

TÀI LIỆU KHÓA HỌC “REACT TEST FRESHER”

Tác giả: Hoi Dân IT

Version: 6.0

Note Update:

- Cập nhật Chapter 8 : fix bugs học viên report trong quá trình học
- Update source code #63

Chapter 0: Giới Thiệu Về Khóa Học	4
#0.0. Demo Kết Quả Đạt Được Khi Kết Thúc Khóa Học	4
#0.1. Hướng Dẫn Sử Dụng Khóa Học Này Hiệu Quả	5
#0.2. Yêu Cầu Của Khóa Học	6
#0.3. Về Khóa Học này	7
#0.4. Về Tác Giả	9
Chapter 1: Setup Environment	10
#1. Lựa chọn công nghệ	10
#2. Công cụ sử dụng	11
#3.1 Setup Base Dự Án React (Redux Toolkit)	12
#3.2 Quá Trình Setup React và Redux	13
#4.1 Tích hợp React Router	14
#4.2. Chia layout và project structure	15
#5. Cài đặt MongoDB Compass	16
#6. Tạo Tài Khoản MongoDB Atlas & Database	17
#7. Check kết nối tới MongoDB từ Localhost	18
#8. Setup dự án Backend	19
#9.1. Cấu trúc dự án Thực Hành	20
Chapter 2: Module Auth	21
#9.2. Các Kiến Thức Sẽ Học	21
#10. Cài đặt thư viện UI Antd - Ant Design	22
#11. Giới Thiệu Form Component	23
#12. Bài tập Design Register Form	24
#13. Axios và Axios Customize	25
#14. Bài tập chức năng Register	26
#15. Bài tập Chức Năng Login	27
#16. Lưu Access Token và Refresh Token	27
#17. Store User Login with Redux Toolkit	28
#18.1 Protected Page	29
#18.2 Check quyền hạn của người dùng	29
#19.1 Bài Tập Design Layout	29
#19.2 Fix bug Loading data	30
#20. Axios Retry - Handle Refresh Token	31
#21. Fix Lỗi Hết Hạn Refresh Token	31
Chapter 3: Module Users	32
#22. Giới Thiệu Table Component	32
#23. API Hiển Thị Danh Sách User	32
#24. Bài Tập Hiển Thị Danh Sách User	33
#25. API Filter	34
#26. Bài Tập Filter Danh Sách User	35
#27. API Sorter	36
#28. Bài tập Sorter	37
#29.1 Bài tập chức năng View Detail User	38

#29.2 Bài tập Add avatar	38
#30. Bài tập chức năng thêm mới user	39
#31. Design modal upload	40
#32. Read File Excel	41
#33. Bài tập Chức năng Import user	42
#34. Download Sample File	42
#35. Export to CSV	43
#36. Bài tập chức năng Export user	43
#37. Bài tập chức năng Update user	43
#38. Bài tập chức năng Delete user	43
#39. Ôn tập các kiến thức đã học	44
Chapter 4: Module Books	45
#40. Bài tập Hiển Thị/Filter/Sort Books	45
#41. Bài tập design chức năng xem chi tiết Book	46
#42. Hoàn thiện chức năng xem chi tiết Book	47
#43. Design Form Add New Book	48
#44. Bài tập Upload file	49
#45. Bài tập Create a new book	50
#46. Bài tập Design form update book	51
#47. Bài tập Update a book	52
#48. Bài tập Delete a book	52
#49. Tổng Kết Các Kiến Thức Đã Học	53
#50. Bài tập Design giao diện client (homepage)	54
#51. Hiển thị HomePage	55
#52. Bài tập Filter/Sorter	56
#53. Url với Params và Query	57
#54. Bài tập Design View Detail Page	58
#55. Chức Năng View Detail Book	59
#56. Thêm Loading Skeleton	60
#57. Hoàn thiện View detail page	61
#58. Quá trình xây dựng giao diện	62
Chapter 5: Module Orders	63
#59. Setup Redux Persist	63
#60. Bài tập Add/Remove Product	64
#61. Bài tập hiển thị số lượng giỏ hàng	64
#62. Design Preview Cart	65
#63. Design base order page	66
#64. Bài tập order page	66
#65. Bài tập Payment Page	67
#66. API Create An Order	68
#67. Bài tập View Order History	69
#68. Bài tập Update Current User	70
#69. Bài tập Display Order page + API Dashboard	71
#70. Các bugs còn tồn đọng	72

Chapter 6: Triển Khai Dự Án Thực Tế	73
#71. Mô hình triển khai	73
#72. Chuẩn Bị Môi Trường Backend	74
#73. Triển Khai Backend với Render	75
#74. Chuẩn bị Môi trường Frontend	76
#75. Triển khai Frontend với Vercel	77
#76. Tại sao dùng Host Miễn Phí Không Lưu Cookies	78
#77. Mua tên miền	79
#78. Chạy app với tên Miền	79
Chapter 7: Góc Nhìn Về Level Fresher	80
#79. Chuyện Học 1 Khóa Học	80
#80. Level Fresher là level như thế nào	80
#81. Học gì tiếp đây ?	80
Chapter 8: Fix Bugs	81
#82. Fix Bug Filter	81
#83. Fix Bug Refresh Token	81
#84. Về Bug Warning của Antd	84

Chapter 0: Giới Thiệu Về Khóa Học

#0.0. Demo Kết Quả Đạt Được Khi Kết Thúc Khóa Học

Video demo : <https://youtu.be/z5ywIBpVusw>

Một vài highlight của khóa học:

- Đây là dự án thực hành (project cuối khóa) của series React Udemy (Hỏi Dân IT).
- Backend dự án này sẽ được cung cấp sẵn (bạn không cần viết backend), và backend này đảm bảo yêu cầu trong thực tế, ví dụ như cơ chế JWT (access token, refresh token...)
- Khóa học này chính là thời gian cho bạn chuyển hóa kiến thức, có thể kể đến như kiến thức React, cách sử dụng Redux, React Router và việc kết hợp render giao diện với dữ liệu lấy từ API backend.
- Sử dụng Ant Design để tiết kiệm thời gian CSS và viết Component React.

#0.1. Hướng Dẫn Sử Dụng Khóa Học Đây Hiệu Quả

1. Nếu bạn tự tin vào khả năng của bản thân mình, nên tự code project này.

Tự code bằng cách:

- Xem video demo để biết được website sẽ xây dựng trông như thế nào (lấy đó làm design)
- Demo website (coming soon...)
- Xem API postman để biết backend đang cung cấp dữ liệu như thế nào

2. Nếu bạn muốn được hướng dẫn, thì có thể:

- Xem theo thứ tự video để biết mình xây dựng website này như thế nào, tuy nhiên:

Cần tự hoàn thiện bài tập được giao, không nên phụ thuộc vào source code mẫu tại video

3. Nếu bạn không tự code dự án này, thì có thể:

Sau khi đã hoàn thiện, bạn nên tự làm lại, từ A tới Z. chỗ nào quên thì Google. Không nên nhìn code mẫu nữa.

Chỉ có thực hành thì mới có tiến bộ. Dự án này mình đã cung cấp API Backend rất xịn sò, đi làm, nó chỉ có hơn, chứ không kém.

Như vậy, bằng cách sử dụng công cụ mình cung cấp, các bạn đang tiệm cận với công việc thực tế đi làm sau này.

Vì vậy, hãy luyện tập nhé.

#0.2. Yêu Cầu Của Khóa Học

Các yêu cầu cần biết trước khi thực hành khóa học:

- Bạn cần có kiến thức cơ bản về React, mình không dạy tại khóa này, vì đã có khóa React Ultimate làm điều này.
- Có thức thức cơ bản về Redux, tương tự như ở trên, mình đã có khóa Redux Ultimate

Khóa học này chính là lúc bạn “tự thực hành”, các video trong khóa học chỉ mang tính chất định hướng, không phải hướng dẫn code từ A tới Z

#0.3. Về Khóa Học này

1. Tại sao mình làm khóa học này?

Mình làm project này để cho bạn có dự án thực hành (dành cho bạn nào không muốn học Backend, muốn có 1 backend đủ xịn sò).

Trước đây (version cũ của khóa học này), mình sử dụng APIs free và nó không đáp ứng được yêu cầu (không phản ánh đúng chất lượng của khóa học Hỏi Dân IT), vì vậy, mình đã tự viết ra phần backend này cho các bạn sử dụng.

Hi vọng qua project này (các bạn chủ động tự làm, mình chỉ định hướng), chúng ta sẽ hiểu rõ và hiểu sâu về React.

Vì đơn giản, practise make perfect :v

2. SOS thì làm sao

Với các bạn học viên Udemy, để đảm bảo quyền lợi, các bạn chủ động inbox qua fanpage Hỏi Dân IT để nhận được sự hỗ trợ trong quá trình học tập nhé:

<https://www.facebook.com/askITwithERIC/>

4. 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 “Hoi 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, CHẲNG CÓ GÌ CHẤT LƯỢNG MÀ MIỄN PHÍ CẢ.
VIỆC BẠN MUA LẬU QUA BÊN THỨ 3, LÀ GIÚP BỌN CHÚNG LÀM GIÀU VÀ GÂY THIỆT 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, CHẲNG 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.com.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 @@

#0.4. 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.com.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/>

Chapter 1: Setup Environment

#1. Lựa chọn công nghệ

1. Javascript hay Typescript ?

Với khóa học này, mình vẫn dùng javascript. Mình không sử dụng typescript vì chưa có khóa học nào hướng dẫn sử dụng typescript với React.

Nếu bạn đã biết Typescript, có thể dựa vào video mình định hướng và 'tự code' typescript, vì đơn giản, khóa học này là khóa thực hành mà :v

2. CRA (create-react-app) hay Vite

Việc sử dụng tool nào để tạo nên dự án React không quan trọng, điều quan trọng là nó mang lại sự thoải mái cho bạn.

Trong khóa học này, mình lựa chọn Vite để mang lại sự trải nghiệm mới (so với khóa React Ultimate sử dụng CRA)

3. Manage State's React

Để quản lý state của React, mình sử dụng Redux (cụ thể là Redux Toolkit), thay vì Context API

Khóa học này mình sử dụng Redux Toolkit thay vì Redux thuần (so với khóa React Ultimate). Lý do là update công nghệ cho các bạn thôi (yên tâm là mindset - cách tư duy giống hệt nhau nhé)

4. Responsive layout

Để tạo nhanh layout, mình sử dụng Antd thay vì Bootstrap (khóa react ultimate)

Mục đích: update công nghệ thay vì chỉ dùng mình Bootstrap. Sau này, mình sẽ có riêng khóa học về antd (cụ thể là antd pro và umijs), khóa học này chỉ học basic antd bằng cách sử dụng component viết sẵn (tương tự như việc sử dụng react-bootstrap).

Lưu ý: Điểm khác biệt dễ thấy khi dùng layout của antd so với bootstrap là layout sẽ lấy chuẩn là 24 columns, thay vì 12 columns.

#2. Công cụ sử dụng

- Công cụ Code: VSCode (cá nhân mình không dùng thêm extension nào đặc biệt, sử dụng cấu hình mặc định của VSCode)
Link download : <https://code.visualstudio.com/download>
- Công cụ quản lý mã nguồn: Git . Cần sử dụng git để cuối khóa có để triển khai dự án thực tế. Bạn nào chưa biết gì về Git, xem tại đây:
<https://www.youtube.com/watch?v=-BtolPy15fg&list=PLncHg6Kn2JT6nWS9MRjSnt6Z-9Rj0pAlo>
- Môi trường thực thi code: Node.JS .Trong khóa này, mình sử dụng node.js version 16.13.1
Thành ra, để hạn chế lỗi tối đa, mình khuyến khích các bạn sử dụng version trên (giống mình)
- Môi trường cài đặt thư viện: npm hoặc yarn
Thông thường, chúng ta hay dùng npm để cài đặt thư viện. Tuy nhiên, nếu trong trường hợp cài đặt thư viện bị lỗi, có thể thử với yarn .

Về yarn, mình giới thiệu và hướng dẫn sử dụng trong khóa học Debug React/Node.js
- Công cụ test code: Google Chrome
Sử dụng trình duyệt phổ biến nhất để test code, ngoài ra, đảm bảo đã cài đặt 2 extension phổ biến dùng với React là:
React Developer Tools
Redux DevTools

#3.1 Setup Base Dự Án React (Redux Toolkit)

Download source code:

<https://drive.google.com/file/d/1vQj28srlueJQsaTZtjcSW-xnbkcSmuPC/view?usp=sharing>

Link Github (backup): <https://github.com/haryphamdev/Udemy-Base-Test-Fresher>

Tài liệu:

<https://vitejs.dev/guide/>

<https://redux-toolkit.js.org/introduction/getting-started>

Một vài lưu ý khi “tự setup dự án React” : (chỉ thực hiện khi bạn tự tin vào bản thân mình, chưa biết gì thì kéo project theo link ở trên ~ khỏi phải suy nghĩ :v)

1. Môi trường cài đặt:

- Node.JS version 16.13.1 (không nhất thiết phải cài đặt chính xác version, cơ mà ít nhất là cùng version node 16)
- Vite mình sử dụng là version 4. React version 18.
- Nếu bạn dùng Node.JS (khác version 16), Vite (khác version 4), React (khác version 18), trong phần còn lại của khóa học, nếu có lỗi phát sinh, mình không chịu trách nhiệm đâu nhé . goodluck với anh Google ~.~

2. Các bước tự setup dự án (để có được dự án React chạy với Vite, kết hợp với Redux Toolkit)

Bước 1: vào trang chủ của Vite => đọc docs rồi làm theo

-> có được dự án React Vite

Bước 2: vào trang chủ của Redux Toolkit => đọc docs rồi làm theo => chạy project này tại local để xem cách thức hoạt động

Bước 3: ghép code từ project bước 2 vào project bước 1

- Cài đặt thư viện: `npm i --save @reduxjs/toolkit react-redux`
- Cấu hình store
- Gọi component sử dụng Redux

Lưu ý: khi sử dụng Redux Toolkit, chỉ cần cài đặt 2 thư viện trên, không cần cài đặt thêm thư viện devtool hay redux thunk... (vì đã được tích hợp sẵn trong redux toolkit)

Bước 4: Test project hoàn chỉnh

#3.2 Quá Trình Setup React và Redux

Đây là video mình quay lại quá trình setup base dự án React (Vite) cùng với Redux toolkit.

Nếu bạn chưa có kỹ năng tự fix bugs, vui lòng không thực hiện theo video này, chỉ xem cho biết, và sử dụng source code mình cung cấp ở video #3.1

Còn nếu bạn muốn thử thách, có thể tự setup dự án yêu thích ứng với công nghệ tương ứng,

tuy nhiên, trong quá trình thực hiện khóa học, nếu có gì không tương thích hoặc có bugs, mình không giúp được đâu nhé. Lúc này chỉ good luck với google thôi :v

Tóm lại là:

- Không biết gì, không cần thực hiện như video này, và sử dụng source code ở video #3.1
- Muốn thử thách. Go ahead and good luck :v

#4.1 Tích hợp React Router

Tài liệu: <https://reactrouter.com/en/main>

Cài đặt thư viện:

```
npm install --save-exact react-router-dom@6.9.0
```

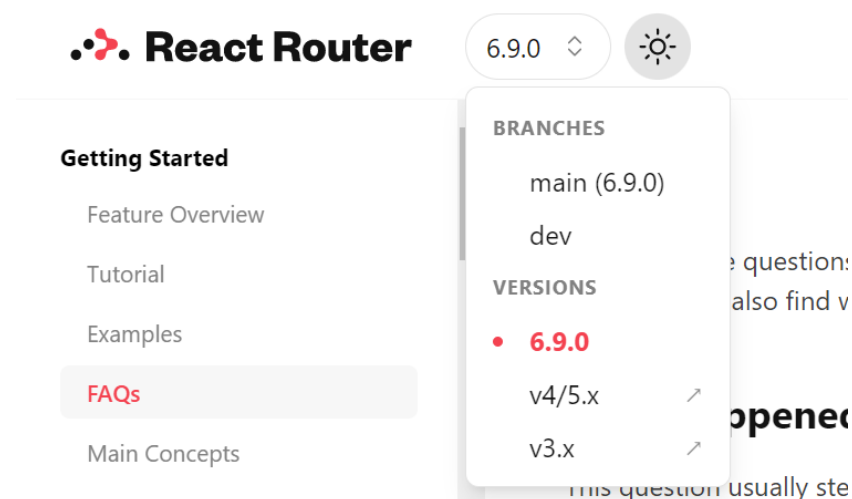
```
yarn add react-router-dom@6.9.0
```

1. Lưu ý về React Router

Version của React Router thay đổi theo thời gian, nếu mà chạy theo công nghệ, lắp tên lửa vào đuôi cũng không kịp.

Giải pháp để giải quyết vấn đề trên là bạn học và hiểu 1 version cụ thể (vì mindset nó không đổi giữa các version).

Sau này, nếu chẳng may thư viện nó thay đổi, cần tìm đúng Component của thư viện, rồi đọc docs của nó, hoặc làm theo cách hiểu của bạn. Không thể trông chờ lúc nào tài liệu nó cũng giống hệt như video mình hướng dẫn được, ok ?



#4.2. Chia layout và project structure

Đối với giao diện, chúng ta sẽ sử dụng lại Header và Footer, chỉ có nội dung ở body là thay đổi

=> sử dụng component Outlet của react router

#5. Cài đặt MongoDB Compass

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

Lưu ý: MongoDB Compass là phần mềm giúp kết nối tới database MongoDB.

Đây là công cụ kết nối tương tự như DBeaver (công cụ sử dụng trong khóa React Ultimate)

MongoDB Compass không phải là database

Trong khóa học, chúng ta sử dụng phần mềm này để có thể kết nối và xem data hiển thị được lưu trong database (nếu cần thiết).

#6. Tạo Tài Khoản MongoDB Atlas & Database

Link : <https://www.mongodb.com/cloud/atlas/register>

#7. Check kết nối tới MongoDB từ Localhost

todo...

#8. Setup dự án Backend

Download source code + file Collection Postman:

<https://drive.google.com/drive/folders/1vIDrrKi8PVkZ6UQR5qfa28Spvbqpo4WS?usp=sharing>

Lưu ý:

- Việc giải nén file .rar, hay ẩn hiện .env là cách dùng máy tính =))
- Update file .env với url database ở video trước

Đọc file Readme để biết cách làm, cài đặt và chạy dự án.

#9.1. Cấu trúc dự án Thực Hành

Dự án thực hành được chia thành các module riêng biệt, tương ứng với các chức năng website xây dựng.

Đối tượng sử dụng hệ thống (actor) gồm 2 loại chính: user (người dùng bình thường) và admin (người quản trị).

Như vậy, các chức năng chính được xây dựng là:

- Module Auth (authentication và authorization) :
 - + user login/register
 - + check quyền theo role (role user hay admin)
 - + access token và refresh token
- Module User (quản lý CRUD Users)
 - + import/export CSV
- Module Book (quản lý CRUD Books)
 - + Hiển thị danh sách books (client)
 - + Hiển thị chi tiết book khi xem chi tiết (client)
- Module Order (quản lý CRUD Orders)
 - + Hiển thị giỏ hàng (client)
 - + Chức năng đặt hàng, xem lịch sử đặt hàng.

Chapter 2: Module Auth

#9.2. Các Kiến Thức Sẽ Học

Lưu ý:

- Video này là video demo các kiến thức sẽ học trong chương, bạn chưa cần phải code theo video này
- Source code của chapter 2 (hết video #21):
https://drive.google.com/file/d/1Yr-8qDvoQEk4MnhRqmHkqpbCDecTilxT/view?usp=share_link

Bạn chỉ cần dùng source code khi cần, không cần tải bây giờ, vì các video tiếp theo (ngoài trừ phần CSS), mình đều hướng dẫn cách code.

Trong khóa học này, mình sử dụng access token và refresh token để xác thực người dùng (stateless).

Access token lưu ở Local Storage vì:

- Tiện cho việc truy cập và sử dụng. Chỉ cần dùng javascript là truy cập được, thông qua method `localStorage.getItem()`
- Thời gian sống của access token là ngắn, vì vậy, nếu có bị lộ token, nó cũng không nguy hiểm (do data lưu ở local storage tồn tại mãi mãi, nên có xác suất bị lộ token. Tuy nhiên xác suất này khá nhỏ)

Refresh token lưu ở cookies vì:

- Cookies sau 1 thời gian nhất định sẽ tự động hết hạn (tự động xóa dữ liệu lưu ở cookies)
- Cookies cho phép cấu hình để chặn truy cập cookies bằng javascript (thuộc tính `httpOnly`)
- Thời gian sống của Refresh token lâu hơn access token. Lưu ở cookies sẽ giảm thiểu rủi ro.

Về cơ chế access token/refresh token, bạn nào muốn học sâu, thì học thêm backend nhé. Mình để cập các kiến thức này ở đây :v

#10. Cài đặt thư viện UI Antd - Ant Design

Tài liệu: <https://ant.design/docs/react/introduce>

Cài đặt antd:

```
npm install --save-exact antd@5.3.2 @ant-design/icons@5.0.1
```

```
yarn add antd@5.3.2 @ant-design/icons@5.0.1
```

antd: thư viện UI viết sẵn React Component

ant-design/icons : thư viện icons

1. Tại sao dùng antd

- Giúp design nhanh giao diện responsive thông qua component, tương tự như react-bootstrap

<https://ant.design/components/overview/>

2. Controlled vs Uncontrolled component

controlled: dữ liệu của component được lưu vào state của React

uncontrolled: dữ liệu được lưu vào DOM (HTML)

Ví dụ về controlled component: form login sử dụng state để lưu giá trị input

Ví dụ uncontrolled component:

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

<https://formik.org/>

Với các thư viện trên, chỉ giúp validate form, không có hỗ trợ component khác.

antd hỗ trợ tất cả điều trên. amazing :v

3. Trade off

Khối lượng bundle của antd lớn, cần tối ưu khi sử dụng => thường dùng làm admin

v5.3.1 => unpacked size = 51.1 MB

bootstrap v5.2.3 = 9.1 MB

#11. Giới Thiệu Form Component

1. Các vấn đề tồn đọng

ví dụ về controlled component:

```
const LoginForm = () => {  
  
  const [email, setEmail] = useState("")  
  const [password, setPassword] = useState("")  
  
  return (  
    <input value={email} onChange={(e)=> setEmail(e.target.value)}/>  
    <input value={password} onChange={(e)=> setPassword(e.target.value)}/>  
  );  
  
}
```

- Nhược điểm của cách làm trên:

+ Chưa tối ưu hóa việc validate dữ liệu

+ Bị render nhiều lần (khi form có nhiều dữ liệu cần nhập) => dẫn tới giật/lag

=> đây là lý do react hook form/formik ra đời

Tuy nhiên, formik hay react hook form chỉ làm được form, phần còn lại của UI không làm được :v

2. Form component của Antd

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

#12. Bài tập Design Register Form

POST /api/v1/user/register

- Tạo mới trang /register để đăng ký tài khoản mới
- Giao diện tự thiết kế theo thẩm mỹ, sử dụng Ant Form
- Gồm 4 fields như api postman
- Validate dữ liệu không được để trống
- Khi submit form, lấy được hết dữ liệu nhập tại input

#13. Axios và Axios Customize

Tài liệu:

<https://www.npmjs.com/package/axios>

Axios Instance: <https://github.com/axios/axios#creating-an-instance>

Cài đặt:

```
npm install --save-exact axios@1.3.4
```

```
yarn add axios@1.3.4
```

1. Tại sao cần customize Axios ?

- Tập trung config lại 1 file
- Can thiệp vào request (trước khi gửi lên server backend)
- Can thiệp vào response (trước khi trả về cho client)

#14. Bài tập chức năng Register

Yêu cầu:

- gọi api register thông qua axios (đã customize)

API: `http://localhost:8080/api/v1/user/register`

- nếu có lỗi, thông báo lỗi

- tạo mới thành công, redirect về trang login

Lưu ý: tự design giao diện Login (gồm email/password dùng để đăng nhập)

#15. Bài tập Chức Năng Login

Yêu cầu:

- Tự design giao diện login
- Tạo input và validate dữ liệu (gồm 2 file username và password)
- Gọi API login

#16. Lưu Access Token và Refresh Token

- Lưu access_token vào Local Storage

- Lưu refresh_token vào cookies

<https://stackoverflow.com/a/72978783>

#17. Store User Login with Redux Toolkit

- store user info to redux

- gán access token vào axios:

<https://stackoverflow.com/a/51445755>

- default, f5 fresh redux => api fetch account

GET <http://localhost:8080/api/v1/auth/account>

lưu ý: với react 18, useEffect run twice (strict mode - không ảnh hưởng khi chạy production, chỉ có tại dev)

<https://github.com/facebook/react/issues/24502>

với việc sử dụng access token để xác định thông tin người dùng (và chỉ server mới decode được token này)

=> cơ chế stateless

mỗi lần f5 là react render lại từ đầu (bắt đầu từ component App, đồng thời khởi tạo redux với giá trị init cho state)

vậy làm sao để redux có dữ liệu, khi không thực hiện hành động nạp data (thông qua login) ?

=> cần gọi api để lấy lại thông tin user

#18.1 Protected Page

- create admin page (protected)

npm install --save-exact react-spinners@0.13.8

yarn add react-spinners@0.13.8

- children concept

- protected page

center screen: <https://github.com/davidhu2000/react-spinners/issues/53>

#18.2 Check quyền hạn của người dùng

Người dùng có Role = admin, mới có quyền vào '/admin'

#19.1 Bài Tập Design Layout

- Chia layout admin/client => design header bar/footer

- Các component của Antd đã sử dụng:

<https://ant.design/components/layout#components-layout-demo-side>

Liệt kê ra các component

lưu ý:

fix bug accountSlide user

#19.2 Fix bug Loading data

todo...

#20. Axios Retry - Handle Refresh Token

- Lưu ý: off react strict mode (only call once)

Tham khảo, không cần cài đặt package

<https://www.npmjs.com/package/axios-retry>

retry on 401:

<https://github.com/axios/axios/issues/934>

<https://stackoverflow.com/questions/73363862/axios-retry-infinite-loop>

#21. Fix Lỗi Hết Hạn Refresh Token

todo...

Chapter 3: Module Users

#22. Giới Thiệu Table Component

Tương tự: <https://www.npmjs.com/package/react-table>

Source code User Component:

https://drive.google.com/file/d/1LwnNqStyMBHvQ2_x8HBHUGiX5K_aPgyK/view?usp=share_link

#23. API Hiển Thị Danh Sách User

POST: <http://localhost:8080/api/v1/user?current=1&pageSize=2>

Lưu ý:

- Cần truyền Bearer Token ở header request
- current : trang hiện tại
- pageSize: số lượng bản ghi muốn lấy

#24. Bài Tập Hiển Thị Danh Sách User
todo...

#25. API Filter

sử dụng API:

GET <http://localhost:8080/api/v1/user?current=1&pageSize=2>

Quy tắc truyền filter: <https://www.npmjs.com/package/api-query-params>

1. Dạng basic

Truyền thêm các tham số cần filter:

ví dụ:

<http://localhost:8080/api/v1/user?current=1&pageSize=2&fullName=abc>

=> filter theo fullName = abc

<http://localhost:8080/api/v1/user?current=1&pageSize=2&fullName=abc&email=test@gmail.com>

=> filter theo fullName = abc và email = test@gmail.com

2. Filter với Regx

ví dụ:

<http://localhost:8080/api/v1/user?current=1&pageSize=2&fullName=/abc/i>

=> filter theo fullName có chứa từ 'abc'. tương tự như điều kiện %like% của SQL

#26. Bài Tập Filter Danh Sách User
todo...

#27. API Sorter

sử dụng API:

GET <http://localhost:8080/api/v1/user?current=1&pageSize=2>

Quy tắc truyền filter: <https://www.npmjs.com/package/api-query-params>

Mặc định, sort theo ascending/descending, tuy nhiên, ở nosql, có thể dùng dấu - (minus)

Ví dụ:

<http://localhost:8080/api/v1/user?current=1&pageSize=2&sort=fullName>

=> sort kết quả theo fullName (asc)

<http://localhost:8080/api/v1/user?current=1&pageSize=2&sort=-fullName>

=> sort kết quả theo fullName (desc)

#28. Bài tập Sorter
todo...

#29.1 Bài tập chức năng View Detail User

Tài liệu:

<https://ant.design/components/descriptions>

- Component Description (sử dụng để hiển thị data read-only)
- Component này tương tự table html, tuy nhiên đã css sẵn
- List props: <https://ant.design/components/descriptions#descriptions>

#29.2 Bài tập Add avatar

Tài liệu:

<https://ant.design/components/avatar>

url hiển thị: (avatar)

url-backend/images/avatar/image-id-here

Hướng dẫn add custom header => render table header with buttons

Table hiển thị thêm total

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

#30. Bài tập chức năng thêm mới user

Tài liệu:

<https://ant.design/components/modal>

sử dụng ref để submit

<https://ant.design/components/form#formuseform>

#31. Design modal upload

Giải pháp về upload file:

Cách 1.

User dùng browser(client) upload file lên website (file raw)

Client gửi file (raw) lên server

Server đọc file (raw) này, convert dữ liệu ra định dạng mong muốn (json),

Server xử lý và lưu dữ liệu vào database

Server trả về kết quả cho client

Cách 2.

User dùng browser(client) upload file lên website (file raw)

Client đọc file upload, convert dữ liệu ra định dạng mong muốn (json),

Client gửi dữ liệu json này lên server (không gửi file raw)

Server đọc dữ liệu json gửi lên, xử lý, lưu dữ liệu vào database

Server trả về kết quả cho Client

Tài liệu:

html input accept excel

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/file#accept>

<https://stackoverflow.com/questions/11832930/html-input-file-accept-attribute-file-type-csv>

antd upload without action

<https://stackoverflow.com/questions/51514757/action-function-is-required-with-antd-upload-control-but-i-dont-need-it>

- upload component (with drag/drop)

<https://ant.design/components/upload#components-upload-demo-drag>

Cần custom upload component:

- custom request (without upload)

- accept only excel file : CSV files (.csv), Excel Files 97-2003 (.xls), Excel Files 2007+ (.xlsx)

- single upload

#32. Read File Excel

Các bạn tự tạo file excel như trong video mình làm nhé 🙌

Tài liệu:

<https://www.npmjs.com/package/xlsx>

```
npm install --save-exact xlsx@0.18.5
```

```
yarn add xlsx@0.18.5
```

<https://docs.sheetjs.com/docs/solutions/input#example-user-submissions>

Upload file, sau đấy đọc dữ liệu, convert ra json

- input: file excel
- output: json object

#33. Bài tập Chức năng Import user

API bulk create:

POST <http://localhost:8080/api/v1/user/bulk-create>

truyền thêm password (mặc định)

#34. Download Sample File

<https://github.com/vitejs/vite/discussions/8271>

Lưu ý:

For those who just want a URL that point to the origin file,
append ?url at the end of the file path: import continentsCSV from
'../assets/SixContinentFirst.csv?url'.

Link docs: <https://vitejs.dev/guide/assets.html#explicit-url-imports>

#35. Export to CSV

tại sao csv, không phải excel ?

#36. Bài tập chức năng Export user

<https://stackoverflow.com/questions/70871254/how-can-i-export-a-json-object-to-excel-using-nextjs-react>

#37. Bài tập chức năng Update user

advance:

<https://ant.design/components/table#components-table-demo-edit-cell>

<https://ant.design/components/table#components-table-demo-edit-row>

#38. Bài tập chức năng Delete user

todo...

#39. Ôn tập các kiến thức đã học

Các component đã sử dụng:

- Table (với Paginate) : hiển thị danh sách người dùng
- Form, Modal : create/update user
- Drawer, Description: view detail user
- Popover : delete user
- Upload (with drag/drop): bulk create
- Avatar/Dropdown: header

Tập thói quen đọc document và đọc ngược lại source code thư viện (kết hợp với google)

Chapter 4: Module Books

#40. Bài tập Hiển Thị/Filter/Sort Books

todo: tương tự như chức năng users

API sử dụng:

GET <http://localhost:8080/api/v1/book?current=1&pageSize=10>

(truyền thêm filter và sort nếu cần thiết)

lưu ý: `const [sortQuery, setSortQuery] = useState("sort=-updatedAt");`
=> mặc định sẽ sắp xếp theo giá trị trường updatedAt (DESC)

#41. Bài tập design chức năng xem chi tiết Book

Source code mẫu của video này:

https://drive.google.com/file/d/18exFTpaeE19Ud2KB-V8c1S9F1L_Lpl/view?usp=share_link

- Sử dụng component upload để hiển thị ảnh

<https://ant.design/components/upload#components-upload-demo-picture-card>

- Để hiển thị ảnh => sử dụng base64

- customize để ko hiển thị icon remove => showUploadList

#42. Hoàn thiện chức năng xem chi tiết Book

Tài liệu: <https://www.npmjs.com/package/uuid>

install uuid

npm install --save-exact uuid@9.0.0

- remove base64

- hiển thị hình ảnh với url từ backend

- merge images thành list images

#43. Design Form Add New Book

Source code mẫu của video này:

https://drive.google.com/file/d/1CW4EnFTMonChAgLJyn1TIAA_iuuXD5Ux/view?usp=sharing

API get category: GET <http://localhost:8080/api/v1/database/category>

sử dụng component Select: <https://ant.design/components/select>

- form upload:

Có sử dụng validate (kiểu file và max size)

#44. Bài tập Upload file

Source code mẫu của video này:

https://drive.google.com/file/d/1zj_2fa0HwzcvOdwZ5ZXu_4hpyUu7EWhC/view?usp=share_link

Các việc cần làm:

- sau khi upload file thành công
=> lưu vào state của react (cần lưu vào state react để submit form + validate file)

- handle remove file
- handle preview images
- khi close modal => reset form

- API Upload Images

POST <http://localhost:8080/api/v1/file/upload>

trong đó:

Header truyền thêm 'upload-type' = book

data body truyền theo định dạng form-data:

fileImg: file ảnh raw upload

```
export const callUploadBookImg = (fileImg) => {  
  const bodyFormData = new FormData();  
  bodyFormData.append('fileImg', fileImg);  
  return axios({  
    method: 'post',  
    url: '/api/v1/file/upload',  
    data: bodyFormData,  
    headers: {  
      "Content-Type": "multipart/form-data",  
      "upload-type": "book"  
    },  
  });  
}
```

#45. Bài tập Create a new book
Todo..

Validate file before upload

#46. Bài tập Design form update book

Source code mẫu của video này:

https://drive.google.com/file/d/1ANH42qGOXla7DQwE0my8rg7_lx4S1SqA/view?usp=sharing

- ý tưởng: clone create book
- add initial data
- build file list (control with state)
- lưu ý về clear state (unmount form)

#47. Bài tập Update a book

- tạo hidden fields (lưu ý về layout)
- truyền thêm _id cho api

API :

method là PUT, cần truyền thêm id cho url:

http://localhost:8080/api/v1/book/id_book_here

#48. Bài tập Delete a book

- done
- export a book :v

#49. Tổng Kết Các Kiến Thức Đã Học

...

lưu ý về warning của antd dưới browser

warning vs error

#50. Bài tập Design giao diện client (homepage)

Source mẫu của video này:

https://drive.google.com/file/d/1NX7UE0xmPDFsbj4zXObn41FilzRHypbr/view?usp=share_link

- category (filter)
- list book
- lưu ý: có paginate

các bước cần làm:

- design base giao diện (hardcode dữ liệu)
- ghép api
- test form

- lưu ý về group checkbox
- group form item (2 names)

Các component sử dụng:

<https://ant.design/components/checkbox#components-checkbox-demo-group>

<https://ant.design/components/rate>

<https://ant.design/components/tabs>

<https://ant.design/components/pagination>

#51. Hiển thị HomePage

- api category

- api get list book (with paginate)

- thêm component loading:

<https://ant.design/components/spin#components-spin-demo-tip>

#52. Bài tập Filter/Sorter

update tabPane -> order by query

- update tabPane -> order by query

- thêm component loading:

<https://ant.design/components/spin#components-spin-demo-tip>

- làm reset filter

<https://www.npmjs.com/package/api-query-params>

Toán tử \$in, \$lte, \$gte

#53. Url với Params và Query

Source code mẫu của video này:

<https://drive.google.com/file/d/1jLCOsiADG-xlxemuU7BRZT1A29M8C20I/view?usp=sharing>

Về router với params: <https://reactrouter.com/en/main/start/concepts#match-params>

- Phân biệt Params và Query string
- khái niệm slug

convert slug

<https://stackoverflow.com/a/5782563>

<https://reactrouter.com/en/main/start/concepts#locations>

update điều kiện redirect login khi f5 page view detail

===

output: get book id

#54. Bài tập Design View Detail Page

- đưa ra yêu cầu

<https://www.npmjs.com/package/react-image-gallery>

```
npm i --save-exact react-image-gallery@1.2.11
```

```
yarn add react-image-gallery@1.2.11
```

Lưu ý về import scss:

<https://github.com/vitejs/vite/issues/5764>

<https://github.com/webpack-contrib/sass-loader#resolving-import-at-rules>

#55. Chức Năng View Detail Book

Source code mẫu của video này:

https://drive.google.com/file/d/12DFjqvcif-h9ptipbUa6u5Xgwn3AvN6g/view?usp=share_link

- giải thích source code

<https://ant.design/components/image>

#56. Thêm Loading Skeleton

Source code mẫu của video này:

https://drive.google.com/file/d/1A6ja44sXOA2v2FH2gykcHzUkB_OcUzhK/view?usp=sharing

1. Sử dụng Skeleton

có nhiều thư viện hỗ trợ tạo skeleton:

<https://www.npmjs.com/package/react-content-loader>

<https://github.com/danilowoz/create-content-loader/blob/master/src/Gallery/insertYourLoaderHere/ProductDetails.js>

<https://www.npmjs.com/package/react-loading-skeleton>

hoặc sử dụng antd: <https://ant.design/components/skeleton>

#57. Hoàn thiện View detail page
- ghép api

Lưu ý về tên class left/right

<https://github.com/xiaolin/react-image-gallery/issues/651>

#58. Quá trình xây dựng giao diện

- Design base (html-css)
- Write react:
 - + (chia component nếu cần thiết)
 - + handle logic (xử lý các function và thao tác của người dùng)

Chapter 5: Module Orders

#59. Setup Redux Persist

Source code mẫu của video này:

<https://drive.google.com/file/d/1SMYFI7RgPe3KaXZcEHNrowJqQt5clrcu/view?usp=sharing>

- Thông thường, với redux, f5 là refresh data => cần nạp data lại từ đầu

redux persist là giải pháp lưu data vào localStorage, sau đấy, nạp ngược lại vào redux

Tích hợp redux persist

npm install --save-exact redux-persist@6.0.0

yarn add redux-persist@6.0.0

Tài liệu: <https://redux-toolkit.js.org/usage/usage-guide#use-with-redux-persist>

<https://www.npmjs.com/package/redux-persist>

#60. Bài tập Add/Remove Product

1. Xử lý tăng giảm button và input

Nhấn trừ => số lượng giảm 1, và giảm tới 1, ko là số âm

Nhấn cộng => số lượng tăng 1, và tăng không quá max quantity

Nhập input thì chỉ cho phép nhập số, và ko tăng quá max quantity

2. Lưu data vào redux

..todo

#61. Bài tập hiển thị số lượng giỏ hàng

- cập nhật số lượng trên header

- thông báo thêm thành công

#62. Design Preview Cart

Sử dụng component Dropdown:

<https://ant.design/components/dropdown#components-dropdown-demo-arrow-center>

group title:

<https://ant.design/components/dropdown#components-dropdown-demo-sub-menu>

hoặc sử dụng popover: <https://ant.design/components/popover>

<https://ant.design/components/popover#components-popover-demo-arrow>

Gồm 2 bước:

- build hardcode
- build dynamic items

1. Lưu ý về css

Modal, Popover... không render trong <div id="root">

<https://react.dev/reference/react-dom/createPortal>

Ở đây là css global => lưu ý về đặt tên class

note 1: css-dev-only class

<https://github.com/ant-design/ant-design/discussions/38753>

Note 2: ko cần check trên mobile vì không có sự kiện hover => bắt buộc phải click để xem chi tiết

2. Hoàn thiện với data từ redux

//todo

#63. Design base order page

Source code tham khảo:

https://drive.google.com/file/d/1v3Co_BFckrDT0yi8buJYnoIXN3u6pQ0N/view?usp=sharing

#64. Bài tập order page

- render động list carts
- tăng giảm số lượng sản phẩm
- xóa sản phẩm
- tính tổng số tiền

#65. Bài tập Payment Page

Các component sử dụng:

- Hiển thị steps:

<https://ant.design/components/steps>

- Hiển thị empty:

<https://ant.design/components/empty>

- Hiển thị kết quả:

<https://ant.design/components/result>

Hiển thị textarea:

<https://ant.design/components/input#components-input-demo-textarea>

Yêu cầu:

Quá trình đặt hàng gồm 3 bước:

Bước 1: Thông tin đơn hàng

- Cho phép tăng/giảm số lượng sản phẩm
- Nếu không có sản phẩm trong giỏ hàng, hiển thị thông báo Empty
- Nhấn button 'mua hàng' để chuyển qua bước 2. Nếu không có sản phẩm trong giỏ hàng thì chặn không cho qua bước 2

Bước 2: Thông tin đặt hàng

- Không cho phép tăng/giảm số lượng sản phẩm
- Yêu cầu điền thông tin đặt hàng, bao gồm: tên, số điện thoại, địa chỉ
- Nhấn nút "Đặt hàng" để gọi API
- Nếu đặt hàng thành công, chuyển qua bước 3. Có lỗi thì hiển thị thông báo lỗi

Bước 3: Hiển thị kết quả

- Cho phép xem lịch sử đơn hàng

#66. API Create An Order
todo...

#67. Bài tập View Order History

API:

GET <http://localhost:8080/api/v1/history>

<https://www.npmjs.com/package/react-json-view>

<https://github.com/mac-s-g/react-json-view/issues/441>

`npm install react-json-view --legacy-peer-deps`

Giải thích về `--legacy-peer-deps`:

<https://stackoverflow.com/questions/66239691/what-does-npm-install-legacy-peer-deps-do-exactly-when-is-it-recommended-wh>

#68. Bài tập Update Current User

1. Update User Info

- Có thể update được avatar và thông tin của người dùng

Lưu ý:

Để cập nhật Avatar, cần gọi api để upload file ảnh, tương tự như upload image book

POST <http://localhost:8080/api/v1/file/upload>

Sử dụng form-data, với header: upload-type === "avatar"

Để cập nhật được thông tin của người dùng, cần gọi API update:

PUT <http://localhost:8080/api/v1/user>

(cần truyền lên avatar, kết quả của việc upload ở trên)

Để cập nhật được thông tin user, sau khi update thành công, cần cập nhật thông tin lưu trong redux, bao gồm fullName, avatar và phone

Đồng thời, "ép tạo token mới", vì token cũ đang lưu thông tin chưa update (đơn giản là xóa access_token là được)

2. Update password

POST <http://localhost:8080/api/v1/user/change-password>

Cần truyền lên: email, oldpass, newpass

#69. Bài tập Display Order page + API Dashboard

npm i react-countup

<https://ant.design/components/statistic#components-statistic-demo-animated>

<https://ant.design/components/card#components-card-demo-border-less>

<https://ant.design/components/breadcrumb#components-breadcrumb-demo-withicon>

#70. Các bugs còn tồn đọng

1. Giao diện mobile

- Tự CSS giao diện
- Update thêm filter cho mobile

2. Protected Route

Update protected route (user/admin)

3. Thanh search bar

<https://reactrouter.com/en/main/hooks/use-outlet-context>

Tương tự như search theo tên sách

4. Lỗi khi F5 ở Admin

Với admin, f5 lại trang thì chưa đúng active menu

<https://stackoverflow.com/questions/41054657/react-routerantd-how-to-highlight-a-menu-item-when-press-back-forward-button>

Chapter 6: Triển Khai Dự Án Thực Tế

#71. Mô hình triển khai

Frontend => Backend => Database

1. Frontend React: triển khai miễn phí với Vercel
2. Backend Node.JS: triển khai miễn phí với Render
3. Database MongoDB: triển khai miễn phí với MongoDB Atlas (đã làm ở #8, khi setup dự án backend)

Do database đã setup từ đầu => cần setup thêm frontend/backend

#72. Chuẩn Bị Môi Trường Backend

1. Triển khai ứng dụng Backend miễn phí với Render.com

- Render cho phép triển khai ứng dụng Backend (node.js) miễn phí.
(Trước 12/2022, chúng ta có Heroku)

- Gói miễn phí phục vụ mục đích testing (học tập). làm thực tế (chạy thật), sẽ không dùng Render.com

Thông tin về gói miễn phí:

<https://render.com/docs/free>

Lưu ý: Sau 15 phút không hoạt động -> hibernate

2. Tạo tài khoản Render

Yêu cầu:

- cần có 1 tài khoản github

- Đẩy source code backend lên github

- Tạo tài khoản Render

#73. Triển Khai Backend với Render

Chuẩn bị:

- Source code đã được đẩy lên github
- tài khoản Render
- đường link URL của MongoDB (update trong file .env).
- Lưu ý về file .env (nếu có)

=====

- Cách đọc logs ứng dụng ???

Sau khi triển khai, cần test postman (hoặc test frontend tại localhost, với Backend đã hosting)

#74. Chuẩn bị Môi trường Frontend

- Source code Frontend đẩy lên github

- Tạo tài khoản Vercel

Thông tin về Vercel: <https://vercel.com/pricing>

#75. Triển khai Frontend với Vercel

- Lưu ý về file .env

- Không dùng version Node.js mới nhất

- Lưu ý: với Vercel, khi F5 (refresh) thì bị lỗi. Để khắc phục, tạo file vercel.json tại thư mục root (tương tự như file .env), với nội dung:

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

=> sau đây push code là Vercel tự động build lại project => problem solve :v

Link tham khảo: <https://stackoverflow.com/a/65644138>

#76. Tại sao dùng Host Miễn Phí Không Lưu Cookies

<https://devcenter.heroku.com/articles/cookies-and-herokuapp-com>

#77. Mua tên miền

<https://vercel.com/guides/a-record-and-caa-with-vercel>

<https://render.com/docs/configure-other-dns>

#78. Chạy app với tên Miền

//todo

Chapter 7: Góc Nhìn Về Level Fresher

#79. Chuyện Học 1 Khóa Học

//todo

#80. Level Fresher là level như thế nào

//todo

#81. Học gì tiếp đây ?

//todo

Chapter 8: Fix Bugs

Fix các bugs học viên “phát hiện” trong quá trình thực hiện khóa học, giúp khóa học trở nên hoàn thiện (hoàn hảo) hơn 🙌

#82. Fix Bug Filter

//todo

#83. Fix Bug Refresh Token

Chuẩn bị:

- On Strict Mode của React
- Setup Access Token với thời gian ngắn, ví dụ 30s
- Off logic redirect to '/login'

Cài đặt thư viện:

npm i --save-exact async-mutex@0.4.0 --legacy-peer-deps

<https://www.npmjs.com/package/async-mutex>

file config axios:

```
import axios from "axios";
import { Mutex } from "async-mutex";

const mutex = new Mutex();

const baseUrl = import.meta.env.VITE_BACKEND_URL;

const instance = axios.create({
  baseUrl: baseUrl,
  withCredentials: true,
});

instance.defaults.headers.common = { 'Authorization': `Bearer ${localStorage.getItem('access_token')}` }

const handleRefreshToken = async () => {
  // const res = await instance.get('/api/v1/auth/refresh');
  // if (res && res.data) return res.data.access_token;
  // else null;

  return await mutex.runExclusive(async () => {
    const res = await instance.get('/api/v1/auth/refresh');
    if (res && res.data) return res.data.access_token;
    else return null;
  });
}

// 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);
});
```

```
const NO_RETRY_HEADER = 'x-no-retry'

// Add a response interceptor
instance.interceptors.response.use(function (response) {
  // Any status code that lie within the range of 2xx cause this function to trigger
  // Do something with response data
  return response && response.data ? response.data : response;
}, async function (error) {
  // Any status codes that falls outside the range of 2xx cause this function to trigger
  // Do something with response error
  if (error.config && error.response
    && +error.response.status === 401
    && !error.config.headers[NO_RETRY_HEADER]
  ) {
    const access_token = await handleRefreshToken();
    error.config.headers[NO_RETRY_HEADER] = 'true'
    if (access_token) {
      error.config.headers['Authorization'] = `Bearer ${access_token}`;
      localStorage.setItem('access_token', access_token)
      return instance.request(error.config);
    }
  }

  if (
    error.config && error.response
    && +error.response.status === 400
    && error.config.url === '/api/v1/auth/refresh'
  ) {
    if (
      window.location.pathname !== '/'
      && !window.location.pathname.startsWith('/book')
    ) {
      window.location.href = '/login';
    }
  }

  return error?.response?.data ?? Promise.reject(error);
});

export default instance;
```

#84. Về Bug Warning của Antd

//todo