạn giỏi, bạn làm luôn bác sĩ, còn đi khám bệnh làm gì.

- Bạn có thể "code theo ý bạn muốn", sau khi "đã kết thúc khóa học"

- Nếu bạn có thắc mắc (hoặc có ý tưởng/nhận thấy bugs), take note lại, bạn hỏi, rồi mình giải đáp.
Chứ không phải là "tự ý làm theo điều các bạn muốn".
Vì đa phần, các bugs trong khóa học mình đã fix hết rồi.

Nên là yên tâm để học theo bạn nhé.

3. Về cách code.

Bạn vui lòng code theo video, từ cách đặt tên biến, hàm. Vì mình đã tuân theo "convention tối thiểu" khi bạn đi làm đấy

4. Về bài tập thực hành

Đối với bài tập thực hành, bạn cứ code theo cách bạn hiểu, và kết hợp với "search Google, stackoverflow..."
KHÔNG DÙNG CHATGPT. Đây giống kiểu chưa học "phép tính", mà đã "dùng máy tính".

Nên nhớ 1 điều, trước 2023, không có chat gpt, thì mình học như thế nào ?
Khi bạn đã đi làm, bạn có quyền dùng cái gì bạn thích, còn với beginner, hãy biết say NO với CHAT GPT.

tương tự bạn dạy con bạn:
học lớp cấp 1: không chịu học tính nhầm => đưa luôn máy tính cho nó. Rời máy tính ra, là không biết làm phép tính

còn với học sinh cấp 2, 3 : dùng máy tính tùy thích

#2. Tài liệu khóa học

//todo

#3. Demo kết quả đạt được
Video demo: https://www.youtube.com/watch?v=PlJ2NDWy2zg

1. Công nghệ sử dụng
React version 18 & 19
React là thư viện với cơ chế CSR - client side rendering

Các kiến thức trọng tâm:

- Phân biệt các phong cách code/sử dụng React trong thực tế
- Học React với đúng tư duy ban đầu của React - React là library UI
- Các kiến thức về React (cốt lõi nhất):
    + Tư duy thiết kế UI với React (sử dụng Component)
    + Render/Re-render giao diện với Props và State (useState hook)
    + Điều hướng trang với React-router-dom
    + Sử dụng useEffect để gọi API backend
    + Sử dụng Context API để sharing data giữa các component
    + Sử dụng mô hình Stateless (với access_token)
    + Sử dụng Ant Design (antd) để làm giao diện chuyên nghiệp (UI - UX)
    + Tối ưu hóa re-render với Uncontrolled Component

Backend (Nestjs) được mình cung cấp sẵn. Chỉ sử dụng và không sửa đổi. (**không học code backend trong khóa học này**). Quan tâm về backend, tham khảo tại đây

Database MongoDB sử dụng online (miễn phí) với MongoDB Atlas

Lưu ý: khóa học này mình sử dụng React với JavaScript (để giảm độ khó và phù hợp với nhiều level của beginner)

- Nếu bạn muốn sử dụng React với Typescript (yêu cầu đã biết về React), tham khảo tại đây

- Nếu bạn muốn sử dụng React với Typescript và sử dụng framework Next.js, tham khảo tại đây

2. Học viên nào có thể học ?

Học viên cần trang bị các kiến thức sau trước khi theo học: **HTML, CSS và cú pháp của Javascript**

3. Triển khai dự án

Đến cuối khóa học, dự án được triển khai:

- Frontend triển khai với Vercel
- Backend triển khai với Render
- Database triển khai với MongoDB Atlas

#4. Yêu cầu để học được khóa học này

1. Biết HTML, CSS và cú pháp Javascript (tự học)

Về HTML, CSS => tự học

Về Javascript, tham khảo: (chỉ cần học để hiểu cú pháp khai báo của javascript)
https://www.w3schools.com/js/

Xem khóa học Javascript miễn phí tại đây

2. Biết sử dụng Git để quản lý mã nguồn

Xem khóa học Git miễn phí tại đây
Khóa học Git trả phí, tham khảo tại đây

#5. Cách Dùng Udemy - Hỗ Trợ Hỏi Đáp Q&A

Lưu ý: không bỏ qua video này. Xem để biết cách sử dụng Udemy, cũng như các đặt Q/A khi cần hỗ trợ (support)

1. Sử dụng trên máy tính

Xem hướng dẫn tài liệu chi tiết tại đây

- Cách bắt đầu sử dụng khóa học (bắt buộc xem)

- Cách đặt câu hỏi cho khóa học (bắt buộc xem)
    - Hướng dẫn cách sử dụng Q&A
    - Hướng dẫn cách liên hệ Instructor qua Message

- Cách sử dụng phím tắt

- Take note trực tiếp trên video đang xem

2. Sử dụng trên điện thoại

Udemy có hỗ trợ ứng dụng trên điện thoại Android/IOS

Xem hướng dẫn tài liệu chi tiết tại đây

#6. Về Tác giả

Về tác giả:

Mọi thông tin về Tác giả Hỏi Dân IT, các bạn có thể tìm kiếm tại đây:

Website chính thức: https://hoidanit.vn/

Youtube “Hỏi Dân IT” : https://www.youtube.com/@hoidanit

Tiktok “Hỏi Dân IT” : https://www.tiktok.com/@hoidanit

Fanpage “Hỏi Dân IT” : https://www.facebook.com/askITwithERIC/

Udemy Hỏi Dân IT: https://www.udemy.com/user/eric-7039/

Nếu bạn muốn nói chuyện với mình (giao lưu trao đổi võ công :v), có thể xem mình livestream trực tiếp tối thứ 2 & thứ 5 hàng tuần trên Youtube Hỏi Dân IT

Chapter 2: Setup Environment
Cài đặt & chuẩn bị môi trường thực hiện dự án

#7. Cài đặt NodeJS
Tài liệu: https://nodejs.org/en
1.Nodejs là gì ?
Nodejs không phải là thư viện (library), không phải framework của javascript.

Nodejs là môi trường để bạn thực thi code javascript, tại browser và server.

Bạn học React (viết bằng Javascript), nên bạn cần Nodejs để có thể chạy được nó (code javascript)

Điều này tương tự với:

Bạn học cách sử dụng Microsoft Excel (react)
Bạn cần cài hệ điều hành Windows để có thể học nó (nodejs)

2. Cài đặt Nodejs
Sai lầm của beginners, là không quan tới tới version của phần mềm. Nên nhớ, công nghệ nó thay đổi theo thời gian, vì vậy, để hạn chế tối đa lỗi tối đa, bạn nên dùng version phần mềm như khóa học hướng dẫn.

Điều này tương tự với:
Bạn đang chơi 1 con game rất ngon trên Windows 7, bạn vác lên Windows 10 để chạy, có điều gì để đảm bảo rằng “sẽ không có lỗi xảy ra” ?

Trong khóa học này, mình sử dụng **version Node.js là 20.14.0.**

Vì vậy, để hạn chế tối đa lỗi có thể xảy ra, bạn vui lòng cài đặt chính xác version nodejs ở trên

Khi code giống nhau, môi trường thực thi code giống nhau (version nodejs), thì rất hiếm khi lỗi xảy ra.

Link tải nodejs v20.14.0:
https://nodejs.org/download/release/v20.14.0/

Sau khi cài đặt xong, kiểm tra bằng cách gõ câu lệnh:

node -v

3. Trường hợp dùng nhiều version Nodejs
//áp dụng cho windows
https://github.com/coreybutler/nvm-windows

Video hướng dẫn cài nvm cho window, xem tại đây

//áp dụng cho macos
Video hướng dẫn cài nvm cho mac, xem tai đây

https://dev.to/ajeetraina/how-to-install-and-configure-nvm-on-mac-os-5fgi

#8. Cài đặt Visual Studio Code

Công cụ code trong dự án sử dụng VSCode, 1 IDE hoàn toàn miễn phí

Link download:
https://code.visualstudio.com/download

#9. Cấu hình Visual Studio Code

1. Format Code

Setup Format on Save

Mục đích: Mỗi lần nhấn Ctrl + S , code sẽ được auto format trông cho đẹp/dễ nhìn

2. Cài đặt Extensions

Lưu ý: off các extension như eslint, prettier … để tránh xung đột

Fact: đi làm, người ta cấu hình eslint, prettier..thông qua code, vì mỗi 1 dự án (1 khách hàng 1 yêu cầu), cài global qua extension thì cái nào cũng giống cái nào

Đồng thời, với rule trên sẽ đảm bảo mọi thành viên trong team sẽ có cấu hình giống nhau

Các extensions cài đặt thêm:

- Code Spell Checker : hỗ trợ check chính tả khi đặt tên tiếng anh
- Auto Complete Tag : hỗ trợ code nhanh HTML

#10. Tại sao mình dùng VScode ?

Có rất nhiều IDE hỗ trợ bạn code React (Javascript):

- Visual Studio Code (gọi tắt là VSCode)
- WebStorm
- Sublime Text
- Notepad, Notepad ++ 😂

IDE thực chất là công cụ code, giúp gợi ý code và phát hiện lỗi
=> dùng công cụ code nào mà bạn "thoải mái nhất"

Mình chọn VScode vì:
- miễn phí (nếu bạn code Frontend thì chắc chắn bạn sẽ thích)
- có gợi ý code và phát hiện lỗi
- theme dark :v
- hỗ trợ git out-of-the-box
- support mạnh mẽ cho "frontend" => tức là 1 IDE cho cả frontend/backend

Trong khóa học này, chỉ cần bạn code giống mình, dùng IDE nào không quan trọng, điều quan trọng, chính là cách chúng ta code ra làm sao :v

Recommend: dùng VScode, để đảm bảo bạn và mình giống nhau 100%, từ coding cho tới debug :v

Yên tâm 1 điều là: điều quan trọng nhất chính là khả năng bạn “tư duy” (mindset), bạn có thể dùng những điều học được, để áp dụng sang IDE bạn thích.

Fact: mình đã từng rơi vào công ty (khá to), cơ mà không mua license bản quyền phần mềm (trong khi không cho dùng crack)
=> bắt buộc phải dùng các công cụ free @@

#11. Cài đặt và sử dụng Git

- Nếu bạn chưa biết gì về Git, xem nhanh tại đây

- Sử dụng Git theo nguyên tắc:

1. Học xong video nào, commit đẩy lên Github/Gitlab
=> tạo cơ hội để thực hành câu lệnh của Git, ví dụ:
git add
git commit
git push...

2. Git là công cụ "mặc định bạn phải biết" khi đi làm phần mềm
=> điều 1 ở trên giúp bạn thực hành

3. Thói quen học xong video nào, đẩy code lên Git, giúp bạn tạo ra bản "backup" cho project của bạn

Ví dụ máy tính bạn bị hỏng đột xuất/bị mất
=> vẫn còn code, chỉ cần pull về code tiếp mà không phải code từ đầu.

Nên nhớ, khóa học này mình không share source code => chỉ bạn giúp được bạn

4. Trong trường hợp bạn bị bug
=> bạn có thể gửi link github/gitlab cho mình xem => support fix bug

=> Mục đích sử dụng git ở đây là : backup code + thực hành công cụ đi làm mà bạn "phải biết" nếu muốn đi thực tập/đi làm.

#12. Cài đặt Google Chrome

Do chúng ta code website, thành ra cần sử dụng browser để test kết quả.

Ở đây, sử dụng google chrome vì nó là ứng dụng phổ biến nhất (trình duyệt web được dùng nhiều nhất)

Bạn nên dùng Google Chrome (thay vì Firefox/Edge…) để đảm bảo rằng thao tác sử dụng giữa bạn và mình là giống nhau (tránh gây khó khăn không cần thiết)

Lưu ý: sử dụng version tiếng anh
=> change language

Mục tiêu:

- Sử dụng Google Chrome để chạy ứng dụng web
- Ngôn ngữ hiển thị là Tiếng Anh
- Set default app là google chrome (nếu nó mở app, thì chạy với google chrome)

  Xem hướng dẫn setup default app cho windows tai đây

Chapter 3: Lịch Sử Phát Triển của React (tính tới React 19)
Làm quen với quá trình phát triển của thư viện React

#13. React là gì ? Tại sao lại học React ?

1. Khái niệm “React” ?
https://github.com/facebook/react

Châm ngôn của React : Learn once, write anywhere.

Có nghĩa là: bạn tốn thời gian học “tư duy của React” đúng 1 lần, bạn có thể áp dụng nó ở nhiều nơi.

Ví dụ:
React JS (hay gọi tắt là React) dùng để code website

React Native dùng để code hybrid mobile app (android/ios)

Electron : desktop app

2. Lý do bạn học React

React được Facebook chống lưng (sau này là Meta), vì vậy, câu chuyện bảo React sẽ die là rất khó (vì ngay cả Facebook/Instagram…) có rất nhiều tính năng được code nên bởi React

React được sử dụng rộng rãi (phổ biến), tài liệu, source code đa dạng, phong phú

=> cái gì phổ biến thì chúng ta học thôi :v

#14. Phân biệt React, Angular, Vue

Để xây dựng frontend website hiện đại, chúng ta có 3 đồng chí nổi tiếng nhất:
1.Angular
https://angular.dev/

- Được chống lưng bởi Google
- Phát triển từ 2010 (version angluarJS), sau này là angular 2 (2016)
- Là **framework**

2. Vue

https://vuejs.org/

- Được chống lưng bởi Evan You và các công ty tài trợ (đa phần là Trung Quốc)
- Phát triển từ 2014
- Là **framework**

3. React

- Được chống lưng bởi Meta (facebook)
- Phát triển từ 2013
- Là **library**

4. So sánh React, Angular, Vue

https://npmtrends.com/@angular/core-vs-react-vs-vue

5. Nên chọn công nghệ nào ?

Vue là con lai giữa React và Angular

#15. Cơ hội việc làm của React

Để đánh giá nhu cầu việc làm của 1 công nghệ, bạn cần dành thời gian để tìm hiểu và nghiên cứu (R&D - research and development)

Tùy từng thời điểm (thời kỳ trong năm), tùy từng chức vụ tuyển dụng, tùy từng vị trí (nơi bạn sinh sống)... sẽ cho ra các kết quả khác nhau.

Bonus: (câu chuyện 1m2 bao nhiêu ông React)

Tham khảo review CV: https://youtube.com/live/xy67XFdAR8Q

#16. Lịch sử phát triển của React - Chúng ta đang ở đâu ?

1.Giai đoạn phát triển

Được chia làm 2 giai đoạn chính:

Giai đoạn 1: từ khi ra đời (2013) tới tháng 4/2022
Trước 2013 trở về trước, chúng ta có cơ chế SSR (server side rendering) và Jquery chính là vua frontend. Jquery + Wordpress là công thức làm nhanh một website.

React/Angular/Vue ra đời, bổ sung cơ chế CSR (client side rendering) và khái niệm SPA (single page application)

React là thư viện. Có nghĩa là, nó đơn giản nhất có thể. Bạn muốn hơn, cần phải tự làm. Ví dụ như router, caching data..

Giai đoạn 2: từ tháng 4/2022 tới nay

React là framework (ví dụ như Next.js).

Website quay lại với SSR (server side rendering) với hình thức SSG (Static side rendering) - React chạy trên server (chứ không phải browser)

2.Chúng ta đứng ở đâu

Với khóa học này, mình sử dụng React như 1 thư viện UI (user interface) để làm giao diện thuần túy. Sử dụng React với cơ chế CSR (client side rendering)

Mình cố gắng giữ mọi thứ đơn giản nhất có thể, như tuy duy ban đầu của React.

Với giai đoạn 2 của React, chính là khóa học level nâng cao hơn:
Khóa React Portfolio, xem tại đây

Khóa React Pro Max với Nextjs, xem tại đây

#17. Có bao nhiêu cách để code React

Có 2 cách chính để code React hiện nay:

Cách 1 (cách khóa học sử dụng), là sử dụng React với cơ chế CSR (client side rendering)

Các ứng dụng phổ biến:

- Các website có nội dung thay đổi liên tục (real-time), ví dụ bạn chat Messenger trên Facebook
  Bảng giá vndirect
  Sàn binance

Ưu điểm:

- Code thuần túy react (đúng ý tưởng ban đầu) - đơn giản nhất có thể
- React là thư viện UI

Nhược điểm:

- SEO không tốt (ví dụ cho Google Search Engine)
- Trải nghiệm của user thấp (trong trường hợp mạng internet chậm)

Cách 2: sử dụng React với cơ chế SSR (Server side rendering)

Các ứng dụng phổ biến:

- Các website cần SEO, đọc tin tức, ví dụ: https://react.dev/
- Các website có độ chịu tải cao & quan trọng hóa trải nghiệm của người dùng (UX), ví dụ: https://tiki.vn/

Ưu điểm:

- React là framework làm UI
- Hỗ trợ SEO và tăng trải nghiệm của người dùng (UX)

Nhược điểm:

- Code phức tạp, vì bạn đang dùng framework

#18. Nên code React với Javascript hay Typescript

Typescript = Javascript + khai báo type

Với Angular, bạn "bắt buộc" phải sử dụng Typescript.

Với Vue và React, bạn đều có thể dùng Javascript/Typescript, tùy sở thích của bạn.

Ưu điểm khi dùng Typescript:

- Hạn chế bugs và lỗi cú pháp (vì check type)
- Khối lượng code base càng lớn (số lượng dòng code), càng dễ sử dụng

Nhược điểm: code dài dòng hơn javascript, vì cần khai báo type

Ưu điểm khi dùng Javascript:

- Code nhanh nhất có thể (gần giống như code cho chạy được)

Nhược điểm: do quá thoải mái (không bị ràng buộc về cú pháp), rất khó để debug/ sửa đổi code khi khối lượng code base lớn

Khóa học này, mình sử dụng Javascript:

- Cú pháp đơn giản. Đồng thời, dự án chúng ta học chỉ có 1 người code và khối lượng dòng code không quá nhiều.
- Dễ dàng tiếp cận với mọi level của học viên, chỉ cần biết cú pháp của Javascript là học được React

Về typescript với React, mình hướng dẫn tại, mình hướng dẫn tại các khóa học tiếp theo, ví dụ:
Khóa React Portfolio, xem tại đây

Khóa React Pro Max với Nextjs, xem tại đây

#19. Tìm tài liệu về React ở đâu

Trang tài liệu mới nhất của React:
https://react.dev/

Trang tài liệu cũ của React (không dùng nữa), chỉ nên sử dụng với React < 18
https://legacy.reactjs.org/

Github của react:
https://github.com/facebook/react

Chapter 4: Hello World với React
Viết chương trình hello world với ứng dụng Reac

#20. Setup dự án thực hành

1.Chuẩn bị
Đảm bảo rằng bạn đã cài đặt Git và Node.js **(version 20.14.0)**
Chưa biết dùng git, học ngay và luôn tại đây

Bạn vui lòng sử dụng chính xác nodejs v20.14.0 để hạn chế tối đa lỗi xảy ra
(mình đã giải thích tại video #7)

2.Cài đặt dự án thực hành
Bước 1: Download source code tại đây

Bước 2: cài đặt thư viện cần thiết
npm i

Lưu ý về warning/error tại terminal

Bước 3: chạy dự án
npm run dev

Truy cập: http://localhost:3000/

3. Có bao nhiêu cách để setup 1 dự án với React

- Create-react-app: https://create-react-app.dev/
- Vite: https://vitejs.dev/guide/
- Sử dụng framework: Nextjs, Gasby, Remix

Khóa học này, mình không sử dụng framework, để giữ mọi thứ đơn giản nhất có thể - hãy học React là 1 thư viện (library), trước khi sử dụng nó như là 1 framework

#21. Cách mình setup dự án (Extra)

Lưu ý, bạn vui lòng không làm theo video này.

Mục đích mình làm video này, để thỏa mãn tính tò mò của nhiều bạn.

Bạn chỉ làm như video này, khi đã có khả năng tự fix bug và đọc tài liệu, cũng như đã học xong khóa học này rồi.

Bạn vui lòng thực hành khóa học bằng cái tải source code mình cung cấp tại #20.

Nếu như bạn tự ý code “theo cách bạn hiểu” và không sử dụng source code tại #20, mình sẽ không hỗ trợ support khi có lỗi xảy ra

#22. Hello World với React

1.Viết chương trình Hello World

Chạy dự án với câu lệnh : **npm run dev**

Truy cập: http://localhost:3000/

Bonus: cách set default app của windows

2. Cách backup source code với github
Đảm bảo rằng bạn đã có kiến thức cơ bản về Git, nếu chưa có, xem tại đây

Bước 1: tạo git repository

git remote set-url origin new.git.url/here

Bước 2: sử dụng lần lượt các câu lệnh sau:
git add .
git commit -m "your message"
git set remote …
git push

Warning của git: https://github.com/orgs/community/discussions/66838

Giải thích:
https://shzhangji.com/blog/2022/08/31/configure-git-line-endings-across-oses/

#23. Cấu trúc dự án thực hành

React được sử dụng ở đây như là 1 library => cấu trúc đơn giản nhất có thể

1.Giải thích ý nghĩa của các file cung cấp

Thư mục:
node_modules: lưu trữ thư viện cài đặt

public : lưu image/css/js (những cái muốn public ra ngoài internet, ai ai cũng có thể truy cập được)

src : nơi dev viết code (99%)

Các files:
.eslintrc.cjs : cấu hình eslint (giúp check code javascript)

.gitignore : quy định những files nào "không đẩy lên git"

index.html : file mà browser sẽ chạy (với mô hình CSR). hiểu đơn gian, code react sẽ được dịch và "nhét" vào đấy

package.json : quy định thông tin về project, như tên thư viện cài đặt, cách chạy project
package-lock.json : quy định chi tiết thông tin cài đặt thư viện

README.md : cung cấp thông tin về project

vite.config.js : cấu hình dự án React với Vite - trình dịch code

2. Cấu trúc dự án thực tế trông như nào (bonus)
Nếu bạn không sử dụng framework, việc tổ chức cấu trúc (structure) như nào, phụ thuộc vào quan điểm và trình độ của mỗi người
https://dev.to/itswillt/folder-structures-in-react-projects-3dp8

#24. Đặt tên file JS/JSX/TS/TSX cho React ?

js: javascript
Dùng để định nghĩa file code javascript, hoặc code react (vì react là javascript mà)

jsx : javascript + JSX
Chỉ dùng để định nghĩa react (javascript)

ts: typescript
Chỉ dùng để định nghĩa file code typescript

tsx: typescript + tsx
Chỉ dùng để định nghĩa react (typescript)

Bonus: Tại sao có trường hợp khi bạn đặt tên là .js hay .jsx cho React, code đều chạy được ?
Ví dụ: https://codesandbox.io/p/sandbox/create-react-app-iuync?

#25. Cơ chế hoạt động của React với Browser (Extra)

1.Với chế độ dev (development)
Dev coding (ngôn ngữ code)
Compile và lưu trong memory

Sử dụng các câu lệnh:
npm run dev

2. Với chế độ prod (production)
Compile
Lưu tại cache của browser

Sử dụng các câu lệnh:

npm run build

npm run preview

#26. Tại sao gọi React là Client Side Rendering (Bonus)

SSR: server side rendering. Mọi công việc render để tạo ra dữ liệu, được xử lý tại phía server

CSR: client side rendering : công việc render và tạo ra dữ liệu xảy ra tại phía client

Giải thích:
Sau khi đã build ứng dụng, miêu tả cơ chế SSR và CSR

Chapter 5: Tư duy thiết kế UI với Component
Sử dụng React dưới dạng Component để tạo nhanh UI ứng dụng

#27. Khái niệm về Component

1. Tư duy của framework/library
Với tư duy của Frontend, chúng ta sẽ chia giao diện thành các component

Mục đích: tăng tính "tái sử dụng" (reuse code)

Component = HTML + CSS + Javascript

Ví dụ: https://4x.ant.design/components/popover/

2. Giới thiệu về arrow function (bổ trợ)

Tài liệu arrow function, xem tại đây

Ví dụ về arrow function, xem tại đây

Phong cách của javascript hiện đại (thậm chí là java), là sử dụng "functional programing).

Why ? máy tính (chỉ hiểu/và chạy nhanh) khi code của chúng ta đơn giản nhất có thể.

Tất cả code của chúng ta, đều là function, tuy nhiên là "arrow function"

//todo : convert App.jsx thành arrow function

#28. Component

Component là 1 khối code

Component = HTML + CSS + Javascript

Component giúp tái sử dụng code

1.Cách định nghĩa 1 component với React

Tài liệu: https://react.dev/learn/your-first-component#defining-a-component

Bước 1: Định nghĩa một arrow function

```
const MyComponent = ( )  => {   }
```

Lưu ý: tên component bắt buộc viết hoa chữ cái đầu tiên

Viết đúng: **My**Component

Viết sai: **my**Component

Bước 2: Để biến Javascript function trở thành component, chúng ta cần return HTML

```
const MyComponent = ( )  => {
        return ( // todo )
 }
```

Bước 3: Sử dụng Component như là 1 thẻ tag của HTML

#29. Import/Export Component

Tài liệu: https://react.dev/learn/importing-and-exporting-components

Mục đích: Reuse (tái sử dụng) code

Export: xuất ra để nơi khác dùng

Import: gọi tới component

Lưu ý, mặc định là **export default**
https://developer.mozilla.org/en-US/docs/web/javascript/reference/statements/export

#30. JSX

Tài liệu: https://react.dev/learn/writing-markup-with-jsx

JSX cho phép bạn code HTML trong file Javascript

1.Single Root

JSX chỉ có 1 cha duy nhất

2. Fragment

Fragment (mảnh vỡ) giúp bạn viết code ngắn đi, và không render "thừa html"

https://react.dev/reference/react/Fragment

3. Sử dụng CSS

Lưu ý: className

Không dùng từ "class" với JSX, vì class là keyword "lớp" của javascript

Lưu ý về inline style (viết theo quy tắc của object)

#31. Cách sử dụng biến số với JSX

Tài liệu: https://react.dev/learn/javascript-in-jsx-with-curly-braces

Về các loại "data types" của Javascript, tham khảo tại đây

Bao gồm 2 loại chính:

Dữ liệu nguyên thủy: string, number, boolean, undefined, null
Dữ liệu object (array)

Nguyên tắc : sử dụng cặp dấu ngoặc nhọn { } để viết code javascript bên trong html

Về hàm JSON.stringify( )

#32. Nested Component - Quan hệ Cha-Con

Cài đặt react devtool extension
https://chromewebstore.google.com/detail/fmkadmapgofadopljbjfkapdkoienihi?hl=en

Quan hệ cha-con (parent-child)

//todo: giải thích ý nghĩa của file main.tsx

#33. Bài tập Components

Xóa hết layout đang có (bao gồm cả css)

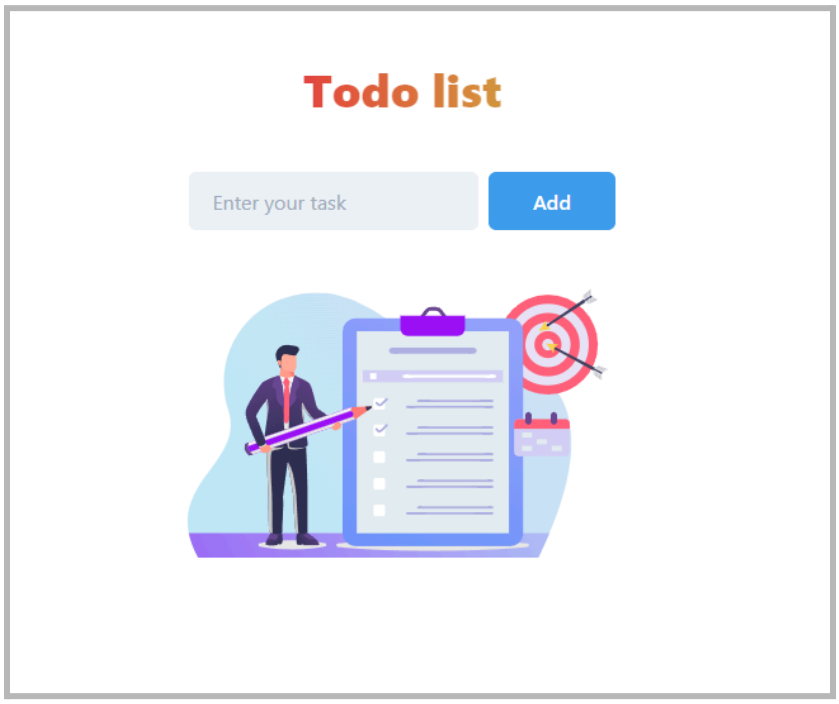
Cách tư duy: vẽ base component (cứ code theo cách bạn hiểu).
Code tất cả mọi thứ trong 1 component, nếu chia được layout thì càng tốt (chia tách parent-child)

Các kiến thức áp dụng:

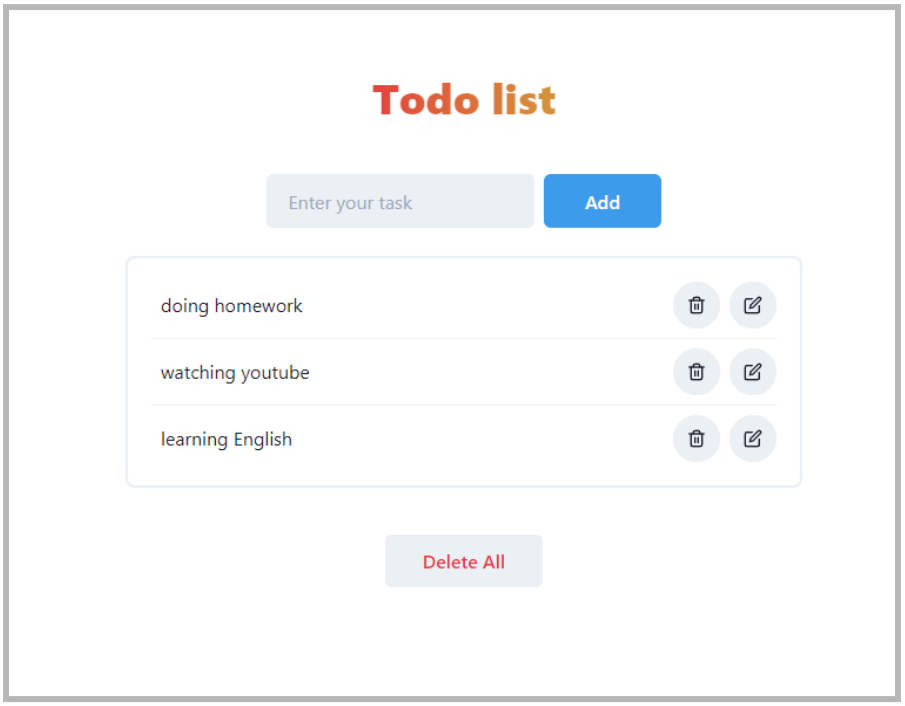
- Tạo component : arrow function và JSX
- Import/Export component
- CSS cho component
- Nested component

Bonus thêm cách sử dụng hình ảnh (logo react)

Khởi tạo Todo List:



Khi có data:



#34. Props

Props = Property (tài sản kế thừa), sử dụng trong mối quan hệ cha con

Component cha, truyền "props" sang **component con**

=> props chính là cách chúng ta truyền "data" giữa các component

1. Cách truyền props

Khai báo props tại component cha

Nhận props tại component con

//check với: number, string, object
//todo: tạo array todo list truyền qua component con

//sử dụng react dev tool để xem trực tiếp data

2. Một vài cách code
//lưu ý về các cách code và destructuring data

Về cú pháp destructuring data, xem tại đây

Cách 1: lấy props trực tiếp từ đầu hàm

Cách 2: sử dụng props với destructuring object

Cách 3: lấy data trực tiếp qua props

#35. Truyền Function từ cha sang con

Ví dụ trường hợp:
Trong table có 1 button, khi nhấn button này sẽ open Modal
https://4x.ant.design/components/modal/

Trong thế giới của JavaScript, data bao gồm biến số và function.

Nếu props cho phép truyền biến số từ cha sang con, liệu có thể là function ?

#36. DOM Events
Tài liệu: https://react.dev/learn/responding-to-events

Danh sách event: https://www.w3schools.com/jsref/dom_obj_event.asp

2 sự kiện hay dùng nhất là **click** và **change**

onClick: sử dụng chuột để click

onChange: gõ giá trị vào input

#37. Kiểm Soát Data với State - useState Hook

Tài liệu: https://react.dev/learn/state-a-components-memory

Mục tiêu: khi nhấn button, có thể lấy được giá trị của input

Đặt state là array trên component cha
Component con sử dụng JSON.stringify

#38. Re-render với State

Javascript random number between range

```
function randomIntFromInterval(min, max) { // min and max included
  return Math.floor(Math.random() * (max - min + 1) + min);
}
```

Mục tiêu: nhấn button Add new => thêm mới data

#39. Render List

Tài liệu: https://react.dev/learn/rendering-lists

1.Render List với map

Giới thiệu về map, tham khảo tại đây

So sánh với for loop:
https://stackoverflow.com/questions/76868163/react-looping-over-map-vs-array

Đối với map, bạn lặp qua từng phần tử và tạo ra 1 array mới
(tức là return một giá trị mới) => rất an toàn để bạn "mutate" data (sửa đổi data)

Đối với vòng lặp for, bạn lặp qua từng phần tử, và "không trả ra array mới"
Về for-each, tham khảo tại đây

//todo: CSS thêm button Delete

#40. Each child in a list should have a unique "key" prop

1. Phân biệt warning và error

Giống nhau: đều có chữ "màu đỏ", màu đỏ "thường" không tốt, right ?

Khác nhau:

- **Error : gần như 99% là bạn cần fix,** vì khi có error, xác suất rất cao ứng dụng sẽ không chạy (trừ khi bạn đã handle nó - try/catch exception)

- **Warning: tỉ lệ cần fix là 50-50**, bạn có thể fix hoặc không. Gọi là "cảnh báo", vì nếu fix được, hiệu năng ứng dụng của bạn sẽ đạt hiệu quả tối đa

2.Tại sao React cần key trong vòng lặp

Khi sử dụng vòng lặp để render, phần tử nào cũng giống phần tử nào.

Vấn đề sẽ phát sinh, nếu chúng ta "mutate" data (bao gồm thêm/sửa/xóa)

Ví dụ: khi bạn thêm mới 1 phần tử vào array sau: (phần tử 4 được thêm mới)

Hoặc, bạn xóa phần tử có giá trị là 2 ra khỏi array ?

 1

 2

 3

React sẽ xử lý như thế nào ?

=> react sử dụng key để định danh phần tử.

Nếu key tồn tại => update /delete

Nếu key không tồn tại => thêm mới

3. Sử dụng key như thế nào cho chuẩn

Mặc định, nếu bạn không dùng key, React sẽ tự động sinh key (sử dụng index của array)

Không nên dùng key là chỉ số của mảng (index), hoặc chuỗi string gắn liền với index

Key phù hợp :

- Sử dụng id lấy từ backend
- Generate id "trước khi vào vòng lặp"

#41. Render với điều kiện

Về toán tử điều kiện của Javascript, tham khảo tại đây

Điều kiện: nếu size = 0, hiển thị hình ảnh
Size > 0, hiển thị danh sách todo list

#42. Bài tập Delete Todo
Gợi ý: sử dụng filter, tham khảo tại đây

Các bước thực hiện:

Bước 1: Viết sự kiện onClick cho button Delete

Mỗi lần nhấn vào button, cần lấy được id của todo cần xóa

Bước 2: Gọi function của component cha
Vì hành động click button xảy ra tại component con, trong khi todoList được component cha quản lý => cần phải truyền function từ cha xuống con

Tại component cha, cần viết function để xóa todo (dựa vào id là input đầu vào)

Bước 3: Xóa todo
Sử dụng hàm filter để xóa
Set State với data đã filter

#43. Tổng kết các kiến thức đã học

1.Kiến thức nền tảng của React
//cài đặt react dev tool (đã làm)

Props: (viết tắt của property - tài sản), là cách chúng ta truyền dữ liệu từ cha sang con

State (trạng thái), được sử dụng qua useState hook

Chỉ cần props/state thay đổi, component sẽ re-render

2.Kiến thức khác

- Cú pháp JSX (dấu { })
- Cách render data (array) với map
- Cách sử dụng câu điều kiện

Cách kiểm tra version của React ?
Tại sao gọi là hook ?
Lịch sử ra đời, react class (react hook) version 16.8

Chapter 6: Điều Hướng Trang Với Router

Tìm hiểu cách tạo nhiều page và điều hướng trang với React Router

#44. Giới Thiệu về Router

React là thư viện (library), nó chỉ chịu trách nhiệm render (vẽ UI) thông qua props/state và JSX

Vì vậy, nếu bạn muốn nhiều hơn, ví dụ như điều hướng trang (router), bạn sẽ cần **"tự làm"**

Nếu React tích hợp sẵn Router, nó sẽ là framework (ví dụ Next.js)

1.Cài đặt thư viện
https://www.npmjs.com/package/react-router-dom

npm i --save-exact react-router-dom@6.23.1

Trang chủ: https://reactrouter.com/en/main

Lưu ý : Bạn vui lòng cài đặt thư viện bằng cách sử dụng câu lệnh ở trên và làm giống như video.

Thư viện router nó sẽ cập nhật theo thời gian, vì vậy, điều quan trọng nhất chính là cách bạn tư duy để giải quyết vấn đề (chính là cái khóa học sẽ hướng dẫn bạn)

#45. Tích Hợp Router

Tài liệu:
https://reactrouter.com/en/main/start/tutorial

Lưu ý: có thể giao diện website sẽ thay đổi theo tương lai, nên điều quan trọng nhất chính là khả năng bạn đọc tài liệu và thực hành (cái mà video hướng dẫn)

Mục tiêu, tạo các page:

- Homepage: /
- /users
- /products
- /login
- /register

Bước 1:

```
import {
  createBrowserRouter,
  RouterProvider,
} from "react-router-dom";
```

Bước 2:

```
const router = createBrowserRouter([
  {
    path: "/",
    element: <div>Hello world!</div>,
  },
]);
```

Bước 3:

```
<RouterProvider router={router} />
```

#46. Cấu trúc dự án React (Extra)

Tài liệu:
https://legacy.reactjs.org/docs/faq-structure.html

https://dev.to/itswillt/folder-structures-in-react-projects-3dp8

React là library, vì vậy, tương tự như router, việc bạn chia cấu trúc như nào, phụ thuộc vào chính bản thân bạn.

Nếu React tích hợp sẵn việc chia cấu trúc thư mục, nó sẽ là framework (ví dụ Next.js)

1. Chia base cấu trúc dự án
Cách chia cấu trúc thư mục trong khóa học mang tính chất tương đối. Điều quan trọng là bạn có khả năng mở rộng và bảo trì dự án của bạn.

src:
- **assets**: lưu hình ảnh
- **components**: lưu các component trong ứng dụng
- **routes**
- **pages**/screens: các route/screen
- **services**: gọi backend

//todo: chia base component

#47. Tạo Header/Footer

Tài liệu:
https://www.w3schools.com/css/css_navbar_horizontal.asp
https://www.w3schools.com/howto/howto_css_fixed_footer.asp

1.Tạo Header

Test nhanh header tại đây

//Về CSS

```
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

li a:hover:not(.active) {
  background-color: #111;
}

.active {
  background-color: #04AA6D;
}
</style>
```

Về HTML:

```
<ul>
  <li><a class="active" href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
```

2. Tạo footer

Test nhanh footer tai đây

```
//CSS
.footer {
   position: fixed;
   left: 0;
   bottom: 0;
   width: 100%;
   background-color: red;
   color: white;
   text-align: center;
}

//HTML
<div class="footer">
  <p>Footer</p>
</div>
```

#48. Nested Routes với Outlet
Tài liệu:
https://reactrouter.com/en/main/start/tutorial#nested-routes

Khi sử dụng nested route, component con sẽ kế thừa lại "layout" của component cha.

Layout kế thừa:
- Header
- Footer

Sử dụng Outlet để render component con (trong layout của component cha)

#49. Client Route với Link

Tài liệu: https://reactrouter.com/en/main/start/tutorial#client-side-routing

Sử dụng Link

#50. Active Link
Tài liệu: https://reactrouter.com/en/main/start/tutorial#active-link-styling

#51. Index Route

Tài liệu:
https://reactrouter.com/en/main/start/tutorial#index-routes

#52. Xử lý NotFound

Tài liệu:
https://reactrouter.com/en/main/start/tutorial#handling-not-found-errors

```
import { useRouteError } from "react-router-dom";

export default function ErrorPage() {
  const error = useRouteError();
  console.error(error);

  return (
    <div id="error-page">
      <h1>Oops!</h1>
      <p>Sorry, an unexpected error has occurred.</p>
      <p>
        <i>{error.statusText || error.message}</i>
      </p>
    </div>
  );
}
```

Note:

Về React Router, không cần học quá sâu vào nó, vì các tính năng chuyên sâu sẽ được các framework sử dụng

Chapter 7: Setup Dự Án Backend

Setup dự án backend để có data phục vụ frontend React

#53. Giới thiệu về dự án thực hành

Mục tiêu: Thực hành sử dụng React để xây dựng website, kết hợp với backend

Công cụ: Frontend React, Backend (được cung cấp sẵn)

Tính năng:

- **CRUD Users** : thêm/sửa/xóa/hiển thị Users với dữ liệu lưu trữ tại backend
- **Đăng ký** (register), **đăng nhập** (login)

  Nếu đăng nhập thành công, trả về **access_token (mô hình stateless)**
- **CRUD Books** : cần truyền access_token để có thể sử dụng

Học xong khóa học này, học tiếp khóa thực hành React Fresher, tham khảo tại đây

#54. Backend là gì ?

1. Website thực tế gồm những gì ?

Gồm 3 thành phần chính:

- **Frontend** (ví dụ như React), chạy tại browser
- **Backend** (ví dụ java, php, nodejs…), chạy tại máy chủ server
- **Database:** nơi lưu trữ dữ liệu

Quan hệ : Frontend (FE) gọi tới => Backend (BE) query => Database

2. Tạo sao mình React không làm được website ?

Cách chúng ta code React từ trước đến nay, đã xây dựng được website, tuy nhiên tính thực tế không cao.

Lý do: data bị mất mỗi lần F5 (refresh), và chúng ta đang "hardcode" dữ liệu

Mục tiêu: xây dựng tiki, shopee, facebook… chúng ta sẽ cần nhiều hơn, và cần sử dụng thêm các công cụ khác.

Why ?

- React là frontend (HTML/CSS/JS), chạy ở phía browser, chỉ chịu trách nhiệm vẽ UI thông qua các component

- Website thực tế cần lưu trữ dữ liệu, để mỗi lần người dùng F5 (refresh), dữ liệu không bị mất như cách chúng ta đang làm

3. Backend là gì ?

Backend là cách chúng ta **quản lý dữ liệu** của website, mục đích phục vụ cho frontend.

Quản lý dữ liệu, bao gồm : truy vấn (query) data (GET) và mutate data (create/update/delete)

#55. API là gì ?

API = application programming interface

API là cầu nối giữa frontend và backend

Ví du:
https://jsonplaceholder.typicode.com/

https://jsonplaceholder.typicode.com/todos

Hiểu đơn giản nhất về API:

API là một đường link URL. Frontend sẽ gọi vào URL đấy để lấy dữ liệu (rồi hiển thị lên giao diện)

Backend chính là người tạo nên API (URL để frontend dùng)

API thường được hiển thị dưới định dạng JSON

#56. Cài đăt MongoDB Compass

MongoDB Compass không phải là database. Nó chỉ đơn thuần là một phần mềm giúp bạn thao tác với database (MongoDB)

Database của khóa học sẽ được tạo tại các video tiếp theo (lưu trữ trên cloud)

Link download: https://www.mongodb.com/try/download/compass

#57. Tạo tài khoản Mongodb Atlas
https://www.mongodb.com/cloud/atlas/register

//todo

#58. Tạo Database cho dự án

Lưu ý check allow anywhere

Lưu lại thông tin kết nối tới database

#59. Kiểm Tra Kết Nối Database
Lưu ý: sau này bị lỗi, có thể tạo lại database

Công cụ: phần mềm Mongodb Compass

Lưu lại connection

#60. Cài đặt Backend

Lưu ý: source code backend được cung cấp sẵn. Chỉ chạy và không sửa đổi

Quan tâm về backend nodejs, tham khảo lộ trình tại đây

Bước 1: download source code backend tại đây

Bước 2: update file .env (với url mongodb)
URL đã có tại video **#59**, **lưu ý thêm tên của database**

Bước 3: chạy dự án

Cài đặt thư viện cần thiết với câu lệnh: **npm i**

Chạy dự án với câu lệnh: **npm run dev**

Truy cập: http://localhost:8080/

Nếu có lỗi xảy ra, cần chú ý terminal của source code
//todo: minh họa lỗi kết nối tới database

#61. Cài đặt Postman Test API

Nguyên tắc khi sử dụng API:

nếu postman gọi được API => backend không có lỗi, lỗi nằm tại code frontend

Bước 1: Cài đặt Postman
https://www.postman.com/downloads/

Bước 2: Import collection
File collection nằm trong source code backend, đã tải tại video #60

File -> Import -> chọn file collection

Bước 3: Chạy backend

Chạy backend tại video #60, với câu lệnh: npm run dev

Bước 4: Test API

Nếu gọi API thành công, chứng tỏ mọi công cụ đã hoạt động (bao gồm backend và database)

Chapter 8: Module Users
Thực hiện CRUD Users với React và Antd

#62. Có bao nhiêu cách code CSS với React
Không có khái niệm là học React, cần phải CSS như nào, vì vốn dĩ, React nó "không quan tâm" tới việc bạn CSS ra sao. Hãy nhớ, React là cách vẽ ra UI, và CSS là công cụ của nó (được React sử dụng)

Vì vậy, trong các dự án thực tế, phụ thuộc vào từng công ty, sẽ có cách code khác nhau. Việc của bạn, là hãy trang bị những kiến thức cơ bản nhất về công cụ đấy.

Cách 1 (dễ nhất) là cách đang sử dụng trong dự án, tạo riêng lẻ các file .css và import vào component

Ưu điểm: đơn giản, tiện lợi
Nhược điểm: code dài dòng và có thể bị trùng tên class (ghi đè css của nhau)

Cách 2: kế thừa cách 1, bổ sung thêm "module" để hạn chế tối đa việc conflict (ghi đè CSS)
Tham khảo: https://create-react-app.dev/docs/adding-a-css-modules-stylesheet/

Nhược điểm: code vẫn dài dòng :v

Cách 3: sử dụng Sass (SCSS) less

Cách 4: Tailwind CSS (framework)
https://tailwindcss.com/

Ưu điểm: tất cả mọi thứ với CSS, có thể code ngắn gọn hơn thông qua các thuộc tính của Tailwind

Nhược điểm: code dài (và yêu cầu bạn cần học tailwind)

Cách 5: CSS in JS (tương tự React Native)
https://cssinjs.org/?v=v10.10.1#react-jss-example

#63. Các Thư Viện Về Component

Mục đích sử dụng thư viện, là giúp giảm thiểu thời gian code Component

Component của các thư viện, cần đảm bảo các yếu tố chính sau đây:

- Component cần phải "đẹp", dễ nhìn
- Component cần phải có sự tương tác (đã hỗ trợ các hiệu ứng CSS), ví dụ như Tooltip, Modal…
- Component cần hỗ trợ chia layout responsive
- Component cần có tài liệu rõ ràng và dễ sử dụng.

1.Các thư viện cung cấp component phổ biến

Bootstrap: nếu bạn đã dùng bootstrap với HTML, bạn cũng có thể dùng nó với React, thông qua react-bootstrap

Ưu điểm: gần gũi với tư duy Bootstrap và HTML
Nhược điểm: số lượng component hạn chế (ít component)

MUI (Material UI): cung cấp component tương tự layout của Google (Google's Material Design)
Tham khảo tại đây

Ưu điểm: bạn control 100% component sử dụng
Nhược điểm : số lượng component ít (với bản FREE)

Antd (Ant Design): cung cấp đa dạng các component, tham khảo tại đây

Ưu điểm: đa dạng component
Nhược điểm: hàng tàu khựa (Trung Quốc)

Ngoài ra còn rất nhiều thư viện UI ngoài kia như Chakra UI, PrimeReact …

#64. Cài đặt Antd

Cài đặt antd:
https://www.npmjs.com/package/antd

Cài đặt antd icons:
https://www.npmjs.com/package/@ant-design/icons

Sử dụng câu lệnh cài đặt sau:
npm i --save-exact antd@5.18.1 @ant-design/icons@5.3.7

Antd: cài đặt để sử dụng các component sẵn có của Antd

Antd-icons: sử dụng các icons của Antd

#65. Cách sử dụng Antd Component (Bonus)

Trang chủ: https://ant.design/

Lưu ý version của thư viện

Trong khóa học này, mình sử dụng version 5.x

#66. Tạo Base Giao Diện Users

//chưa làm giao diện responsive

//tạo form user phía trên table
https://ant.design/components/input

https://ant.design/components/button

//sử dụng component table
https://ant.design/components/table

#67. State Hóa Form
//lấy data của user khi submit form

#68. Sử dụng thư viện để gọi API

Có rất nhiều cách để gọi API từ frontend, có thể dùng thư viện, hoặc không (browser hỗ trợ sẵn)

Với browser, hỗ trợ sẵn **fetch** (mà không cần cài đặt gì)

Axios

Request

Cài đặt thư viện axios:
https://www.npmjs.com/package/axios

npm i --save-exact axios@1.7.2

#69. Tạo mới User

Chuẩn bị:

- Đảm bảo rằng bạn đã chạy backend
- Có thể test api = postman để biết api có hoạt động hay không

1.Giới thiệu nhanh về RESTful API

GET : lấy dữ liệu (query/fetch data)
POST: tạo mới dữ liệu
PUT: update dữ liệu
DELETE: xóa dữ liệu

2.Gọi API

Sử dụng method POST với axios:
https://axios-http.com/docs/post_example

Mục tiêu: tạo mới user thành công

#70. Config Axios Interceptor

1.Kỹ năng f12 để check api

F12 (hoặc mở Google Chrome devtool) -> chọn tab Network -> chọn API

//tách service

//lưu ý: chưa validate dữ liệu

//nếu tạo thành công, hiển thị thông báo

2.Cấu hình Interceptors

https://github.com/axios/axios?tab=readme-ov-file#custom-instance-defaults

https://axios-http.com/docs/interceptors

Interceptor tương tự middleware (người đứng giữa), giúp bạn can thiệp vào request và response của lời gọi từ frontend lên backend

Mô hình chiều đi (request):
Frontend -> gọi tới axios -> **interceptor xử lý request** (gán thêm thông tin) -> backend

Mô hình chiều về (response)
Backend -> **interceptor xử lý response** (format data) -> frontend

#71. Xử Lý Lỗi với Interceptor
Tài liệu:
https://vitejs.dev/guide/env-and-mode

https://github.com/axios/axios?tab=readme-ov-file#custom-instance-defaults

https://axios-http.com/docs/interceptors

Interceptor tương tự middleware (người đứng giữa), giúp bạn can thiệp vào request và response của lời gọi từ frontend lên backend

Mô hình chiều đi (request):
Frontend -> gọi tới axios -> **interceptor xử lý request** (gán thêm thông tin) -> backend

Mô hình chiều về (response)
Backend -> **interceptor xử lý response** (format data) -> frontend

#72. React Lifecycle
Tài liệu:
https://legacy.reactjs.org/docs/state-and-lifecycle.html
https://react.dev/learn/lifecycle-of-reactive-effects

Yêu cầu: bạn cần chạy backend để thực hiện video này

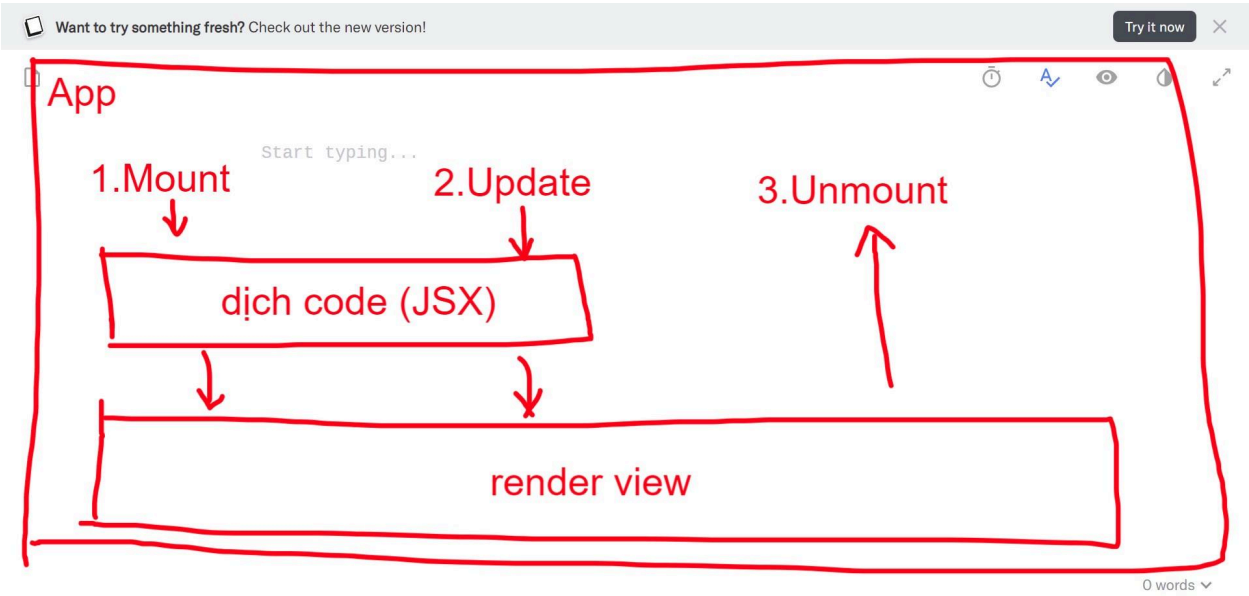
1.Bài toán đặt ra
//todo: viết api fetch user (chưa phân trang - pagination)

Cần gọi api để lấy danh sách users, render vào table

Việc gọi API là bất đồng bộ (tốn thời gian) => sử dụng async/await

Chỉ render data, khi đã có HTML

2. React Lifecycle
Tại sao lại cần useEffect (tức là ko viết await ở trên), rồi cho chạy code



#73. useEffect Hook

1.useEffect Hook

Là 1 function đặc biệt, chạy sau khi component của bạn “đã mount”

Có nghĩa là đã có HTML để cho bạn sửa đổi “giao diện”

=> dùng useEffect để can thiệp vào giai đoạn update (ép component re-render)

//fetch tất cả user

#74. Design Modal Create User
Tài liệu: https://ant.design/components/modal

```
 <Modal title="Basic Modal" open={isModalOpen} onOk={handleOk}
onCancel={handleCancel}>
      <p>Some contents...</p>
      <p>Some contents...</p>
      <p>Some contents...</p>
 </Modal>
```

#75. Lifting State Up - Hoàn Thiện Create User
Tài liệu:
https://react.dev/learn/sharing-state-between-components

Lift-up State là cách chúng ta đưa state của component con, đưa cho component cha quản lý.

Component cha sẽ truyền lại data thông qua props xuống component con.

Mục đích: chia sẻ data giữa các component “cùng cấp”.

Lift Up state là tìm cha gần nhất của các component, rồi sử dụng props để chia sẻ data

#76. Design Modal Update User

Design header với menu:
https://ant.design/components/menu

Design table với action update/delete:
https://ant.design/components/table

Design modal update:
https://ant.design/components/modal

#77. useEffect với Dependency

Sử dụng useEffect với dependency array, mỗi lần giá trị của dependency thay đổi, useEffect sẽ được chạy

#78. Hoàn Thiện Update User
//api update user

#79. Bài Tập Xem Chi Tiết User

Bước 1:
Sử dụng component:
https://ant.design/components/drawer

Bước 2: Viết logic tương tự như khi update user

#80. Bài Tập Delete User

Bước 1:
Sử dụng component:
https://ant.design/components/popconfirm

Bước 2: Viết logic xóa với API Delete

Chapter 9: Controlled Component vs Uncontrolled Component
So sánh ưu, nhược điểm của các cách code React và áp dụng uncontrolled component để nâng cao hiệu năng

#81. Setup Eslint Giúp Phát Hiện Lỗi

Mục tiêu: khi gõ code, nếu gõ sai, phát hiện lỗi và cảnh báo

Bước 1: Cài thêm extension ESLint
Restart VsCode nếu cần thiết

Bước 2: cấu hình Rule
'react/prop-types': 'off',

#82. Hiển Thị Avatar User

Hiển thị user với image (lấy từ backend)
http://localhost:8080/images/avatar/file-name

Tạo base upload button

#83. Xử Lý Sự Kiện onChange với File

CSS Image, tham khảo tại đây

1.Sự kiện onChange với File

File là loại dữ liệu đặc biệt, giúp bạn thao tác với "file upload" mà client gửi lên.

Tham khảo ví dụ về sự kiện onChange tại đây

Khi upload file, bạn có thể upload single file (1file/lần) hoặc multiple files (nhiều file/lần)

Trong khóa học này, để đơn giản nhất có thể, mình sử dụng upload single file (1 file/lần)

Quá trình upload file gồm 2 bước chính sau:

Bước 1: người dùng sử dụng browser, nhấn upload btn, rồi chọn file cần upload

Quá trình xảy ra: sau khi user chọn file upload, **file sẽ được lưu trữ trong memory của browser**
Điều này đồng nghĩa, nếu bạn không xử lý gì thêm (ví dụ lưu trữ file), chẳng có gì xảy ra (tương tự todo, refresh website là mất dữ liệu)

Khi file được lưu trữ trong memory, bạn có thể làm tính năng preview image, tức là hiển thị hình ảnh mà không cần tới backend

Bước 2: nếu muốn lưu trữ file lâu dài, client cần tới backend để lưu trữ file.

Lưu ý: sử dụng form-data thay vì json để upload file

//todo: test api upload file

2. Xây dựng preview Image với React

Mục đích: client upload file và hiển thị tại browser (chưa lưu trữ tại backend)

Tham khảo: https://stackoverflow.com/a/57781164

#84. Hoàn thiện Update Avatar

Bước 1: Gọi API upload

Nếu gọi thành công, có được tên file upload

Bước 2: Gọi API Update

Truyền thêm avatar

#85. Khái niệm Phân Trang - Pagination
Ví dụ phân trang với shopee, tiki

1.Tại sao cần phân trang
Tại 1 thời điểm, người dùng chỉ có thể xem một số lượng data nhất định (ví dụ 10 sản phẩm, 20 sản phẩm), trong khi bạn có cả trăm ngàn sản phẩm ???

=> phân trang giúp tăng trải nghiệm của người dùng (chỉ hiển thị data cần thiết), đồng thời đảm bảo tốc độ load trang nhanh nhất có thể (ít data load sẽ nhanh)

2.Kiến thức sử dụng khi phân trang
Bản chất của việc phân trang, là sử dụng OFFSET và LIMIT (với sql)
Tham khảo: https://www.w3schools.com/php/php_mysql_select_limit.asp
https://www.sqltutorial.org/sql-limit/

Backend sẽ dựa vào OFFSET và LIMIT để lấy lên dữ liệu tương ứng.

Tuy nhiên, **ở Frontend chỉ cần truyền số lượng phần tử lấy tối đa (LIMIT) và trang muốn lấy (PAGE), backend sẽ cần tự tính toán phần OFFSET**

Ví dụ: Frontend muốn lấy data tại page = 1, limit = 10
Page =2, limit = 10

3. Test API Phân trang
//todo
//viết base api phân trang
//tạo fake data
Truyền thêm 2 tham số là current và pageSize

#86. Sử Dụng Phân Trang với Antd

Tài liệu:

https://ant.design/components/result

https://ant.design/components/pagination

https://ant.design/components/table#pagination

```
pagination={
{
current: current,
pageSize: pageSize,
showSizeChanger: true,
total: total,
showTotal: (total, range) => { return (<div> {range[0]}-{range[1]} trên {total} rows</div>) }
} }
```

//todo: update khi vào màn hình, cần gọi api phân trang

Xử lý sự kiện onChange:

```
const onChange = (pagination, filters, sorter, extra) => {  };
```

#87. Hoàn thiện Phân Trang User

//todo

#88. Khái niệm Re-render

1.Nguyên tắc của Re-render

State/Props thay đổi => component re-render

Nếu component cha render -> tất cả component component con re-render

Cần chú ý tới form, vì có sự kiện onChange (mỗi lần typing là re-render)

Lưu ý: React có hiệu năng rất tốt. Bạn re-render hàng chục/tới trăm lần trên giây, nó vẫn chưa giật/lag

2.Controlled Component vs Uncontrolled Component

Control: kiểm soát

Cách chúng ta đã làm, sử dụng state/props với form, gọi là controlled component

Ưu điểm: bạn "Kiểm soát" từ a tới z . **Muốn gì được nấy, đổi lại, bạn cần "tự code"**

Nhược điểm: re-render "quá nhiều lần" sẽ dẫn tới giật/lag giao diện

3.Các thư viện hay dùng

- render ít
- hỗ trợ validate dữ liệu
- hỗ trợ format dữ liệu...

react hook form

https://react-hook-form.com/

https://www.npmjs.com/package/react-hook-form

formik

https://formik.org/

https://www.npmjs.com/package/formik

#89. Sử Dụng Uncontrolled Component Cho Register

Tài liệu:
https://ant.design/components/form

//todo: design base form Register (fullName, email, password, phone)

Mục tiêu :

- lấy được data khi submit
- Validate dữ liệu
- Hạn chế render (không sử dụng state)

#90. Hoàn thiện tính năng Register

- Validate dữ liệu

```
{
    required: true,
    pattern: new RegExp(/\d+/g),
    message: "Wrong format!"
}
```

- Gọi API backend

- Giới thiệu API getFormValues/setFormValue

#91. Chia Layout Responsive (Extra)

Tài liệu:
https://ant.design/components/grid

Khác với Bootstrap (12 columns), Antd dùng 24 columns.
(Bạn tự trang bị kiến thức về Bootstrap, như vậy sẽ thấy antd "giống hệt" nó)
https://getbootstrap.com/docs/5.0/layout/breakpoints/#available-breakpoints

Breakpoint	Class infix	Dimensions
X-Small	*None*	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

```
$grid-breakpoints: (
  xs: 0,
  sm: 576px,
  md: 768px,
  lg: 992px,
  xl: 1200px,
  xxl: 1400px
);
```

Chapter 10: Module Auth

Chức năng đăng ký/đăng nhập sử dụng access token

#92. Bài Tập Design Login

- Design giao diện
- Lấy được data khi submit form

#93. Hoàn Thiện Tính Năng Login

Thêm delay/thêm loading

#94. Cơ chế Stateless sử dụng Token

Có 2 cơ chế chính để xây dựng website, là **stateful** và **stateless**
Stateful: chứa full trạng thái (state), sử dụng session
Stateless: không chứ trạng thái (state), sử dụng token

1.Mô hình truy cập

request từ client -> gửi lên server. Server gửi phản hồi (response) cho client

Nếu như client (giao diện HTML) và server (logic xử lý) để trong cùng 1 source code
=> đây là cơ chế stateful, vì server kiểm soát từ A tới Z

Ngược lại, nếu bạn tách riêng code frontend và code backend, đây là mô hình stateless
Ví dụ: bạn code frontend = react và backend = java (2 source code khác nhau)

2.Làm sao để xác thực người dùng đã đăng nhập

Với stateful, bạn sử dụng session (nằm ngoại phạm vi của khóa học)
Với stateless (tách riêng frontend và backend):

Hãy tưởng tượng, bạn đi du lịch tại nước ngoài (ví dụ từ Việt Nam sang Lào)

Việt Nam (frontend), Lào (backend), và bạn muốn mua gỗ của Lào (API backend)

Bước 1: bạn cần xin visa, vì visa sẽ định danh bạn là ai (passport). Visa này cần do Lào cấp (backend)
=> đây là quá trình login, bạn login thành công, backend trả về access_token (giúp định danh bạn là ai)

Bước 2: Bạn muốn buôn gỗ từ Lào về Việt Nam (frontend muốn gọi API của backend)

Bạn sẽ cần chứng minh bạn "có quyền hợp pháp" (show passport ra)

Như vậy, tại mỗi lời gọi API, bạn cần kèm theo access_token. Backend sẽ check cái token này, nếu hợp lệ, cho bạn truy cập API, ngược lại, từ chối truy cập

#95. Access Token sử dụng với Stateless

Stateless, hiểu đơn giản là bạn tách riêng frontend và backend

Quy trình:

- Frontend cần login để lấy token (access_token)

- Frontend muốn truy cập endpoint (api của backend), sẽ cần cung cấp token (access_token). Nếu token hợp lệ, backend cho phép truy cập nguồn tài nguyên, ngược lại thì không.

Tại sao cần token ?
Nếu không có cơ chế "xác thực/định danh ai là người đang đăng nhập", thì người dùng - ai ai cũng có "vai trò giống nhau" (quyền hạn giống hệt nhau)

Sau này bạn muốn: admin có quyền CRUD products, còn người dùng thông thường chỉ được phép xem data chẳng hạn.

1.JWT - JSON Web Token
https://jwt.io/

Là một dạng "mã hóa" dữ liệu, giúp bạn lưu trữ thông tin người dùng đăng nhập.

Về cơ chế tạo ra token cho frontend sử dụng, được học tại các khóa backend, tham khảo tại đây

2. Các lỗi thường gặp

Để cho tiện lợi, chúng ta thường gán "token" vào header của request (như vậy sẽ không cần sử url/body của api)

Lỗi 1: không truyền Access Token (đối với các API check quyền hạn/token) ở header

Lỗi 2: xem lại lỗi 1 :v

#96. Nơi nào dùng để lưu trữ Token tại Frontend (Extra)

Tài liệu: https://datatracker.ietf.org/doc/html/rfc6749

Thực tế, cách chúng ta đang làm, là tuân theo chuẩn OAuth2.

Với Access Token, được lưu trữ tại LocalStorage

Bonus: Phân biệt LocalStorage, SessionStorage, Cookies

Local Storage:

- Data lưu mãi mãi (chỉ mất nếu bạn xóa nó đi)
- Code javascript có thể truy cập được

Session Storage:

- Data sẽ bị mất (nếu bạn đóng tab/browser)
- Code javascript có thể truy cập được

Cookie:

- Data sẽ “tự động mất” khi hết hạn
- Bạn có thể chặn/cho phép code javascript truy cập giá trị

#97. Logic Xử Lý Sau Khi Login

Bạn đã đăng nhập thành công, **backend trả ra access_token và thông tin user đăng nhập.**

Với thông tin user đăng nhập:

- Bạn cần lưu thông tin lại, để sử dụng ở các component khác nhau, ví dụ Header cần hiển thị thông tin user đăng nhập

- **Mỗi lần F5 (refresh website), thông tin này sẽ bị mất.** Cần có cơ chế để lấy lại thông tin người dùng.

  Giải pháp: gọi API backend (truyền access token)

Với access_token:

- Bạn lưu vào local storage
- **Mỗi lần f5 (refresh website), bạn không bị mất thông tin token này**
- Cấu hình để mỗi lần gọi API, sẽ tự động truyền thêm token ở header

1.Giới thiệu về React Context
https://react.dev/reference/react/createContext

Bài toán: sharing data giữa các component ?

#98. Sử Dụng React Context API

Tài liệu: https://react.dev/reference/react/createContext

Mục tiêu:

- Lưu thông tin user vào React Context
- Hiển thị thông tin user đăng nhập lên header

Bước 1: tạo context

Bước 2: wrap component

Bước 3: sử dụng context

#99. React props.Children

Tài liệu:
https://legacy.reactjs.org/docs/jsx-in-depth.html#children-in-jsx

https://react.dev/learn/passing-props-to-a-component#passing-jsx-as-children

https://stackoverflow.com/questions/49706823/what-is-this-props-children-and-when-you-should-use-it

https://stackoverflow.com/questions/66892066/conditionally-add-object-to-an-array-while-being-declared

```
 ...(!user.id ? [{
        label: <Link to={"/login"}>Đăng nhập</Link>,
        key: 'login',
        icon: <LoginOutlined />,
    }] : []),

    ...(user.id ? [{
        label: `Welcome ${user.fullName}`,
        key: 'setting',
        icon: <AliwangwangOutlined />,
        children: [
            {
                label: 'Đăng xuất',
                key: 'logout',
            },
        ],
    }] : []),
```

#100. Xử Lý F5 (Refresh Page)

//nhấn enter submit form login
https://stackoverflow.com/a/59148029

//gán token vào header với interceptor

```
// Add a request interceptor
instance.interceptors.request.use(function (config) {
   if (typeof window !== "undefined" && window && window.localStorage &&
window.localStorage.getItem('access_token')) {
       config.headers.Authorization = 'Bearer ' + window.localStorage.getItem('access_token');
   }
   // Do something before request is sent
   return config;
}, function (error) {
   // Do something with request error
   return Promise.reject(error);
});
```

//gọi API fetch Account
//todo

#101. Private Route với React

Tài liệu: https://www.robinwieruch.de/react-router-private-routes/

https://stackoverflow.com/a/66289280

#102. Chức năng Logout

Mapped {/api/v1/auth/logout, POST}
truyền bearer token ở header
//todo

#103. Tổng Kết về mô hình Stateless với Access Token (JWT)

1.Luồng login và sử dụng Token

Cơ chế xác thực người dùng được làm tại backend. Frontend sau khi login thành công, sẽ được cấp token để truy cập API (access_token)

Mỗi lần truy cập API, frontend cần truyền access_token (vào header request) để xác thực. Token thường được viết dưới dạng JWT (json web token)

Mỗi lần f5, thông tin lưu trữ tại frontend sẽ bị mất => cần gọi API để lấy thông tin của user đăng nhập

2.Chia sẻ data giữa các component

Sử dụng React Context API để chia sẻ data giữa các component (tránh tình trạng cần truyền props từ cha sang con/hoặc giữa các component không có mối quan hệ với nhau)

Có thể sử dụng các thư viện để quản lý state hiệu quả hơn.

Tham khảo về Redux Javascript (miễn phí) tại đây

Redux (typescript) tại đây

Chapter 11: Module Book (Luyện Tập)
Luyện tập CRUD và upload file với model Book

#104. Nguyên Tắc Thực Hành

Đây là chương học thực hành, bạn sẽ cần "tự code". Vì không có thực hành, kiến thức không là của bạn

Tuy nhiên, các video vẫn cung cấp đầy đủ source code (trong trường hợp bạn cần tham khảo). Video chỉ hướng dẫn, định hướng cách làm, không code từ a tới z.

Nguyên tắc cần tuân theo:

1.Bạn không cần "nhớ code" từ a tới z. Điều quan trọng nhất, là bạn cần hiểu bạn đang làm gì. Tránh tình trạng code như 1 cái máy (không biết bản thân làm gì)

Bạn có thể tham khảo lại các chương học trước (tham khảo code để hiểu tính năng), có thể copy/paste code thoải mái. Tuy nhiên, cần hiểu bạn copy/paste cái gì ? Và việc copy/paste đấy có tác dụng gì ?

2. Quá trình thực hành sẽ bao gồm:

- Bạn xem video để nắm được yêu cầu cần làm, cũng như sản phẩm cần đạt được
- Với API backend, bạn có thể dùng Postman để test trước khi code frontend React
- Bạn code theo cách bạn hiểu kiến thức. Bạn được tùy ý sử dụng code tại các chương học trước (phần nào quên có thể xem lại, copy/paste)
- Bạn cần tự thực hành, trước khi xem đáp án/source code cung cấp
- Chỗ nào bạn quên, bạn có thể đọc Google và đọc tài liệu (react, youtube, stackoverflow…)

#105. Bài Tập Hiển Thị Book

Yêu cầu: Hiển thị data Book với antd Table

//giải thích các field & ý nghĩa của Book
"_id": id của book
"thumbnail": ảnh thumbnail
"slider": [ ] : ảnh slider (bỏ qua)
"mainText": tiêu đề
"author": tác giả
"price": giá tiền
"sold": số lượng đã bán
"quantity": số lượng
"category": thể loại

//Lưu ý: fetch data tại table
//format giá tiền

Bước 1: Tạo Fake data (hardcode)
https://ant.design/components/table

Với table của Antd, gồm 2 thành phần: dataSource và columns
columns là nơi khai báo các cột của table, và cách mapping data
dataSource là data của table (array chứa object)

Bước 2: Gọi API để lấy data
//bạn có thể hardcode url api để có dữ liệu

Sử dụng useState để lưu trữ data của table
Sử dụng useEffect để gọi API

Bước 3: Xử lý onChange table (pagination)
Sử dụng useState để lưu trữ current/pageSize/total

Sử dụng useEffect với dependency [ ]

#106. Bài Tập Xem Chi Tiết Book

Mục tiêu: hiển thị chi tiết book (có hình ảnh) khi xem chi tiết

Bước 1: Tạo Drawer
https://ant.design/components/drawer

Bước 2: Quản lý state

State open/close được khai báo tại Table

State data detail được khai báo tại Table

Bước 3: Hiển thị data
Nhận props từ component cha và hiển thị data

Hiển thị book với image (lấy từ backend)
http://localhost:8080/images/book/file-name

#107. Bài Tập Thêm Mới Book (Controlled Component)

Sử dụng controlled component với state của React

Lưu ý: chưa check validate tại frontend => khi test cần điền đầy đủ thông tin

Bước 1: Tạo form
https://ant.design/components/modal

https://ant.design/components/input-number
//lưu ý về sự kiện onChange

https://ant.design/components/select
//lưu ý về sự kiện onChange

options={[
{ value: 'Arts', label: 'Arts' },
{ value: 'Business', label: 'Business' },
{ value: 'Comics', label: 'Comics' },

{ value: 'Cooking', label: 'Cooking' },
{ value: 'Entertainment', label: 'Entertainment' },
{ value: 'History', label: 'History' },

{ value: 'Music', label: 'Music' },
{ value: 'Sports', label: 'Sports' },
{ value: 'Teen', label: 'Teen' },
{ value: 'Travel', label: 'Travel' },

]}

Bước 2: Truyền thông tin vào API

Bắt buộc cần upload hình ảnh thumbnail trước khi gọi api

Upload hình ảnh, sau đấy gọi api tạo mới sách
const resUpload = await handleUploadFile(selectedFile, "**book**");

POST **http://localhost:8080/api/v1/book**

```
{
   "thumbnail": "abc.png",
   "mainText": "bla bla",
   "author": "hoidanit",
   "price": 150000,
   "quantity": 100,
   "category": "Arts"
}
```

Fix bug click vào upload image => đóng modal => click lại chính xác image đấy
https://stackoverflow.com/a/40429197

Bước 3: Reset data
Sau khi gọi API thành công, cần fetch lại data của table, đồng thời reset state cũng như đóng modal

#108. Bài Tập Thêm Mới Book (Uncontrolled Component)
Sử dụng uncontrolled component với antd
//Bạn có thể off component tại video trước và thay bằng component mới

Yêu cầu: sử dụng form của antd để làm uncontrolled component

Các input sau sẽ dùng form với antd

```
const [mainText, setMainText] = useState("");
const [author, setAuthor] = useState("");
const [price, setPrice] = useState("");
const [quantity, setQuantity] = useState("");
const [category, setCategory] = useState("");
```

Bước 1: khai báo form
https://ant.design/components/form
const [form] = Form.useForm();

Và, gán vào Component:
<Form
form={form}
onFinish={handleSubmitBtn}
>

Bước 2: gán name cho input (tham khảo phần **login**) (validate nếu muốn)
Cần bọc input vào form item

Bạn xóa hết state của React gán cho input, thay thế bằng **name**

```
<Form.Item
 label="Email"
 name="email"
>
  <Input />
</Form.Item>
```

Riêng phần upload hình ảnh vẫn dùng state
//input file
style={{ display: "none" }}

Bước 3: submit form
onOk={() => form.submit()}

const handleSubmitBtn = async (values) => { }
Và được lấy ra dựa vào form của antd, ví dụ:
const { mainText, author, price, quantity, category} = values;

Upload hình ảnh, sau đấy gọi api tạo mới sách

Bước 4: reset data trên form
form.resetFields();

#109. Bài Tập Cập Nhật Book (Controlled Component)

Sử dụng Controlled component với state của React

Bạn tạo mới data để test (không nên sử dụng data được tạo sẵn)

1.Sự khác biệt giữa Update và Create

Khi update, bạn cần **truyền thêm _id**

Hàm useEffect sẽ được dùng để lắng nghe sự thay đổi và gán ra trị đầu vào cho modal

2. Các bước thực hiện

Bước 1: khai báo state giống như khi tạo mới
Khai báo thêm const [id, setId] = useState("");

Bước 2: gán giá trị cho modal

Sử dụng useEffect

```
 useEffect(() => {
      if (dataUpdate && dataUpdate._id) {
       //your code
       }
   }, [dataUpdate])

//hiển thị hình ảnh preview lấy trực tiếp từ backend
setPreview(`${import.meta.env.VITE_BACKEND_URL}/images/book/${dataUpdate.thumbnail}`)
```

Bước 3: logic khi update
//không có ảnh preview + không có file => return

//có ảnh preview và không có file => không upload file
=> giá trị của thumbnail lấy từ state của dataUpdate

//có ảnh preview và có file => upload file
=> giá trị của thumbnail lấy từ kết quả của upload file

API update:
PUT **http://localhost:8080/api/v1/book**

```
{
    "_id": "66693c9066d5d0fb5fca16c0",
    "thumbnail": "abc.png",
    "mainText": "bla bla",
    "author": "hoidanit",
    "price": 150000,
    "quantity": 100,
    "category": "Arts"
}
```

#110. Bài Tập Cập Nhật Book (Uncontrolled Component)

Sử dụng uncontrolled component với form của Antd

Làm tương tự như việc bạn sử dụng Create với uncontrolled component

Bước 1: khai báo form
https://ant.design/components/form
const [form] = Form.useForm();

Và, gán vào Component:
<Form
form={form}
onFinish={handleSubmitBtn}
>

Bước 2: gán giá trị cho modal tại useEffect
https://ant.design/components/form#components-form-demo-control-hooks

form.setFieldsValue({
//your code
})

Bước 3: **update form**
Cần bọc input vào form item

Bạn xóa hết state của React gán cho input, thay thế bằng **name**
<**Form.Item**
label="Email"
name="email"
>
<Input />
</Form.Item>

Riêng phần upload hình ảnh vẫn dùng state
//input file
style={{ display: "none" }}

Bước 4: submit form
onOk={() => form.submit()}

const handleSubmitBtn = async (values) => { }
Và được lấy ra dựa vào form của antd, ví dụ:
```
 const {id,  mainText, author, price, quantity, category} = values;

//todo
```

Bước 5: reset data trên form
form.resetFields();

#111. Bài Tập Xóa Book

//nên tạo mới data, rồi xóa

//nếu muốn có lại data fake, xóa hết data, chạy lại backend, backend sẽ tự động tạo data fake (khi count = 0)

//Sử dụng component Popconfirm:
https://ant.design/components/popconfirm

DELETE http://localhost:8080/api/v1/book/ID-BOOK

Chapter 12: Tổng kết
Tổng kết các kiến thức đã học

#112. Thêm Loading Bar (Extra)
Tài liệu:
https://www.npmjs.com/package/nprogress
https://github.com/rstacruz/nprogress

Bước 1: cài đặt
npm install --save-exact nprogress@0.2.0

Bước 2: cấu hình axios

```
import NProgress from 'nprogress';

NProgress.configure({
    showSpinner: false,
    trickleSpeed: 100,
});

// Add a request interceptor
instance.interceptors.request.use(function (config) {
    NProgress.start();
}, function (error) {
    NProgress.done();
});

// Add a response interceptor
instance.interceptors.response.use(function (response) {
    NProgress.done();
}, function (error) {
    NProgress.done();
 });
```

#113. Fix Các Bug Còn Tồn Đọng

//bug 1:
F5 lại trang, fix active menu header
https://reactrouter.com/en/main/hooks/use-location

//bug 2:
Thêm loading cho table hoặc button (sử dụng phần delay api)

//bug 3:
Lưu ý về mỗi lần fetch api (khi nhấn nút button - nếu ngon, về mặt UX - cần thêm button loading)

//bug 4:
Fix lỗi eslint
https://github.com/facebook/react/issues/14920
https://stackoverflow.com/a/77324978

#114. Hook Là Gì ?

React có bao nhiêu hook ?

- Bắt đầu bằng keyword **use**, ví dụ useState, useEffect

Rule khi sử dụng hook :
https://react.dev/warnings/invalid-hook-call-warning
Các lưu ý khi sử dụng hook (như khai báo ở đầu function)

Không khai báo hook trong function thông thường

Chỉ cần **useState** và **useEffect** là đủ (thiếu gì, google cái đó)

#115. Phân Tích Câu Chuyện Deploy ?

Quy trình phát triển một sản phẩm phần mềm (website), bao gồm 3 bước chính:
Bước 1: development (phát triển), hay còn hiểu là bước coding
Bạn chạy tất cả tại máy tính cá nhân bạn (localhost)

=> đây còn gọi là môi trường dev (development), giải thích cho lý do tại sao bạn hay gõ là : **npm run dev**

Bước 2: testing (kiểm thử). Tester sẽ tiến hành kiểm thử phần mềm của bạn (nhằm phát hiện bugs và đảo bảo chất lượng của phần mềm)

=> chúng ta không làm cái này nên tạm thời bỏ qua :v

Bước 3: production (sản phẩm thực tế) sẽ được chạy tại máy chủ + với tên miền. Quá trình để có được sản phẩm chạy thực tế, gọi là deployment (triển khai)

Ví dụ: sau khi code được website (bước 1), mình đã deploy lên server tại hoidanit.vn

Để chạy production (prod), chúng ta cần build ứng dụng, rồi chạy (như vậy nó sẽ tối ưu hóa hiệu năng), giải thích cho lý do tại sao chúng ta dùng:
npm run build
npm start

1. Ghi CV có cần sản phẩm deploy ?
Không bắt buộc bạn ghi CV là cần phải có link sản phẩm deploy. Nếu có thì càng tốt, còn không có nó cũng chẳng sao, ở đây là 50/50.

Câu chuyện deploy nó chỉ xảy ra với beginner (xin thực tập/fresher), vì khi đã đi làm, chắc chắn 100% sẽ không ghi link sản phẩm (vì đây là sản phẩm của công ty)

Trường hợp bạn không có link sản phẩm demo, bạn cần có link github sản phẩm của bạn (hoặc bạn có thể quay video demo nếu muốn)

2. Quá trình Deploy với 1 Website

Gồm 3 bước chính (là 3 thành phần của website)

- Deploy frontend
- Deploy backend
- Deploy database

3. Chuyện Dùng Miễn Phí và Trả phí

Miễn phí thì không có chuyện "ngon, bổ, rẻ", vì deploy, chính là việc bạn "chạy thực tế".

Ngày nay, ngày càng ít các nơi "hosting" (nơi chứa mã nguồn website) miễn phí.

Nếu có FREE, sẽ bị các nhược điểm sau:

- Giới hạn nguồn tài nguyên (ví dụ RAM 512 MB, lưu trữ 1GB, không lưu trữ file...)
- Data (giữ liệu) sẽ bị xóa sau 1 khoảng thời gian nhất định (3 tháng, 6 tháng...)
- Bị ngủ đông (hibernate) nếu như không có người dùng truy cập (tắt đi cho đỡ tốn điện)

  Khi cần truy cập, sẽ cần cho nó wake-up (chờ từ 30s tới 1 phút)

Dùng "Trả phí" có các lợi thế sau:

- Không bị các nhược điểm của cách làm FREE

Nhược điểm của cách làm trả phí:

- **Bạn cần trả phí nhiều** (nếu muốn không làm gì). Pay as you go
  Chi phí mua tên miền, mua vps, mua database, backup dữ liệu...
- **Bạn trả phí ít**, bắt buộc bạn cần có kiến thức về deploy (ví dụ docker, mua tên miền, cấu hình vps...

#116. Deploy Backend Với Render

Lưu ý: Copy mã nguồn sang 1 folder khác, tránh tình trạng ảnh hưởng tới dự án đang chạy tại máy tính của bạn

Cần chuẩn bị:

- Tài khoản Github
- Đẩy mã nguồn lên Github

Bước 1: đăng nhập vào render sử dụng Github/Gitlab/Google

Trang chủ: https://render.com/

Trang để đăng nhập: https://dashboard.render.com/

Bước 2: Triển khai với Render

#117. Deploy Frontend Với Vercel

Lưu ý: copy mã nguồn sang 1 folder khác, tránh tình trạng ảnh hưởng tới dự án đang chạy tại máy tính của bạn

//update source code frontend
https://vercel.com/login

//vercel.json
```
{
  "rewrites": [
    { "source": "/(.*)", "destination": "/" }
  ]
}
```

Cần chuẩn bị:

- Tài khoản github
- Đẩy mã nguồn lên github

Bước 1: đăng nhập vào Vercel
Trang chủ: https://vercel.com/

Trang login: https://vercel.com/login

Bước 2: Triển khai với Vercel

#118. Nhận xét về dự án thực hành

Ưu điểm:

- Nắm vững các kiến thức cốt lõi của React : state & props
- Tối ưu hóa Render với uncontrolled component
- Chia sẻ data giữa các component với React Context API
- Tăng tính trải nghiệm UI/UX với thư viện antd
- Thực hành dự án React với API của backend

Nhược điểm:

- Axios retry : api failed thì cần có cơ chế tự động gọi lại
- Refresh token
- Table với filter, sort
- Thực hiện nhiều CRUD hơn

#119. What's next

1.Về các kiến thức của React

Sử dụng các hook của React : useRef, useCallback, useMemo..

Nên đọc qua tài liệu của React để có góc nhìn tổng quan:
https://react.dev/learn/escape-hatches

2. Về lộ trình React của Hỏi Dân IT
Nếu bạn muốn chinh phục React từ A tới Z, chi tiết về lộ trình, tham khảo tại đây

Các khóa học tiếp theo của hoidanit, học theo thứ tự sau:

Khóa 1: React Typescript dự án Portfolio (link tham khảo tại đây)
Khóa học này sẽ giúp bạn chuyển đổi từ code Javascript sang code React với Typescript

Khóa 2: React Test Fresher (link tham khảo tại đây)
Mục đích của khóa học này, là cung cấp backend để cho bạn "**tự thực hành code React**" theo gợi ý của mình (tương tự như chapter 11 - module book)

Khóa 3: React ProMax với Framework Nextjs (link tham khảo tại đây)
Khóa học này sử dụng React với Nextjs (typescript), công nghệ đang là trending của React

3.Về phát triển một sản phẩm website hoàn chỉnh

React là frontend, nếu bạn muốn có 1 website hoàn chỉnh, cần học backend

Lộ trình backend Node.js , tham khảo tại đây

Lộ trình backend Java Spring, tham khảo tại đây

#120. Suy nghĩ về level Intern/Fresher

Q: Học xong khóa học, có thể đi thực tập/đi làm được không

Trả lời chi tiết, tham khảo tại đây

Chapter 13: React 19 (Bonus)
Tìm hiểu tổng quan về React version 19

#121. React 19 ra đời khi nào ?

1.Lịch sử ra đời

React 19 RC (release candidate) ra đời vào 25/04/2024,

Đây là version beta, dùng để test và lắng nghe feedback của cộng đồng (community)

https://github.com/facebook/react/blob/main/CHANGELOG.md

v16.8.0 - 06/02/2019 (React sử dụng Hook - cách code của khóa học này)

v18.0.0 - 29/03/2022

v18.3.1 - 26/04/2024 (chuẩn bị cho ra đời chính thức của React 19)

2. Các điểm mới của React 19
https://react.dev/blog/2024/04/25/react-19

Các điểm chính:

Điểm 1: cải thiện form (client). Hãy tưởng tượng, bạn code php :v

Điểm 2: server component (đã bao gồm form tại server)

Điểm 3: cải thiện React : compiler, metadata, ref…

https://react.dev/blog/2024/02/15/react-labs-what-we-have-been-working-on-february-2024

https://www.developerway.com/posts/react-compiler-soon

#122. Upgrade Project to React 19 (RC)

Chỉ thực hiện cách làm này (nếu bạn sử dụng version cũ hơn và React chưa ra version 19 chính thức)

Đi làm thực tế, ít khi bạn upgrade (khi dự án đã golive)

//checkout sang nhánh code khác để test (hoặc sử dụng git để rollback code)

Tham khảo cách làm tại đây:
https://react.dev/blog/2024/04/25/react-19-upgrade-guide

Lịch sử public version của React:
https://www.npmjs.com/package/react?activeTab=versions

Bước 1: cài đặt
npm install --save-exact react@19.0.0-rc-3563387fe3-20240621 react-dom@19.0.0-rc-3563387fe3-20240621

Bước 2: Test dự án

#123. Câu Chuyện Về Next.JS

Tương lai của React là Next.js

Trang chủ của React, gợi ý trực tiếp sử dụng Next.js (điều này chỉ xảy ra từ tháng 4/2023)
https://react.dev/learn/start-a-new-react-project

Tuy nhiên, với beginner, mình không dùng trực tiếp Next.js (vì độ khó nó cao), tương tự như cách bạn học Angular, bạn cần:

+ Bắt buộc biết Typescript
+ Bắt buộc có tư duy về DI (dependency injection và IoC - inversion of control)

React khi dùng với Next.js, nó chính thức là framework (tương tự Angular và Vue)

Lời Kết

Như vậy là chúng ta đã cùng nhau trải qua hơn 125+ video về sử dụng React dành cho frontend developer.

Tất cả các kiến thức mình chia sẻ, đều được lấy từ kinh nghiệm đi làm của mình và… các trang tài liệu về React.

Dĩ nhiên rằng, trong quá trình quá trình thực hiện khóa học này, mình sẽ không thể tránh khỏi những sai sót (vì nếu không có sai số thì mình làm member của team React rồi :v).

Vì vậy, nếu thấy sai sót, các bạn cứ thoải mái đóng góp qua Fanpage Hỏi Dân IT nhé.
https://www.facebook.com/askITwithERIC

Nếu bạn thấy khóa học này hữu ích, đừng quên Review đánh giá trên Udemy nhé ^^

Hẹn gặp lại các bạn ở các khóa học tiếp theo ….
Hỏi Dân IT (Eric)