

TÀI LIỆU KHÓA HỌC “React Typescript - Portfolio”

Tác giả: Hởi Dân IT & Eric

Version: 1.0

Note cập nhật:

Update tài liệu

Chapter 1: Bắt buộc xem - Không bỏ qua chương học này	3
#1. Hướng dẫn sử dụng khóa học này hiệu quả	3
#2. Tài liệu của khóa học	5
#3. Tại sao khóa học này ra đời ?	6
#4. Demo kết quả đạt được	8
#5. Sử Dụng Source Code của Khóa học	9
#6.1 Cách Dùng Udemy - Hỗ Trợ Hỏi Đáp Q&A	10
#6.2 Về Tác giả	11
Chapter 2: Setup Environment	13
#7. Chuyện Cài Đặt Công Cụ (Bắt Buộc Xem)	13
#8. Cài đặt Node.js	15
#9. Cài đặt Visual Studio Code	17
#10. Cấu hình Visual Studio Code	17
#11. Tại sao mình dùng VScode ?	18
#12. Cài đặt và sử dụng Git	19
#13. Cài đặt Google Chrome	20
Chapter 3: React với TypeScript	21
#14. Tổng quan về chapter	21
#15. Why Coding (Frontend) with Typescript	21
#16.1 Ôn tập kiến thức React với Javascript	23
#16.2 Ôn tập Kiến Thức TypeScript	23
#17. Tài liệu sử dụng React với Typescript	24
#18. Cài Đặt Dự Án Thực Hành	25
#19. Cách mình setup dự án (Extra)	26
#20. Hello World với React TypeScript	27
#21. Component	28
#22. React Function Component/React.FC (Extra)	30
#23. Props	31
#24. Type vs Interface (Extra)	32
#25. Default/Optional Props	32
#26. TypeScript Generics (Extra)	33
#27. Sử dụng State	34
#28. Truyền function từ cha sang con	34
#29. Bài tập Delete Todo	35
#30. Tổng kết về chapter	35
Chapter 4: Thực Hành Dự Án Portfolio	36
#31. Tổng quan về chapter	36
#32. Dự án thực hành (JavaScript)	36
#33. Tích hợp TypeScript vào dự án React JavaScript	37
#34. Hero Section (TypeScript)	38
#35. TypeScript Absolute Import	38
#36. Hoàn thiện HomePage Screen	39

#37. React Context với TypeScript	39
#38. useRef với TypeScript	40
#39. Hoàn thiện Project Screen	40
#40. Hoàn thiện About Screen	41
#41. Tổng kết về chapter	41
Chapter 5: Deploy Production - Triển Khai Dự Án Thực Tế	42
#42. Tổng quan về chapter	42
#43. Create Vercel Account	42
#44. Link Vercel với Github	42
#45. Triển khai dự án React Vite với Vercel	42
#46. Mua Tên Miền (Mắt Bão)	43
#47. Mapping tên miền tới Vercel	44
#48. Cách tự code dự án React với TypeScript của bạn	45
#49. What's next ?	46
Đánh Giá (Review/Rating) Khóa Học	47

Chapter 1: Bắt buộc xem - Không bỏ qua chương học này

Hướng dẫn sử dụng khóa học hiệu quả

#1. Hướng dẫn sử dụng khóa học này 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 nhẩm => đưa luôn máy tính cho nó. Rồi máy tính ra, là không biết làm phép tính

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

5. Về source của cả khóa học

Source code của cả khóa học ĐƯỢC CUNG CẤP (full cả khóa và theo từng video)

#2. Tài liệu của khóa học

//Link download tài liệu khóa học

Bookmark và download (bản backup) theo [link này](#).

Trong quá trình sử dụng tài liệu, nếu bạn thấy chỗ nào chưa đúng, hoặc link tham khảo trong tài liệu không sử dụng được (dead), các bạn chủ động inbox qua Fanpage Hỏi Dân IT (hoặc gửi email tới địa chỉ ads.hoidanit@gmail.com) để mình update tài liệu các bạn nhé

Link fanpage Hỏi Dân IT: <https://www.facebook.com/askITwithERIC/>

Note: Các tiêu đề bài học có chữ “Extra”, có nghĩa là chủ đề mở rộng (thông thường sẽ là các topic nâng cao mình làm thêm để đáp ứng nhu cầu của học viên)

#3. Tại sao khóa học này ra đời ?

1. Các vấn đề được giải quyết trong khóa học

- Cách code ứng dụng React sử dụng với Typescript (hướng dẫn từ cách đọc tài liệu, search google cho tới cách thực hành)
- Hướng dẫn cách “migrate” (chuyển đổi) từ code React với JavaScript sang code React với TypeScript
- Là nền tảng để sử dụng các framework chuyên nghiệp của React, ví dụ Next.js (không học code Next.js trong khóa học này)
- Giúp bạn có nhanh 1 CV (portfolio) xịn sò cho cá nhân bạn (hosting online)
Link demo sản phẩm đạt được, tham khảo tại đây: <https://cv.hoidanit.vn/>

Khóa học này không phải (không giải quyết) các vấn đề sau:

Vấn đề 1: bạn bắt đầu từ số 0, muốn học code React với TypeScript

Để học được React, bạn cần biết tới HTML, CSS và JavaScript trước tiên.
Chưa học HTML, CSS và Javascript, không thể học React

Vấn đề 2: bạn biết duy nhất HTML, CSS, JS, muốn học code React với TypeScript

React (nguyên thủy) là thư viện của JavaScript.

Vì vậy, nếu học HTML, CSS, JS rồi học React (Javascript) sẽ dễ và “nhàn” hơn cho bạn (vì bạn không cần học TypeScript).

Tham khảo khóa học React với JavaScript [tại đây](#)

=> khóa học này yêu cầu bạn đã biết code & hiểu React với JavaScript.

Việc còn lại, **sử dụng TypeScript với React như thế nào, sẽ được hướng dẫn trong khóa học.**

2. Yêu cầu để học khóa học này

Tất cả yêu cầu bên dưới, là nền tảng để học tập hiệu quả.

Nếu như khóa học này bao gồm tất cả các vấn đề liệt kê, thì chắc chắn thời lượng sẽ rất dài và không tập trung vào kiến thức React sử dụng với Typescript.

Yêu cầu 1: Biết React.js làm website với JavaScript

Nếu bạn chưa biết gì về React.js làm website, có thể tham khảo:

- Khóa học miễn phí về React làm website [tại đây](#)
- Khóa học trả phí về React làm web sử dụng JavaScript [tại đây](#)

Yêu cầu 2: Biết cú pháp của Typescript

Việc sử dụng React với Typescript như thế nào, sẽ được mình hướng dẫn trong khóa học.

Tuy nhiên, bạn cần “tự học” cú pháp của Typescript trước.

Nếu bạn chưa biết gì về cú pháp của Typescript, học nhanh [tại đây](#)

Yêu cầu 3: Biết sử dụng Git để quản lý mã nguồn

Kiến thức về Git sẽ giúp bạn 2 việc quan trọng:

- Có khả năng backup code của chính bạn, tránh trường hợp máy tính bị hư hỏng, dẫn tới mất code. Ngoài ra, khi cần mình support, bạn có thể gửi project cho mình
- Bạn có khả năng sử dụng code mà khóa học cung cấp

Nếu bạn chưa biết gì về Git, xem khóa học Git miễn phí [tại đây](#)

3. Khóa học này nằm ở đâu trong lộ trình Frontend của tác giả Hỏi Dân IT

Khóa học này là khóa học thứ 2 , sau khi bạn đã học xong [khóa học React Cơ Bản](#)

Chi tiết về lộ trình Frontend React của tác giả Hỏi Dân IT, tham khảo [tại đây](#).

#4. Demo kết quả đạt được

Mục tiêu của khóa học: chỉ cần bạn biết code React.JS với JavaScript, code React với TypeScript như thế nào, sẽ được hướng dẫn chi tiết trong khóa học.

Link demo sản phẩm đạt được, tham khảo tại đây: <https://cv.hoidanit.vn/>

1. Công nghệ sử dụng

- Cấu hình dự án React với TypeScript và Vite từ số 0
- Migrate từ javascript sang typescript

2. Học viên nào có thể học ?

Học viên cần trang bị các kiến thức sau trước khi theo học:

- Có kiến thức về HTML, CSS và JavaScript.

- **Đã học (đã biết) React với JavaScript.**

Khóa học này không phải là học React từ số 0.

Nếu bạn chưa biết gì về React, tham khảo khóa học React cơ bản [tại đây](#)

- Có hiểu biết về cú pháp của TypeScript

Lưu ý:

Source code cả khóa học (full project) được cung cấp (cũng như cung cấp theo từng video hướng dẫn)

#5. Sử Dụng Source Code của Khóa học

Đây là khóa học thực hành, mình “KHÔNG khuyến khích” sử dụng source code.
Vì nếu các bạn “không tự code”, kiến thức không bao giờ (never) là của các bạn.

Mục đích video này ra đời, để phục vụ trường hợp: bạn muốn có source code của cả khóa học (mà không cần code), hoặc muốn có source code ứng với video bạn đang xem.

Bạn xem video này cho biết, chỉ sử dụng khi nào cần, còn source code thực hành khóa học sẽ được mình cung cấp tại các video tiếp theo (cứ xem video lần lượt)

Link source code **dự án Todo TypeScript**, xem [tại đây](#)

Lưu ý: Bạn vẫn cần xem video để biết cấu hình các tham số môi trường (nếu có)

Link source code **dự án Portfolio TypeScript**, xem [tại đây](#)

Lưu ý: Bạn vẫn cần xem video để biết cấu hình các tham số môi trường (nếu có)

1. Cách xem source code theo từng video thực hành

Lưu ý: Lựa chọn đúng branch của code (trong khóa học này là master)

Cách 1: download source code cuối mỗi bài giảng

Tại mỗi video bạn xem, mình có để link để download nhanh source code.

//todo (minh họa với video [#33](#))

Cách 2: sử dụng git (khuyến khích dùng cách này)

git checkout commit_hash

#6.1 Cách Dùng Udemý - Hỗ Trợ Hỏi Đáp Q&A

Lưu ý: không bỏ qua video này. Xem để biết cách sử dụng Udemý, cũng như cách 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

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

Xem hướng dẫn tài liệu chi tiết [tại đây](#)

#6.2 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](#)

Về chuyện leak khóa học và mua lậu

Mình biết rất nhiều bạn khi học khóa học này của mình, là mua lậu qua bên thứ 3. chuyện này là hoàn toàn bình thường, vì thương hiệu “Hỏi Dân IT” đang ngày càng khẳng định được vị thế của mình.

Nhiều bạn hỏi mình, sao mình không ‘chặn việc mua lậu’. nói thật, nếu mình làm, là làm được đấy, cơ mà nó sẽ gây ra sự bất tiện cho học viên chân chính (con sâu làm rầu nồi canh). Với lại, ngay cả hệ điều hành windows, còn bị crack nữa là @@

Mình cũng có 1 bài post facebook về chuyện này:

<https://www.facebook.com/askitwitheric/posts/pfbid02gyasktd3semqxt6nevnvwh4c8epzu3i7kpzhr7s7gmmfcvucyz96eb8avnvgnhl>

Với các bạn học viên chân chính, mình tin rằng, những cái các bạn nhận được từ mình khi đã chấp nhận đầu tư, nó sẽ hoàn toàn xứng đáng. vì đơn giản, với cá nhân mình, khách hàng là thượng đế.

VỚI CÁC BẠN MUA LẬU, MÌNH CHỈ MUỐN CHIA SẺ THẾ NÀY:

1. TRÊN ĐỜI NÀY, CHẴNG CÓ GÌ CHẤT LƯỢNG MÀ MIỄN PHÍ CẢ.
VIỆC BẠN MUA LẬU QUA BÊN THỨ 3, LÀ GIÚP BỌN CHÚNG LÀM GIÀU VÀ GÂY THIẾT HẠI CHO TÁC GIẢ.
NEẾU NHÌN VỀ TƯƠNG LAI => CÀNG NGÀY CÀNG ÍT TÁC GIẢ LÀM KHÓA HỌC => NGƯỜI BỊ HẠI CUỐI CÙNG VẪN LÀ HỌC VIÊN
2. HÃY HỌC THÓI QUEN TRÂN TRỌNG GIÁ TRỊ LAO ĐỘNG
NÓ LÀ THÓI QUEN, CŨNG NHƯ SẼ LÀ MỘT PHẦN TÍNH CÁCH CỦA BẠN.
ĐỪNG VÌ NGHÈO QUÁ MÀ LÀM MẤT ĐI TÍNH CÁCH CỦA BẢN THÂN.
NEẾU KHÓ KHĂN, CỨ INBOX MÌNH, MÌNH HỖ TRỢ. VIỆC GÌ PHẢI LÀM VẬY =))
3. MÌNH ĐÃ TỪNG LÀ SINH VIÊN GIỐNG BẠN, MÌNH HIỂU TẠI SAO CÁC BẠN LÀM VẬY. HÃY BIẾT CHO ĐI. SỐNG ÍCH KỶ, THÌ THEO LUẬT NHÂN QUẢ ĐẤY, CHẴNG CÓ GÌ LÀ NGẪU NHIÊN CẢ
4. NEẾU BẠN THẤY KHÓA HỌC HAY, HÃY BIẾT DONATE ĐỂ ỦNG HỘ TÁC GIẢ. LINK DONATE: <https://hoidanit.vn/donate>

Hành động nhỏ nhưng mang ý nghĩa lớn. Hãy vì 1 cộng đồng IT Việt Nam phát triển. Nếu làm như các bạn, có lẽ chúng ta đã không có Iphone, không có Apple như ngày nay rồi @@

Chapter 2: Setup Environment

Cài đặt & chuẩn bị môi trường thực hiện dự án

#7. Chuyện Cài Đặt Công Cụ (Bắt Buộc Xem)

1. Mục đích

Chương học này sẽ hướng dẫn chi tiết cách cài đặt các công cụ cần thiết để phục vụ cho khóa học. Vì vậy, **bạn vui lòng không bỏ qua video nào, xem lần lượt theo thứ tự**

Có 2 sai lầm mà các bạn hay gặp phải, đặc biệt là những bạn “đã biết 1 chút”

Sai lầm 1: Bỏ qua các video cài đặt công cụ vì bạn cho rằng bạn “đã biết rồi”

Hãy nhớ rằng, cài công cụ là 1 phần, đang còn phải “cấu hình” nó nữa.

Khóa học được sinh ra, và đã tối ưu. Bạn chỉ dành 5 tới 10 phút để xem video, đổi lại tiết kiệm cho bạn cả giờ đồng hồ ngồi mò mẫm.

Sai lầm 2: Không quan tâm tới version của phần mềm

Khi thực hiện khóa học, **bạn vui lòng download và cài đặt version phần mềm giống như video**. Điều này sẽ đảm bảo môi trường thực thi code là giống nhau. (hạn chế tối đa bug có thể xảy ra)

2. Version Phần Mềm theo thời gian

Bạn đừng sợ phần mềm (công cụ) nó thay đổi version, hay thậm chí là “chê bai” version cũ. Vì vốn dĩ, công nghệ nó là vậy, luôn thay đổi theo thời gian.

Bạn yêu cầu “version mới nhất” cho cái bạn học, mình đã làm điều đó tại thời điểm quay video khóa học, tuy nhiên, sẽ là cố định 1 version.

Lý do: công nghệ sẽ cập nhật theo thời gian. Cho dù bạn muốn hay không, hoặc thậm chí lắp tên lửa vào đít, đuổi cũng không kịp.

Điều bạn cần làm là: học 1 version, và quan trọng hơn, là bạn học, bạn cần hiểu nó

Sau đấy, nếu cần thiết, bạn học version mới hơn. Điểm khác biệt ở đây, là khi học version mới, bạn không phải là người bắt đầu từ số 0 (do đã có base từ version cũ)

Chỉ không học version cũ khi và chỉ khi: sản phẩm của version cũ không dùng được

Đây là lý do tại sao các khóa học của mình, khi học xong, mình mới hướng dẫn nâng cấp version.

#8. Cài đặt Node.js

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 [tại đây](#)

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

#9. Cài đặt Visual Studio Code

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

Link download:

<https://code.visualstudio.com/download>

#10. 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

#11. 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/Editor 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 @@

#12. 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.

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.

#13. Cài đặt Google Chrome

Chúng ta code mobile, tuy nhiên, để có dữ liệu, vẫn cần website để tạo và quản lý dữ liệu đấy.

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 [tại đây](#)

Chapter 3: React với TypeScript

Sử dụng React sử dụng ngôn ngữ TypeScript từ số 0

#14. Tổng quan về chapter

//todo

#15. Why Coding (Frontend) with Typescript

Tài liệu: <https://npm trends.com/typescript>
<https://www.npmjs.com/package/typescript>

React, Vue => (optional) Vanilla Javascript / Typescript

Angular => only Typescript

1. Tại sao chọn Javascript ?

- Vì javascript phổ biến, dễ học (phù hợp với beginners, chưa từng học ngôn ngữ OOP)
- Rút ngắn thời gian phát triển sản phẩm (code ít hơn typescript)
- Phù hợp với dự án nhỏ /team nhỏ
 - => không quan tâm tới việc mở rộng/tối ưu/bảo trì code
 - => code cho chạy được
 - => người code không cần IDE gợi ý => cần hiểu mình đang làm gì

2. Khi nào cần Typescript (why) ?

- Dự án lớn (có nhiều thành viên tham gia/nhiều chức năng)
 - => khối lượng codebase lớn
 - => quan tâm tới việc mở rộng/tối ưu/bảo trì code
- Gợi ý code (từ IDE) khi code
- Thông báo lỗi khi dịch code (compile), not runtime (sau khi đã build và chạy)
 - => chất lượng sản phẩm cao hơn, ít bugs

3. Những khó khăn khi dùng với Typescript ?

- Biết javascript là chưa đủ => cần học cú pháp của typescript
- Code nhiều hơn javascript

Đổi lại:

- Gợi ý code (từ IDE) khi code
 - Thông báo lỗi khi dịch code (compile), not runtime (sau khi đã build và chạy)
- => chất lượng sản phẩm cao hơn, ít bugs

4. Để học 1 framework/library với Typescript cần gì ?

- Có hiểu biết về javascript/typescript
- Có hiểu biết về OOP. Nếu bạn đã biết các ngôn ngữ như C#, Java... đây là lợi thế
- Có khả năng đọc source code (optional)
- Search Google (always) => làm nhiều thì quen tay

Give it a try ?

- Bởi vì có thể code song song 1 dự án, dùng cả Javascript lẫn Typescript :v

- Có thể code javascript trong file typescript, còn ngược lại thì không.
(Không thể sử dụng cú pháp TS trong file JS)

Xu hướng chuyển dịch của thư viện/framework đang là Typescript ???

Dev dùng typescript => đỡ bugs

Máy (dùng javascript, version đã compile từ mã nguồn TS được viết)

=> như vậy javascript không die, mà chẳng qua, cách code javascript cần khác đi

=> nâng cao chất lượng sản phẩm, đảm bảo tính scale/maintain (các doanh nghiệp cần cái này)

#16.1 Ôn tập kiến thức React với Javascript

Tài liệu: <https://react.dev/>

Các kiến thức về React cần nắm vững:

- Components (JSX)
- State
- Props
- Hook (useState, useEffect)

Nếu bạn chưa biết gì về React với JavaScript, tham khảo khóa học React Ultimate (sử dụng HTML, CSS và JavaScript) [tại đây](#)

#16.2 Ôn tập Kiến Thức TypeScript

TypeScript = JavaScript + Khai báo type

Tài liệu:

<https://www.typescriptlang.org/docs/handbook/2/everyday-types.html>

Yêu cầu: Nắm được cách khai báo các data type và sử dụng function với TypeScript

Khai báo object với data type/interface

Định nghĩa function

Nếu bạn chưa biết gì về TypeScript, tham khảo khóa học cú pháp TypeScript cơ bản [tại đây](#)

#17. Tài liệu sử dụng React với Typescript

Tài liệu: <https://react.dev/learn/typescript>

Các nguồn học cú pháp của React với TypeScript:
<https://react.dev/learn/typescript#further-learning>

Thực chất là chúng ta khai báo component bình thường (giống với javascript), chỉ có điều định nghĩa thêm type cho typescript

Nếu bạn sử dụng các framework hoặc công cụ để tạo dự án React, đa phần đã được cấu hình sẵn về TypeScript (không cần cài đặt). Việc của bạn là viết code sử dụng TypeScript

#18. Cài Đặt 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 [#8](#))

2. Cài đặt dự án thực hành

Lưu ý : đường dẫn nơi lưu dự án thực hành, không nên đặt tên tiếng việt để hạn chế lỗi các bạn nhé

Ví dụ: D:/dự án/react-typescript => thay bằng: D:/du-an/react-typescript

Bước 1: clone 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 start (hoặc câu lệnh : npm run dev)

#19. 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 [#18](#)

Nếu như bạn tự ý code “theo cách bạn hiểu” và không sử dụng source code tại [#18](#) mình sẽ không hỗ trợ support khi có lỗi xảy ra

Tài liệu: <https://vite.dev/guide/>

#20. Hello World với React TypeScript

// hướng dẫn viết hello world

//giải thích cấu trúc dự án thực hành

//hướng dẫn cách đẩy code lên github/gitlab của chính bạn

Nếu bạn chưa biết gì về Git, học ngay và luôn [tại đây](#)

Bước 1: Tạo repository trên github/gitlab

Bước 2: Đẩy dự án lên repository vừa tạo

git add .

git commit -m "init project"

<https://stackoverflow.com/questions/2432764/how-do-i-change-the-uri-url-for-a-remote-git-repository>

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

git push origin master

Bước 3: Minh họa khi code xong 1 video, chỉ cần sử dụng 3 câu lệnh bên dưới

git add .

git commit

git push

#21. Component

Yêu cầu: (với javascript)

- Đã hiểu tác dụng của component
- Đã biết cách khai báo và sử dụng component
- Sử dụng function component

//code javascript với React Vite (Typescript)

<https://stackoverflow.com/a/78276639>

1. Convert từ javascript sang typescript

- thay vì định nghĩa file .js (.jsx) => định nghĩa .tsx
- tên component sẽ viết Hoa chữ đầu tiên

Cách 1:

```
const App = () => {  
    return ( <div> Hello world ts </div> )  
}
```

Cách 2:

```
const App = () => <div> Hello world ts </div>
```

Lưu ý về code: khuyến khích viết theo cách 1, như vậy cấu trúc rõ ràng (giống OOP), đồng thời, sau này cần thêm logic xử lý thì không cần sửa lại cấu trúc function

- Chuyển sang function component với typescript: giống hệt cách kể trên
- Với cách viết 1 ở trên, typescript đủ thông minh để biết đây là component (khi return JSX element)

2. Lưu ý về cách định nghĩa function component:

Không nên dùng các cách sau:

Link giải thích lý do:

<https://github.com/facebook/create-react-app/pull/8177>

```
const App: React.FunctionComponent<{}> = () => (  
  <div> Hello world ts </div>  
)
```

hoặc

```
const App: React.FC = () => (  
  <div> Hello world ts </div>  
)
```

hoặc

```
const App: React.VoidFunctionComponent<{}> = () => (  
  <div> Hello world ts </div>  
)
```

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#22. React Function Component/React.FC (Extra)

1. Giải thích về implicit vs explicit

Implicit programming refers to processes happening behind the scenes in your code. It relies on default behaviors, type inference, and built-in functions to handle tasks without you having to specify every detail.

Explicit programming is all about clearly and specifically stating what you want your code to do. It involves directly specifying types, function return values, and other details in your code rather than relying on the language to infer or assume anything.

<https://solverfox.dev/writing/no-implicit-children/>

Explicit props are the ones that are written out in the props interface.

Implicit props are the ones that **@types/react** automatically adds.

<https://react.dev/blog/2024/04/25/react-19-upgrade-guide#removed-proptypes-and-defaultprops>

<https://www.totaltypescript.com/you-can-stop-hating-react-fc>

Bạn có thể sử dụng React.FC và function component, tùy thuộc vào sở thích và thói quen của bạn.

Tuy nhiên, mình khuyến khích “đơn giản nhất có thể”, nên không “bắt buộc” phải sử dụng các công cụ trên.

#23. Props

//hardcode data tại component cha

```
const todos = [  
  {  
    id: 1,  
    title: "Learn React TypeScript",  
    isComplete: false  
  },  
  {  
    id: 2,  
    title: "Subscribe Youtube HoiDanIT",  
    isComplete: true  
  },  
  {  
    id: 3,  
    title: "Learn English",  
    isComplete: true  
  },  
]
```

//sử dụng props dưới dạng any

//viết props dưới dạng type/interface

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#24. Type vs Interface (Extra)

Tham khảo các topics sau:

<https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#interfaces>

<https://www.typescriptlang.org/play/?#example/types-vs-interfaces>

<https://stackoverflow.com/questions/37233735/interfaces-vs-types-in-typescript/52682220#52682220>

#25. Default/Optional Props

//todo

Tạo component todo.input.tsx

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#26. TypeScript Generics (Extra)

Tài liệu:

<https://www.typescripttutorial.net/typescript-tutorial/typescript-generics/>

https://www.w3schools.com/typescript/typescript_basic_generics.php

=> generic được sử dụng, khi "chưa biết trước kiểu type" (được dùng nhiều khi viết function tổng quát/reusable code)

Ví dụ:

```
//function useState<S>(initialState: S | (() => S)): [S, Dispatch<SetStateAction<S>>];
```

Hàm **useState** là 1 **generic**, có thể truyền vào initialState với kiểu type (S - chưa biết trước).

hàm này sẽ trả về 2 giá trị [S, Dispatch<SetStateAction<S>>]

=> đây là 1 array, phần tử đầu tiên là biến với type = S, phần tử thứ 2 là function giúp cập nhật giá trị.

Cụ thể:

```
const [name, setName] = useState("eric");
```

ở đây, biến truyền vào có kiểu giá trị là "string"

```
useState<string>(initialState: string | (() => string)): [string,  
Dispatch<SetStateAction<string>>];
```

Do hàm useState trả về 1 array theo cú pháp trên => const [name, setName] =... có nghĩa là:

biến name (phần tử đầu tiên) sẽ có kiểu giá trị là string

biến setName sẽ có khả năng cập nhật được biến name, do kiểu giá trị là Dispatch<SetStateAction<string>>

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#27. Sử dụng State

//todo

//list Todo là state của cha, từ cha truyền xuống con

//sử dụng event với typescript

<https://stackoverflow.com/a/42645711>

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#28. Truyền function từ cha sang con

//tính năng add todo

//random number between range

<https://stackoverflow.com/a/29246176>

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#29. Bài tập Delete Todo

//todo

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#30. Tổng kết về chapter

Kiến thức trọng tâm đã học, là **migrate (chuyển đổi) từ code React JavaScript sang React với TypeScript**, bao gồm:

- Component
- Props
- State
- Thực hành qua ứng dụng Todo

Chapter 4: Thực Hành Dự Án Portfolio

Thực hành và luyện tập React với TypeScript với dự án Portfolio

#31. Tổng quan về chapter

Mục tiêu: thực hành code React với TypeScript

#32. Dự án thực hành (JavaScript)

Mục tiêu: test thành quả đạt được (mục tiêu cần làm)

Bước 1:

Clone dự án [tại đây](#)

Bước 2: Cài đặt thư viện

npm i

Bước 3: chạy dự án

npm run dev

//todo

Giải thích cấu trúc dự án thực hành

#33. Tích hợp TypeScript vào dự án React JavaScript

Tài liệu:

<https://stackoverflow.com/a/75192263>

<https://github.com/vitejs/vite/discussions/6799#discussioncomment-4557486>

Bước 1: Cài đặt các thư viện hỗ trợ cho typescript

**npm i --save-dev --save-exact @types/node@22.7.5 typescript@5.6.3
typescript-eslint@8.8.1**

Bước 2: Cấu hình cho TypeScript

//todo

Ghi đè các file từ dự án React Vite Typescript, bao gồm tsconfig, eslint...

Mẹo: lấy các package từ dự án React Vite Typescript

Với typescript, chỉ dùng cho developer (không liên quan tới browser - javascript), nên chỉ cài đặt ở dev Dependency

Bước 3: Test dự án

//todo

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#34. Hero Section (TypeScript)

```
declare module "*.png";  
declare module "*.jpg";  
declare module "*.jpeg";  
declare module "*.svg";  
declare module "*.gif";
```

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#35. TypeScript Absolute Import

Mục tiêu: import sử dụng absolute path

Bước 1: cấu hình Vite

`npm i --save-exact --save-dev vite-tsconfig-paths@5.0.1`

<https://stackoverflow.com/questions/68241263/absolute-path-not-working-in-vite-project-react-ts>

<https://github.com/aleclarson/vite-tsconfig-paths>

Bước 2: khai báo Typescript `baseUrl` và `path`

Về typescript path: <https://www.typescriptlang.org/tsconfig/#paths>

Bước 3: test import

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#36. Hoàn thiện HomePage Screen

<https://www.typescriptlang.org/docs/handbook/triple-slash-directives.html>

//bổ sung file vite-env.d.ts

```
/// <reference types="vite/client" />
```

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#37. React Context với TypeScript

Tài liệu: <https://react.dev/reference/react/createContext#creating-context>

JSX children props:

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

Sử dụng với Context với typescript:

<https://react-typescript-cheatsheet.netlify.app/docs/basic/getting-started/context/>

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#38. useRef với TypeScript

Miêu tiêu: làm tính năng auto-scroll tới section mong muốn

<https://stackoverflow.com/a/52528619>

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#39. Hoàn thiện Project Screen

//todo

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#40. Hoàn thiện About Screen

//todo

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

#41. Tổng kết về chapter

//todo

//upload CV lên google drive, view online

Bước 1: Upload CV dưới định dạng .pdf

Bước 2: Share Link

//nhớ test link trước khi share

<https://stackoverflow.com/a/63627688>

<https://stackoverflow.com/a/74457365>

Về source code video này:

Các file thay đổi tại video, xem [tại đây](#)

Source code ứng với video này, xem [tại đây](#)

Video hướng dẫn cách download source sau khi kết thúc video này, xem [tại đây](#)

Chapter 5: Deploy Production - Triển Khai Dự Án Thực Tế

Hướng dẫn triển khai dự án thực hành với tên miền thực tế

#42. Tổng quan về chapter

//todo

#43. Create Vercel Account

//todo : nếu bạn đã có tài khoản Vercel rồi, có thể bỏ qua video này

#44. Link Vercel với Github

//todo

#45. Triển khai dự án React Vite với Vercel

//update source code frontend

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

Cần chuẩn bị:

- Tài khoản github, và đã đẩy source code lên github
- Đã có tài khoản Vercel, và liên kết với Github thành công

Bước 1: đăng nhập vào Vercel

Trang chủ: <https://vercel.com/>

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

Bước 2: Triển khai với Vercel

Lưu ý: cấu hình cài đặt thư viện với yarn

Sử dụng câu lệnh cài đặt sau: **yarn install**

#46. Mua Tên Miền (Mắt Bão)

Tài liệu: <https://www.matbao.net/>

Trong khóa học này, mình lựa chọn “mắt bão” để mua tên miền, một nhà cung cấp hàng đầu trong nước (vì nó dễ thanh toán & chấp nhận sử dụng ngân hàng nội địa)

Nếu bạn có thẻ visa/master card, có thể mua với nhà cung cấp nước ngoài.

Lưu ý:

Trong quá trình mua tên miền, nếu bạn gặp khó khăn gì, thì đừng ngại gọi support của mắt bão nhé. Thông tin nó ghi hết trên trang chủ ấy.

Yên tâm là Mắt Bão thuộc top đầu của Việt Nam về kinh doanh tên miền các bạn nhé.

Support 24/7

1900 1830 (1000 đ/phút)

Tư vấn miền Nam

(028) 3622 9999

Tư vấn miền Bắc

(024) 35 123456

#47. Mapping tên miền tới Vercel

Yêu cầu: bạn đã mua và sở hữu tên miền (domain)

Bước 1: Cập nhật bản ghi với tên miền

Tài liệu: <https://vercel.com/guides/a-record-and-aaa-with-vercel>

Cập nhật bản ghi A, với ip: **76.76.21.21**

Nơi thực hiện: trang quản lý của đối tác bạn mua tên miền

Bước 2: Tạo custom domain

Nơi thực hiện: tại project bạn hosting trên Vercel

Nếu có lỗi phát sinh vui lòng chờ đợi tối đa 24h để chắc chắn mọi thứ sẽ hoạt động chính xác bạn nhé.

//fix lỗi khi vào website, mặc định là theme dark

//app.context.tsx

```
const [theme, setTheme] = useState<ThemeContextType>(() => {  
  const initialTheme =  
    (localStorage?.getItem("theme") ?? "dark") as ThemeContextType;  
  return initialTheme;  
});
```

#48. Cách tự code dự án React với TypeScript của bạn

1. Về ý tưởng

Tham khảo các website về làm portfolio:

Template 1:

<https://github.com/soumyajit4419/Portfolio>

Template 2:

<https://github.com/said7388/developer-portfolio>

2. Về kiến thức

Đọc lại các cú pháp với typescript

Tham khảo các source codebases:

<https://react-typescript-cheatsheet.netlify.app/docs/basic/recommended/codebases/>

#49. What's next ?

Nhận xét về ưu/nhược điểm:

- Thực hành dự án React với TypeScript
- Hướng dẫn cách dùng typescript với React
- Cần phải thực hành nhiều hơn nữa

Về lộ trình frontend của Hỏi Dân IT, các bạn xem [tại đây](#)

Giải thích về lộ trình (cách học), các bạn xem [tại đây](#)

Đánh Giá (Review/Rating) Khóa Học

1. Lời cảm ơn:

Mình rất hy vọng bạn thấy khóa học này có giá trị, nhưng dù thế nào đi nữa, hãy để lại đánh giá và chia sẻ trải nghiệm của bạn.

Hãy cho mình biết bạn nghĩ gì và viết đánh giá khi bạn có thể.

Mình cũng luôn sẵn sàng trả lời câu hỏi của bạn - hãy gửi tin nhắn trực tiếp cho mình bất cứ lúc nào.

2. Cách đánh giá (review) khóa học, tham khảo chi tiết [tại đây](#)

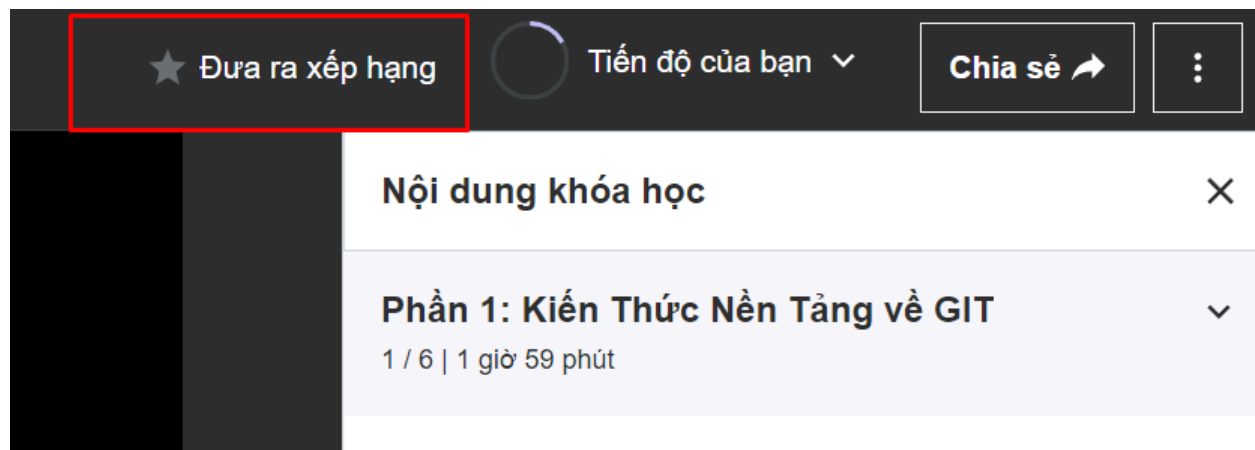
Tạo review đánh giá:

Nhấn vào “đưa ra xếp hạng”. Sau đây rating (star) và viết đánh giá chia sẻ trải nghiệm.

Nếu màn hình của bạn không hiển thị, có 2 nguyên nhân:

1 là bạn đã đánh giá rồi, xem mục 3 bên dưới để biết cách sửa/xóa đánh giá

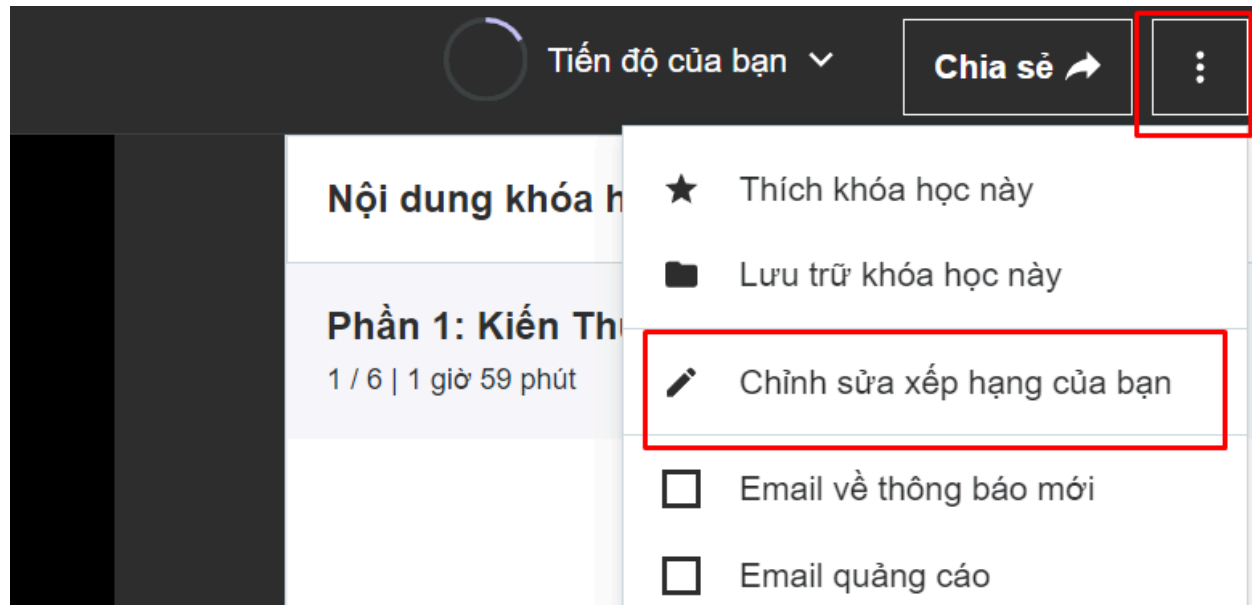
2 là thời gian bạn xem khóa học chưa đủ nên không thể đánh giá



3. Chỉnh sửa/xóa review đã đánh giá:

Bước 1: chọn vào hình 3 dấu chấm tại góc màn hình

Bước 2: chọn chỉnh sửa review



4. Mẹo để Viết Đánh Giá Tốt

Dưới đây là một số điều bạn nên lưu ý khi viết đánh giá cho một khóa học:

Hãy cho chúng tôi biết lý do: Ngoài việc để lại đánh giá sao, vui lòng chia sẻ suy nghĩ của bạn.

Ý kiến của bạn về khóa học rất có giá trị đối với những người học khác, nhưng khi chỉ có đánh giá sao, thật khó để người dùng khác hiểu vì sao bạn lại đưa ra đánh giá đó.

Cụ thể là tốt: Tính cụ thể giúp người học khác quyết định xem khóa học có phù hợp với họ không. Có điểm nào cần cải thiện không?

Khóa học có đáp ứng được kỳ vọng của bạn không?

Phần nào của khóa học khiến bạn thích nhất?

Trung thực: Đánh giá là một trong những yếu tố quan trọng nhất mà mọi người xem xét trước khi đăng ký một khóa học.

Tuy nhiên, đánh giá chỉ có giá trị khi bạn trung thực về cảm nhận của mình về khóa học.

Miễn là bạn chia sẻ cảm nhận một cách tôn trọng, ý kiến phản hồi của bạn rất có giá trị và hữu ích cho cộng đồng học tập của chúng tôi.

Lời Kết

Như vậy là chúng ta đã cùng nhau trải qua hơn 50+ video về sử dụng React với TypeScript dành cho front-end 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 với TypeScript.

Dĩ nhiên rằng, trong quá trình quá trình thực hiện khóa học này, mình sẽ không thể tránh khỏi những sai sót.

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)