TÀI LIỆU KHÓA HỌC “React Native Ultimate”

Tác giả: Hỏi Dân IT & Eric

Version: 1.0

Note cập nhật:

- Update tài liệu

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 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

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. Demo kết quả đạt được

Mục tiêu của khóa học: chỉ cần bạn biết code React làm website, làm mobile với React như nào, đã có khóa học lo (hướng dẫn từ số 0 khi làm mobile)

1. Công nghệ sử dụng
React Native version 0.74
Expo SDK version 51.0.0
(cuối khóa học có hướng dẫn nâng cấp lên SDK version mới nhất)

Các kiến thức trọng tâm:

- Hướng Dẫn React Native từ số 0, Expo Framework từ số 0

- Sử dụng Expo Framework kết hợp với Expo Router (thay vì React Native CLI và React Navigation)

- Chú trọng việc design UI/UX dành cho Frontend Developer, cũng như tái sử dụng component (reuse) với React native

- Sử dụng uncontrolled component với Formik để tối ưu hiệu năng
- Sử dụng React Context để chia sẻ data giữa các component (đơn giản nhất)

- Dự án thực hành : Food App

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

Frontend Admin (React-Antd) được mình cung cấp. Chỉ sử dụng và không sửa đổi (**không học code React làm web trong khóa học này**). Quan tâm về frontend React làm website, tham khảo tại đây

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

2. Học viên nào có thể học ?

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

- Có kiến thức về React làm website: state/props, calling APIs backend
- Cú pháp của TypeScript
- Máy tính bạn có thể cài đặt và mở ứng dụng Android Studio

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

#4. Yêu cầu để học được khóa học này

1. Biết React.js làm website

Các kiến thức về React làm website rất quan trọng và bổ trợ đắc lực cho quá trình học React Native, bởi vì:

React Native = kiến thức React + kiến thức về Native platform (Android/iOS)

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 tại đây

2. Biết cú pháp của Typescript

Việc sử dụng React nói chung (React Native nói riêng) 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

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

#5.1 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 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 project final, 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 #80)

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

#5.2 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 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

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

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/pfbid02gyasktd3semgxat6nevnvwh4c8epzu3i7kpzhr7s7gmmfcvucyz96eb8avnvgwnhl
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.
   NẾU 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.
   NẾU 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. **NẾU 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

Đấy 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-5fgi

#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
- Expo Tools

#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

#14. Cách Test Ứng Dụng React Native

Do chúng ta code mobile, vì vậy, sẽ cần test trên thiết bị di động.
Gồm 2 loại thiết bị: **máy ảo** (simulator) và **máy thật** (chính là smartphone của bạn)

Có 2 khả năng xảy ra:

Khả năng 1: máy tính bạn thực hành coding có cấu hình đủ mạnh

Yêu cầu tối thiểu: RAM min = 8GB và CPU cần đủ khỏe
Chi tiết: https://developer.android.com/studio/install

Nếu máy tính của bạn cài Android Studio, chạy và sử dụng không bị giật/lag, chúng ta sẽ dùng máy ảo (simulator) để test ứng dụng mobile.

(Video hướng dẫn cài đặt Android Studio là video tiếp theo)

Ngược lại, máy bạn cấu hình yếu, hoặc chạy bị giật lag, chuyển qua khả năng 2

Khả năng 2: máy tính của bạn cấu hình yếu

Với điện thoại Android, tải ứng dụng Expo

Với điện thoại iOS, tải dụng Expo Go

Lưu ý về mobile:

- **Khóa học này mình ưu tiên test trên điện thoại Android** (thông qua máy ảo). Lý do bởi vì mình đang dùng máy tính windows, không thể (impossible) chạy máy ảo iOS

- Nếu bạn nào muốn test iOS thì : một là dùng chính iphone của bạn để test, hoặc sử dụng Macbook, cài đặt phần mềm XCode
  Bạn không thể (hoặc không nên) test iOS trên Windows (Apple Rules)

- Nếu bạn sở hữu Macbook, bạn vẫn cài đặt và sử Android Studio như bình thường. Bạn xem và thao tác giống hệt trong video.

#15. Cài Đặt Android Studio

Mục đích cài đặt Android Studio, là bạn có thể chạy máy ảo android (simulator)

Nếu máy tính của bạn sau khi cài đặt Android Studio, gặp tình trạng giật/lag, bạn không cần phải sử dụng công cụ này trong quá trình học.

Thay vì vậy, bạn cài đặt ứng dụng Expo để test trên chính smartphone của bạn (vẫn theo dõi và thực hành khóa học như bình thường)

Bước 1: Tải version mới nhất tại đây
https://developer.android.com/studio

Bước 2: Cài đặt

Lưu ý: quá trình cài đặt diễn ra từ 3 tới 5 phút (vì vậy hãy kiên nhẫn cài đặt)

#16. Cài Đặt Máy Ảo (Simulator)

- Hướng dẫn cách thêm máy ảo

- Hướng dẫn cách restart/tắt app khi cần thiết (sẽ được mình hướng dẫn chi tiết tại video #16.1)

#16.1 Sử dụng máy ảo Android & trường hợp bị giật/lag/đơ máy ảo ?

Các tính năng cần biết:

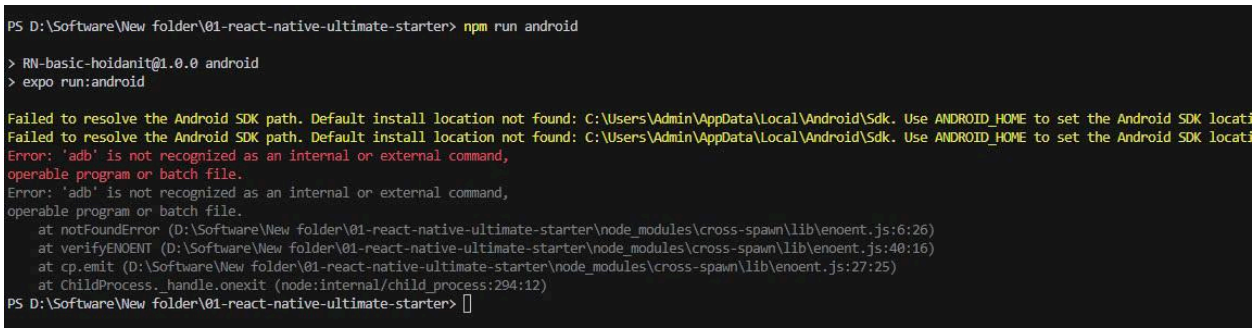
- Tắt/Mở máy ảo/thiết bị test
- Nút back, home, điều hướng của android
- Nút capture screen màn hình (Ctrl + S)
- Cách xử lý trường hợp máy ảo bị giật/lag/đơ máy ?
- Cách đổi ngôn ngữ bàn phím/theme dark,light (Nút Settings)

#16.2 Cấu hình Java và Android Path

Tham khảo: (sử dụng VPN để xem)
https://medium.com/@prasadkatkade008/how-to-set-up-expo-dev-client-a-complete-guide-2024-be898a519ec1

Nếu bạn không cấu hình, sẽ gặp lỗi như sau:
Failed to resolve the Android SDK path. Use ANDROID_HOME to set the Android SDK location.

Error: 'adb' is not recognized as an internal or external command,
operable program or batch file.



Bước 1: Cài đặt môi trường Java
Lưu ý cài đặt tối thiểu Java version 17 trở lên.
Trong khóa học, mình sử dụng java version 17.

Download java tại đây:
https://www.oracle.com/in/java/technologies/downloads/archive/

Sau khi cài đặt xong, cần cấu hình **JAVA_HOME**

Với MAC OS, tham khảo tại đây:
https://www.youtube.com/watch?v=8RPC1ncNBUc
https://www.baeldung.com/java-home-on-windows-mac-os-x-linux

Sau khi cấu hình xong, kiểm tra bằng cách gõ câu lệnh: java

Bước 2: Cài đặt Android Studio
//đã làm tại video #15

Bước 3: Cấu hình tham số môi trường
Cấu hình ANDROID_HOME

Với máy tính Windows:

ANDROID_HOME = C:\Users\Username\AppData\Local\Android\Sdk

//update Path
- C:\Users\Username\AppData\Local\Android\Sdk\platform-tools
- C:\Users\Username\AppData\Local\Android\Sdk\emulator

Với Macos, tham khảo tai đây (dùng vpn để xem)

Sau khi cấu hình xong, kiểm tra bằng cách gõ câu lệnh: adb

Chapter 3: Tổng Quan về React Native
Làm quen với quá trình phát triển của framework React Native. Viết chương trình Hello World đầu tiên.

#17. React Native là gì ?

1.Phân biệt các thuật ngữ thường gặp: React, React js, React Native

React là một công nghệ do Facebook (Meta) phát triển từ 2013.

Khi nói tới React, chúng ta thường hiểu là React.js, một thư viện (library) để phát triển website
https://react.dev/

React Native là một framework sử dụng React để tạo nên ứng dụng chạy trên Android và iOS

2.Lịch sử phát triển của React Native
https://endoflife.date/react-native

Về tài liệu của React Native, đọc tại đây:
https://reactnative.dev/

#18. Base dự án với React Native

Giới thiệu về expo, react native cli

Về cách setup 1 dự án React Native, tham khảo:
https://reactnative.dev/docs/environment-setup

Có 2 cách chính:

- Sử dụng framework Expo: https://expo.dev/

- Sử dụng React Native CLI:

  https://reactnative.dev/docs/getting-started-without-a-framework

  https://github.com/react-native-community/cli

Minh recommend (khuyến khích) sử dụng Expo thay vì sử dụng React Native CLI

Điều này tương tự sử dụng Next.js, thay vì code React.js thuần túy

Lý do:

- Phần lớn các tính năng hay dùng (các vấn đề hay gặp), framework đã giải quyết. Bạn không nên đi sáng tạo lại cái bánh xe (re-invented the wheel)

- Trang chủ của React Native cũng khuyến khích làm vậy.

- Bạn chỉ dùng React Native CLI khi bạn muốn kiểm soát 100%, hoặc sử dụng Expo không làm thỏa mãn bạn (không giải quyết tốt vấn đề của bạn)

Tài liệu về React Native, bạn đọc tại:
https://reactnative.dev/

https://expo.dev/

#19. Các Thuật Ngữ Hay Gặp
Lưu ý:
(không nên dịch nghĩa anh-việt, nên cảm nhận ý nghĩa của nó)

Web app: hiểu đơn giản là website, chạy với browser

Cross-platform: chạy đa nền tảng
Ví dụ: ứng dụng Facebook Messenger chạy trên web, desktop app, mobile app (android/ios). Thậm chí hỗ trợ hệ điều hành windows/macOS/Linux

Hybrid app: (con lai) 1 source code, chạy đa nền tảng, từ web cho tới mobile
Đại diện tiêu biểu là ứng React Native và Flutter

Native App (tương tự native speaker): ứng dụng được xây dựng với "ngôn ngữ gốc" (native)
Ví dụ: ứng dụng chạy trên android, được viết = kotlin
Ứng dụng chạy trên iOS, được viết = swift

=> native app là các ứng dụng được viết bởi ngôn ngữ gốc (kotlin/swift)

React Native là framework, giúp bạn xây dựng ứng dụng chạy trên mobile (android/ios) và web từ duy nhất 1 source code base
=> bạn có thể thể gọi React Native là Cross-platform app / hybrid app

So sánh với flutter
Tương tự việc bạn so sánh quả cam và quả táo (apple vs orange). Nó không mang ý nghĩa gì cả

Điều quan trọng, là bạn học 1 công cụ, và công cụ đấy giải quyết hiệu quả vấn đề bạn gặp phải => done

#20. Tại sao Công Ty Dùng React Native

1.Tại sao công ty sử dụng React Native, thay vì Native app ?
Trước khi quyết định lựa chọn công nghệ cho dự án, điều cần làm là :

- Phân tích nghiệp vụ (yêu cầu - requirement) của dự án

- Lựa chọn các công nghệ "có thể" dùng: Kotlin (android), Swift (iOS), Flutter (Dart), React Native (Javascript)

- **Cân nhắc nguồn lực của công ty**: hiện tại có thể làm được ngôn ngữ gì, có cần tuyển thêm người hay không ? Nếu tuyển, thi tuyển level nào (SENIOR, FRESHER)

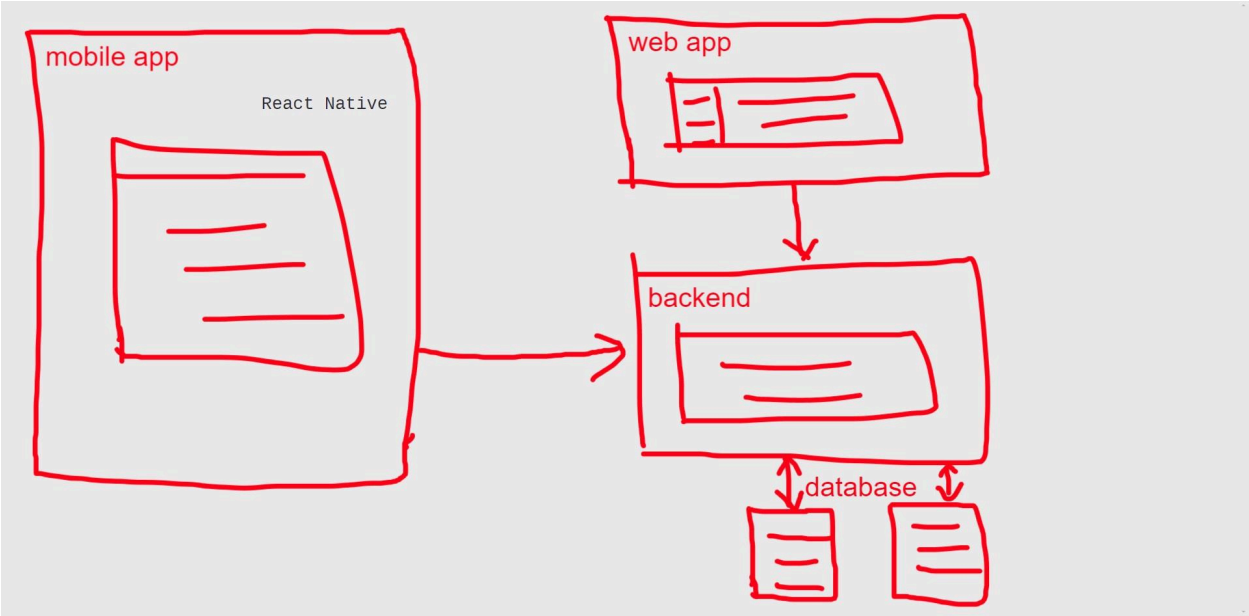
- **Cân nhắc chi phí của dự án**: nếu chọn native app, sẽ cần 2 team (1 cho android và 1 cho iOS). Nếu chọn cross-platform app thì cần 1 (flutter hoặc react native)

- **Cân nhắc tính tối ưu** (performance): muốn ứng dụng cần phải có hiệu năng tối đa (ví dụ bạn chơi game chẳng han), hay ứng dụng mang tính "tương đối" là được ?
  Nếu performance là ưu tiên số 1 => chọn native app
  Ngược lại, chọn hybrid app với flutter/react native

Đối với 1 công ty, việc lựa chọn công nghệ sẽ cần cân bằng các yếu tố ở trên.
Làm gì thì làm, miễn sao phải đảm bảo lợi nhật là tối đa.

2. Mô hình dự án của công ty (basic nhất)

- Vẽ ra mô hình để hình dung (các thành phần của react native)



#21. Đặt tên file JS/JSX/TS/TSX cho React (Extra)

Recommend Typescript

Cách đặt tên file jsx, js , ts, tsx

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)

Extra: 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?

#22. 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 1: nếu bạn dùng macOS, chạy bị lỗi, tham khảo mục 3 bên dưới

Lưu ý 2: 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-native => thay bằng: D:/du-an/react-native

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)

Lưu ý về warning của expo (nếu có)

Trường hợp 1: nếu máy tính của bạn cấu hình yếu (không chạy được Android Studio), dùng smartphone của bạn, quét mã QR tại terminal

Trường hợp 2: máy tính của bạn có khả năng chạy được Android Studio

Chạy ứng dụng thông qua máy ảo (simulator)

3. Fix lỗi trên MacOS (nếu có)

Lưu ý: dùng version nodejs yêu cầu như trong khóa học để hạn chế tối đa lỗi

Lỗi 1: **EMFILE: too many open files, watch ...**

Bước 1: cài đặt watchman
brew update
brew install watchman

Bước 2: chạy lại dự án

Nếu bước 2 failed, chuyển bước 3

Bước 3:
Xóa file **package_lock.json**, xóa thư mục **node_modules**

Cài lại thư viện với câu lệnh: **npm i**

Chạy lại dự án, nếu failed, chuyển bước 4

Bước 4:
Xóa project hiện tại, restart lại máy tính.

Clone lại dự án thực hành, cài đặt thư viện (npm i) rồi chạy dự án

Nếu bước 4 vẫn failed, tham khảo:
https://stackoverflow.com/questions/76129467/expo-giving-an-error-emfile-too-many-open-files-for-my-react-native-and-fi

https://github.com/expo/expo/issues/29083

#23. 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 #22.

Nếu như bạn tự ý code “theo cách bạn hiểu” và không sử dụng source code tại #22, mình sẽ không hỗ trợ support khi có lỗi xảy ra

Tài liệu: https://docs.expo.dev/get-started/create-a-project/

#24. Hello World với React Native

1. Hướng dẫn các câu lệnh hay dùng

Cách refresh app (giới thiệu về hot reloading)

Cách force - kill app

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/

#25. Cấu trúc Dự Án Thực Hành

node_modules : lưu trữ các thư viện sử dụng cho dự án

assets: lưu trữ hình ảnh/font (static files)

.gitignore: các file không dùng git để quản lý

App.tsx : file main dùng để chạy dự án

README.MD : file cung cấp thông tin về dự án

app.json: cấu hình project của expo (được dùng để build app sau này)

babel.config.js : cấu hình babel, giúp dịch/chuyển đổi code javascript

package.json : các thư viện cài đặt và các câu lệnh để chạy dự án

package-lock.json : chi tiết các thư viện cài đặt

tsconfig.json : cấu hình typescript sử dụng cho dự án

Chapter 4: Sử Dụng UI - User Interface

Cách trang trí cho ứng dụng React Native với Styles và sử dụng Core Component

#26. Tổng quan về chapter

Các kiến thức trọng tâm của chương học:

- Phân biết Core Component và Native Component, hiểu được cách React Native hoạt động
- Sử dụng Core Component của React Native : View, Text…
- Trang trí cho component với Styles của React Native
- Nắm vững Flexbox để có thể Styles cho Component

#27. Phân biệt Core Component và Native Component

Tài liệu: https://reactnative.dev/docs/intro-react-native-components

Native Component: các component thuộc về Android và iOS (native platform)

Core Component: các component cốt lõi của React Native (quan trọng nhất).

Tất cả các component còn lại, đều được tạo nên từ những Core Component này.

Ví dụ:
https://github.com/facebook/react-native/blob/main/packages/react-native/Libraries/Components/Pressable/Pressable.js#L346

#28. View, Text & Styles

1.Nguyên tắc cần tuân theo

View (ánh xạ sang web là **div**), bạn dùng để bọc layout (tạo base layout)

Text (ánh xạ sang web là **p**), chỉ dùng để hiển thị text.
Không viết text đơn lẻ, cần sử dụng Text Component => báo lỗi nếu không tuân theo

Đối với React Native, không có khái niệm CSS, chẳng qua framework cung cấp cho chúng ta một công cụ, nó có nhiều điểm tương đồng với cách code CSS

Chúng ta dùng **Styles** để có thể trang trí cho component

2.Về các trang trí (Styles) cho component

Có 2 cách:

- **Dùng inline với style props**: lưu ý có nhiều thuộc tính không giống như CSS
- Dùng styled component (css-in-js)

Với react native, child component không kế thừa lại thuộc tính của cha

(ví dụ color) => cũng không bị ghi đè css của cha và con (ví dụ như khái niệm important)

Ngoại lệ (text lồng trong text có thể kế thừa styles)

3. 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. Sử dụng State và Data Type

Các loại data (thường dùng) được sử dụng với React:

- **String** : chuỗi ký tự
- **Number** : số nguyên, số thực…
- Boolean, null, undefined
- Array/Object

Tip: để check giá trị của Array/Object, cần convert sang định dạng String

Về hàm **JSON.stringify**, tham khảo tại đây

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. TextInput

Tài liệu: https://reactnative.dev/docs/textinput

Một vài props hay dùng:

keyboardType

multiline

maxLength

onChangeText

autoCapitalize="none"

autoCorrect={false}

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

#31. Button & Array

1.Giới thiệu về component Button
https://reactnative.dev/docs/button

Lưu ý: với button mặc định, không thể CSS cho button

Muốn CSS cho button, cần tạo custom button (sẽ hướng dẫn sau).
Ý tưởng về button, tham khảo tại đây

2. Render dữ liệu với map
Về vòng lặp map, tham khảo tại đây

```
const [todoList, setTodoList] = useState([
   { id: 1, title: "Learn React Native" },
   { id: 2, title: "Learn React.js" },
   { id: 3, title: "Watching Netflix" },
   { id: 4, title: "Playing ESport" },
   { id: 5, title: "Subscribe Hỏi Dân IT :v" },
   { id: 6, title: "Watching Youtube" },
   { id: 7, title: "CR 7" },
   { id: 8, title: "Tony Kroos" },
   { id: 9, title: "Nine" },
   { id: 10, title: "M10" },
 ])
```

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

#32. ScrollView

Tài liệu:
https://reactnative.dev/docs/scrollview

Render array với scroll

Each child should have a unique key

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

#33. FlatList

Tài liệu: https://reactnative.dev/docs/flatlist

Sử dụng FlatList thay vì ScrollView, sẽ tốt cho performance dự án của bạn

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. Todo App (Part 1)

//chia tách component
//sử dụng component của React với typescript

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. Todo App (Part 2)

//thêm động todo (add todo)

//truyền function từ cha sang con (với props)

//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

#36. Todo App (Part 3)

//component **TouchableOpacity**
https://reactnative.dev/docs/touchableopacity

//xóa 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

#37. Todo App (Part 4)

//component **Alert**
https://reactnative.dev/docs/alert

//click outside, đóng keyboard
https://reactnative.dev/docs/touchablewithoutfeedback

https://reactnative.dev/docs/keyboard

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. Custom Component (Button)

//sử dụng icons với Expo:
https://icons.expo.fyi/Index

https://docs.expo.dev/guides/icons/

//pressable
https://reactnative.dev/docs/pressable

Css with opacity:
https://stackoverflow.com/a/76192331

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. Sử dụng Flexbox (Part 1)

Tài liệu: https://reactnative.dev/docs/flexbox
https://www.w3schools.com/css/css3_flexbox.asp

Các thuộc tính hay dùng của Flexbox với React Native:
flexDirection, alignItems, and justifyContent

Các lưu ý khi sử dụng với React Native:
flexDirection: mặc định là column, không phải là row
flex parameter only support a single number.

Các nội dung cần học:

- Chia tỉ lệ với **flex** : 1, 2, 3..
- Chiều hướng với **flexDirection**

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. Sử dụng Flexbox (Part 2)

- **Justify Content** vs **Align Items** (cần lưu ý về trục ox và oy)

  Nếu container cha, **direction là row, justify sẽ căn theo trục ox**

  Align item sẽ căn theo trục oy

  Nếu container cha, **direction là column, justify sẽ căn theo trục oy**

  Align item sẽ căn theo trục ox

- Inline-block: **Align Self**
- Row Gap, Column Gap and Gap

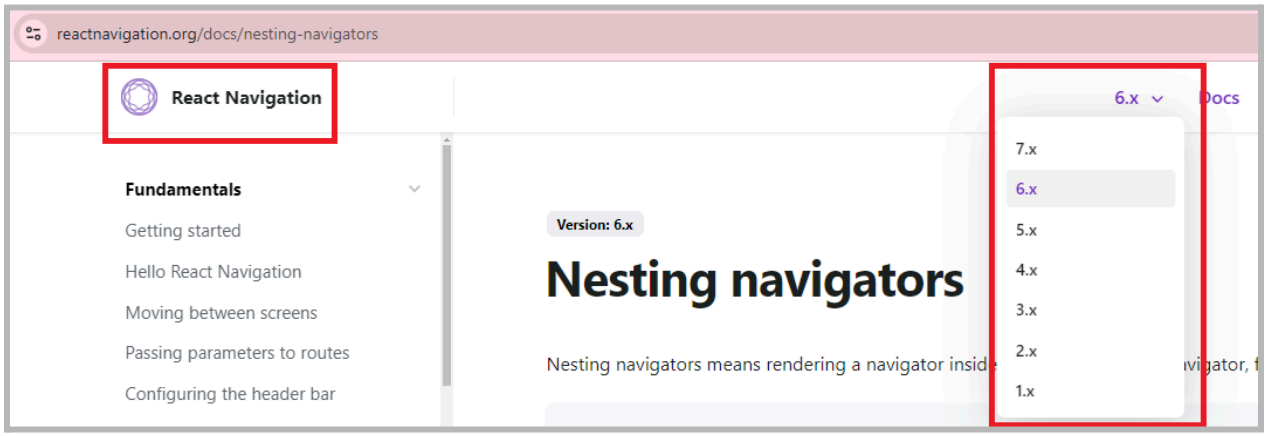
Về source code video này:
Các file thay đổi tại video, xem tại đây
Source code ứng với video này, xem tại đây
Video hướng dẫn cách download source sau khi kết thúc video này, xem tại đây

#41. Sử dụng Keyboard đa ngôn ngữ (Extra)
//todo

Chapter 5: Navigation với React Native
Điều hướng trang với React Native

#42. Tổng quan về chapter

Lưu ý: về chapter này, mình sử dụng tài liệu version 6 của React Navigation. Vì vậy, trong tương lai (nếu có sự thay đổi), vui lòng sử dụng version 6 của tài liệu trên https://reactnavigation.org/



Mục đích của chapter:

- **Hiểu về tư duy điều hướng trang (navigate) khi sử dụng mobile** (giống và khác gì so với website)

- **Sử dụng thư viện React Navigation**, công cụ này đặc biệt ở chỗ:
    + Bạn dùng React Native CLI, bắt buộc dùng

    + Bạn dùng Expo Router (behind the scenes), nó cũng dùng thư viện này
    Việc học từ đầu thư viện React Navigation giúp bạn dùng Expo Router (chapter sau) dễ dàng hơn

#43. Navigation (Routing) là gì ?
Tới thời điểm này, chúng ta đã hoàn thiện ứng dụng todo app, tuy nhiên, nó đang rất cơ bản và chưa thể hiện nhiều tính chất “thực tế”

1.Khái niệm Navigation (Routing)
Tương tự như website, người dùng sẽ điều hướng qua lại giữa các page (screen)

Ví dụ: bạn dùng website facebook thì:

- facebook.com === facebook.com**/** : đây là trang chủ, bạn lướt “feed”, bao gồm bài đăng, video
  Route (URL) = **/**

- facebook.com**/profile** : đây là trang cá nhân của bạn, thể hiện thông tin tài khoản bạn đã đăng nhập
  Route (URL) = **/profile**

- facebook.com**/groups**: thể hiện các groups bạn đã tham gia
  Route (URL) = **/groups**

Đối với React Native, việc điều hướng giữa các màn hình khác nhau, chính là sự thay đổi của các Route (URL), điểm khác biệt so với website, là bạn không nhìn thấy đường link URL, mà chỉ React Native nhìn thấy (việc của bạn là nói cho biết nó cần làm gì)

2.Có bao nhiêu cách Routing ?

Có 2 cách chính hay dùng:

Cách 1: sử dụng thư viện React Navigation
https://reactnavigation.org/

Cách làm này bạn dùng React Native CLI chắc chắn dùng

Cách 2: sử dụng Expo Router (tích hợp bên trong framework Expo)

Fact: Expo là team đứng sau React Navigation (maintain/develop)

#44. Setup React Navigation
Tài liệu: https://reactnavigation.org/docs/getting-started

Lưu ý: version cài đặt trong khóa học là version 6.x

1.Cài đặt thư viện
Phân biệt sự khác nhau của các câu lệnh
Cách 1: npm install tên-thư-viện
=> bạn dùng npm để cài đặt như thông thường

Cách 2: expo install tên-thư-viện
=> bạn dùng expo cli để cài đặt thư viện.
Ưu điểm: expo sẽ tự động tìm version của thư viện phù hợp, ứng với version của expo bạn cài đặt trong project
Nhược điểm: **bạn cần cài đặt expo CLI**

Cách 3: npx expo install tên-thư-viện
Tác dụng: y hệt như cách 2, tuy nhiên, **bạn KHÔNG cần cài đặt expo CLI**

Lưu ý: trong quá trình thực hiện khóa học, bạn vui lòng không tự cài đặt thư viện, mà sử dụng các câu lệnh tác giả cung cấp

- Áp dụng khi sử dụng SDK 51:
npm i --save-exact @react-navigation/native@6.1.18
react-native-safe-area-context@4.10.5 react-native-screens@3.31.1

2. Có bao nhiêu cách điều hướng (navigation) trên mobile ?
Có 3 cách chính:
- Sử dụng Stack
- Sử dụng Bottom Tab
- Sử dụng Drawer

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

#45. Sử Dụng Stack Navigation
Tài liệu: https://reactnavigation.org/docs/hello-react-navigation
Demo: https://reactnavigation.org/assets/navigators/stack/stack.mp4

1.Cài đặt
- Áp dụng khi sử dụng SDK 51:
npm i --save-exact @react-navigation/native-stack@6.11.0

Phân biệt sự khác biệt giữa @react-navigation/**native-stack** và @react-navigation/**stack**

"native" stack: sử dụng native component để điều hướng (UINavigationController on iOS and Fragment on Android)
=> tối đa về performance, tuy nhiên ít customize

stack: sử dụng javascript để điều hướng
=> performance không cao bằng, tuy nhiên lại flexible về customize

Native-Stack	Stack
- Tài liệu tại đây - Trông giống nhất và performance tốt nhất (native look & feel + performance) - Hạn chế về khả năng customize - Sẽ có những animation chỉ có trên Android/IOS	- Tài liệu tại đây - Tối đa về customization - Đồng bộ về animation giữa Android và IOS

Khuyến khích sử dụng native-stack trước, khi nào nó không đáp ứng được yêu cầu của project (bài toán gặp phải), thì đấy là lúc sử dụng stack

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

#46. Moving Between Screens
Tài liệu: https://reactnavigation.org/docs/navigating

Sử dụng hook:
https://reactnavigation.org/docs/use-navigation

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

#47. Passing parameters to routes
Tài liệu: https://reactnavigation.org/docs/params

Sử dụng hook: https://reactnavigation.org/docs/use-route

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

#48. Customize Header (Extra)
Hướng dẫn cách customize header
Tài liệu: https://reactnavigation.org/docs/headers

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

#49. Sử Dụng Drawer Navigation
Tài liệu: https://reactnavigation.org/docs/drawer-navigator

1.Cài đặt
- Áp dụng khi sử dụng SDK 51:
npm i --save-exact @react-navigation/drawer@6.7.2
react-native-gesture-handler@2.16.1 react-native-reanimated@3.10.1

Ví dụ: https://reactnavigation.org/docs/drawer-navigator#example

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

#50. Sử Dụng Bottom Tab Navigation
Tài liệu: https://reactnavigation.org/docs/tab-based-navigation

Cài đặt:
- Áp dụng khi sử dụng SDK 51:
npm i --save-exact @react-navigation/bottom-tabs@6.6.1

Ví dụ: https://reactnavigation.org/docs/bottom-tab-navigator#example

https://icons.expo.fyi/Index/Ionicons/aperture

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

#51. Nesting Navigators
Tài liệu: https://reactnavigation.org/docs/nesting-navigators

Yêu cầu: tạo các component sau
Home
HomeDetail
Like
LikeDetail
About
ChangePassword

Rule bạn cần tuân theo khi sử dụng nested navigator

1.Sử dụng Stack kết hợp với Bottom Tabs
//todo
Stack là cha, Tabs là con

2. Sử dụng Stack + Bottom Tabs + Drawer
//todo
Drawer là cha

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 6: Expo Router
Sử dụng React Navigation với Expo Framework

#52. Tổng quan về chapter

Kế thừa lại thành quả của chương học trước (chapter 5), chúng ta đã hình dung được với React Native, có 3 kiểu điều hướng chính:

- Stack
- Drawer
- Tab

Tuy nhiên, nếu chúng ta code thuần với React Navigation, code khá dài, vì vậy, chapter này ra đời

Mục tiêu:

- Sử dụng cơ chế routing của Expo Router (thông qua đặt tên folder)

- Thấy được điểm tương đồng giữa React-Navigation (chapter 5) và Expo Router

#53. Hello World với Expo Router
Video này chúng ta sẽ test ứng dụng Expo đã tích hợp sẵn Router

1. Cài đặt dự án

Link dự án thực hành chapter này tai đây

Bước 1: clone dự án

Bước 2: Cài đặt thư viện

Bước 3: chạy dự án

Lưu ý: React Native có thể chạy trên web :v
=> React Native chạy trên android/ios/web

Các tính năng có trong dự án:

- File-based routing (tương tự bạn dùng Next.js)
- Sử dụng image
- Sử dụng custom font
- Light/dark mode (hướng dẫn cách on/off dark mode)
- Animation (khi vào app có waveHand)

2. Why Expo Router ?
https://docs.expo.dev/router/introduction/

//todo

#54. Cách Mình Setup Project (Extra)
Tương tự như video #23 (tạo base project mà không có expo-router), video này là cách mình tạo project Expo đã tích hợp sẵn router

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.

Tài liệu: https://docs.expo.dev/get-started/create-a-project/

#55. Setup Expo Router
Tài liệu: https://docs.expo.dev/router/installation/

Ôn lại: Tại Sao chúng ta cần Expo Router
Chúng ta sử dụng Expo Router là một cách khác để điều hướng (navigate) giữa các màn hình khác nhau (screen), và cách làm này đơn giản và ngắn gọn so với chapter 5

Bước 1: Cài đặt thư viện
- Áp dụng khi sử dụng SDK 51:
npm i --save-exact expo-router@3.5.23 expo-linking@6.3.1 expo-constants@16.0.2

Bước 2: update file
//package.json
"main": "expo-router/entry"

//app.json
```
  "expo": {
    "scheme": "hoidanit-app",
    "plugins": [
      "expo-router"
    ],
    "experiments": {
      "typedRoutes": true
    }
  }
```

Bước 3: Tạo src folder
https://docs.expo.dev/router/reference/src-directory/

Tạo folder src => app

Chạy project với câu lệnh sau:
npx expo start -c

Bước 4: Viết Hello World

Bên trong thư mục app, tạo file **index.tsx**

Bước 5: Tổ chức lại source code

Xóa file App.tsx

Move tất cả thư mục còn lại vào thư mục src

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

#56. Create Pages
Tài liệu: https://docs.expo.dev/router/create-pages/

Lưu ý: với Next.js (làm web), bạn đặt tên file là page.tsx, với Expo Router là index.tsx

- Để tạo 1 đường link url (1 screen), bạn cần **tạo bên trong thư mục app**
Tên folder, tên file sẽ quyết định đường link url.

- Bạn có thể sử dụng các tiền tố .js, .jsx, .ts hoặc .tsx. Tuy nhiên, khi dùng typescript, vui lòng **sử dụng .tsx**

- Tạo Dynamic routes

- Basic Navigate: https://docs.expo.dev/router/navigating-pages/

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

#57. Navigate between pages
Tài liệu: https://docs.expo.dev/router/navigating-pages/

1.Navigate
Có 2 cách để navigate:
- Sử dụng component Link
- Sử dụng hook (ví dụ, khi bạn login, bạn gọi api, nếu thành công, redirect về home)

Phân biệt navigate, push, replace:

Tip: ưu tiên dùng navigate (default). Nếu khi nào có vấn đề xảy ra, đấy là khi dùng push/replace

Push: đẩy thêm screen vào stack (mà không quan tâm screen ấy đã tồn tại hay chưa)

Replace: ghi đè screen đang đúng hiện tại

2. Group Routes
https://docs.expo.dev/router/layouts/#groups

- Dùng để gom nhóm route, mà không tạo ra đường link url

3. Layout
https://docs.expo.dev/router/layouts/

- File chạy đầu tiên
- Dùng để chia sẻ data giữa các component

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

#58. Stack Navigation
Tài liệu: https://docs.expo.dev/router/advanced/stack/

Chúng ta cần stack, để hỗ trợ điều hướng qua lại giữa các màn hình (header/back...)

//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

#59. Bottom Tab Navigation
Tài liệu: https://docs.expo.dev/router/advanced/tabs/

Nguyên tắc: ưu tiên stack là cha, tabs là con

//sử dụng kết hợp với stack

//chưa làm drawer (được giới thiệu khi kết thúc khóa học)

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

#60. Tổng kết về Expo Router

Tài liệu: https://docs.expo.dev/router/introduction/

Khuyến khích tự đọc & tìm hiểu thêm về expo-router

- Sử dụng cơ chế file-based router
- Điều hướng trang với **Link** hoặc **router** function
- Tư duy giống hệt React Navigation
- Sử dụng Stack/Bottom Tab navigation
- Sử dụng Nested Navigator

Chapter 7: Setup Dự Án Thực Hành
Setup backend và tìm hiểu tổng quan về dự án thực hành

#61. Tổng quan về chapter

Ý tưởng của dự án thực hành: Clone Shopee Food

Tham khảo: https://shopeefood.vn/

Chi tiết về dự án, mình đã có video demo chi tiết tại #3

Mục tiêu của chương học:

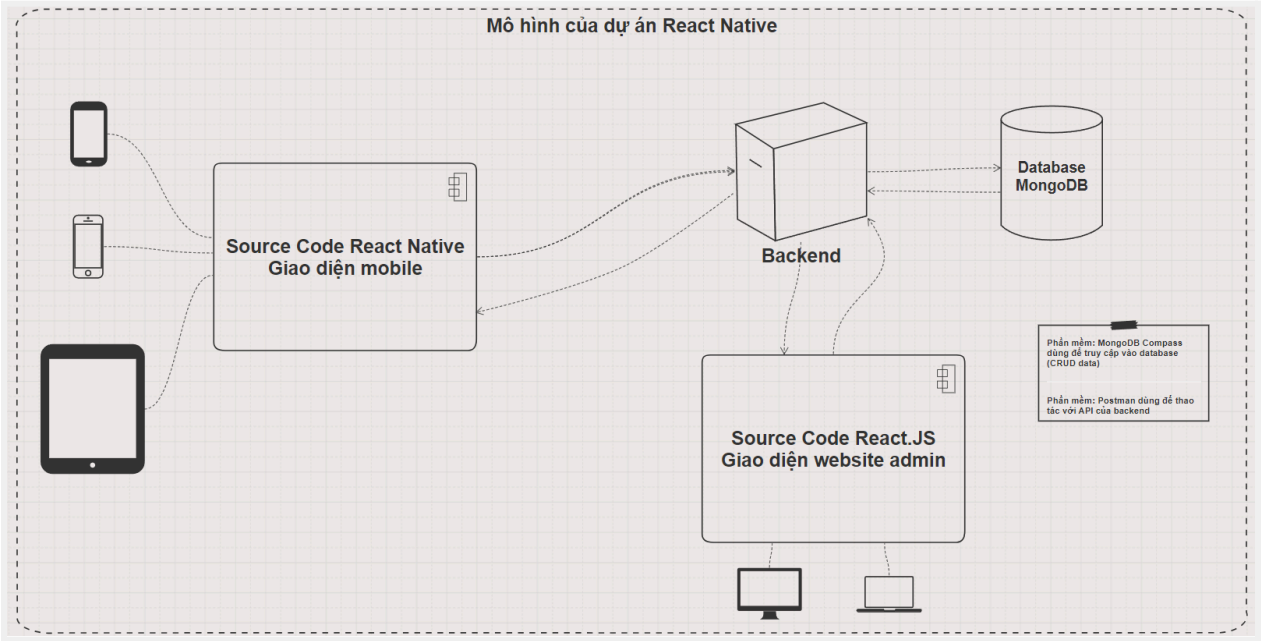
- Phân tích về cấu trúc (thiết kế) dự án thực hành

  Tham khảo file design giao diện tại đây

- Cài đặt các công cụ cần thiết phục vụ cho React Native, bao gồm backend và giao diện quản lý website

#62. Phân Tích Dự Án Thực Hành

Các thành tham gia vào dự án



Ý nghĩa của các thành phần:

- **App Mobile (React Native)** : đây là kiến thức khóa học cung cấp (code từ A tới Z) Sử dụng mobile app phục vụ client (khách hàng)

- **Giao diện Website** (viết bằng React/Antd): source code cung cấp sẵn, chỉ sử dụng và không sửa đổi.
  Bạn không học code Website trong khóa học này. Bạn muốn học website, tham khảo lộ trình khóa học tại đây

- **Backend** (viết bằng Nest.js): source code cung cấp sẵn, chỉ sử dụng và không sửa đổi.
  Bạn không học code Backend trong khóa học này. Bạn muốn học backend, tham khảo khảo lộ trình Nodejs tại đây, hoặc lộ trình Java Spring tại đây

  MongoDB Atlas là bên dịch vụ cung cấp database MongoDB. Lý do mình chọn MongoDB vì nó miễn phí và có khả năng sử dụng online (không cần cài đặt)

#63. Cài đặt MongoDB Compass

MongoDB Compass không phải là database. Nó chỉ đơn thuần là một phần mềm cung cấp giao diện, 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

#64. Tạo Tài Khoản MongoDB Atlas

Link đăng ký tài khoản MongoDB Atlas (miễn phí)
https://www.mongodb.com/cloud/atlas/register

#65. Tạo Database cho dự án
Lưu ý check allow anywhere

Lưu lại thông tin kết nối tới database

#66. Kiểm Tra Kết Nối Database

Công cụ: phần mềm Mongodb Compass

Lưu lại connection để tiện sử dụng

#67. Kích hoạt dự án Backend

Lưu ý : học viên sử dụng tài khoản Udemy Business, vui lòng inbox qua Facebook Hỏi Dân IT để được mình hỗ trợ kích hoạt backend.

Bước 1: Clone dự án thực hành tại đây

Project cần làm trong video, có tên là : **share-create-id-react-native-basic**

Bước 2: Chạy dự án (chi tiết ghi tại file README.md)

- Chạy câu lệnh : **npm i** để cài đặt các thư viện cần thiết

- Chạy dự án với câu lệnh: **npm start**

Bước 3: Lấy file kết quả ID
Nếu không có lỗi gì xảy ra, sẽ có message thông báo Done.
Kết quả được lưu tại file **/dist/submit-license/result.txt**

Bước 4: Submit kết quả
Truy cập: https://active.hoidanit.vn/license

Tại đây, bạn cần cung cấp các thông tin cần thiết, bao gồm:

- **Tên khóa học**: chọn **React Native Basic**
- Tên tài khoản Facebook mua khóa học
- URL tài khoản Facebook mua khóa học
- Email đăng ký của học viên khi mua khóa học

cũng như **file result.txt** (có được tại bước 3)
Nhấn submit.

Bước 5: Nhận kết quả
Sau khi Submit kết quả ở trên, bạn chủ động inbox Facebook Hỏi Dân IT:
https://www.facebook.com/askITwithERIC/

Mình sẽ gửi kết quả qua email.
Thời gian phản hồi là 24h (từ thứ 2 tới thứ 6). 1 tới 3 ngày đối với thứ 7 & chủ nhật (ngày lễ/holiday)
Làm bước 5 là thành công rồi, chuyển qua video tiếp theo các bạn nhé.

#68. Setup Dự Án Backend

Lưu ý 1: bạn cần xem và thực hành #67 trước khi xem video này.
Cần được xác nhận kết quả submit qua email cũng như license key để thực hiện.

Lưu ý 2: học viên sử dụng tài khoản Udemy Business, vui lòng inbox qua Facebook Hỏi Dân IT để được mình hỗ trợ kích hoạt backend.

Bước 1: Clone dự án thực hành tại đây

Project cần làm trong video, có tên là : **share-backend-react-native-basic**

Bước 2: Chạy dự án

- Cài đặt thư viện cần thiết với câu lệnh: **npm install --omit=dev**
- Update file **.env:**
  Cập nhật tham số cho database **MONGODB_URI**
  Cập nhật license key (đã gửi trong email kích hoạt) **ACTIVE_KEY**

- Chạy dự án với câu lệnh: **npm start**

Bước 3: Kích hoạt bản quyền
Sử dụng kết quả nhận được email tại video #43, **cập nhật file result.txt**

Bước 4: Chạy lại dự án: npm start
Test dự án bằng cách:
Truy cập http://localhost:8080/

#69. Setup Postman && Google App Password

1.Cài đặt PostMan

Download postman: https://www.postman.com/downloads/

Tạo tài khoản = tài khoản google để đồng bộ giữa nhiều thiết bị

2.Sử dụng collection

Link file collection của dự án thực hành tại đây

//todo

Setup environments variables, bao gồm backend url và access token

Test api login và api get user (sử dụng access token)

3. Setup Google App Password

Mục đích : gửi email với Gmail

- Tài khoản Gmail cần bật xác thực 2 lớp

https://myaccount.google.com/apppasswords

- Sử dụng tài khoản email test (không nên dùng email chính của bạn) để giảm thiểu tối đa rủi ro (ví dụ bạn "vô tình" public source code dẫn tới lộ app password)

//testing gửi email với cái api hello world /email

#70. Setup Admin Web

Lưu ý: bạn cần xem và thực hành #67 trước khi xem video này.

Bước 1: Clone dự án thực hành tại đây

Project cần làm trong video, có tên là : **share-frontend-react-native-basic**

Chapter 8: Module Signup

Thực hiện chức năng đăng ký tài khoản người dùng

#71. Tổng quan về chapter

Mindset chia layout với flex

Về UI/UX:

- Css text between line, background gradient
- Reuse button, input
- Input password (show/hide)
- Hiệu ứng loading

Gọi api, cách debug debug với React native

- Hoàn thiện tính năng register tài khoản có gửi email

#72. Design Welcome Screen

Lưu ý: initialRouteName không hoạt động tốt với Expo-Router
Tham khảo tại đây

File design giao diện, tham khảo tại đây

1.Cách tư duy khi chia layout
//todo

2.Style giao diện
//todo

- Có thể sử dụng flex dưới dạng số thập phân
  Ví dụ: **flex: 0.5**

- Background fit-content, sử dụng
  alignSelf: "flex-start"

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

#73. Reuse Component Button (Part 1)
Tài liệu:
Link tham khảo file button tại đây

//tái sử dụng color
backgroundColor: "#f4511e",

Mục tiêu: tái sử dụng button
Các tiêu chí của button:
- width "fit-content", hoặc full 100%
- truyền động padding, borderRadius, backgroundColor, textColor (textStyles)
- truyền động icons (có thể có hoặc không)

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

#74. Reuse Component Button (Part 2)
Tài liệu:
https://icons.expo.fyi/Index

//Cấu hình typescript cho absolute path (vs relative path)
https://docs.expo.dev/guides/typescript/#path-aliases-optional

//tsconfig.json => cần restart project

```
 "compilerOptions": {
   "strict": true,
   "baseUrl": ".",
   "paths": {
     "@/*": ["src/*"],
     "components/*": ["src/components/*"],
     "utils/*": ["src/utils/*"]
   }
 }
```

//CSS text between line
https://stackoverflow.com/a/5214204
//CSS layout
https://icons.expo.fyi/Index

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

#75. Sử dụng Image/ImageBackground

1.Sử dụng image
Link download images tại đây

https://reactnative.dev/docs/image

https://reactnative.dev/docs/imagebackground

Cấu hình typescript sử dụng image:
https://stackoverflow.com/a/69944521

2.Sử dụng Linear gradient
Tài liệu : https://docs.expo.dev/versions/v51.0.0/sdk/linear-gradient/

Cài đặt thư viện:
- Áp dụng khi sử dụng SDK 51:
npm i --save-exact expo-linear-gradient@13.0.2

colors={['transparent', '#191B2F']}
locations={[0.2, 0.8]}

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

#76. Design Sign Up Screen

Tài liệu: https://reactnative.dev/docs/textinput

//todo

Sử dụng Redirect với expo:
https://docs.expo.dev/router/reference/redirects/

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

#77. Reuse Input Group

Sử dụng SafeAreaView
https://docs.expo.dev/versions/v51.0.0/sdk/safe-area-context/

Lưu ý: không dùng của React Native, vì chỉ hỗ trợ IOS
https://reactnative.dev/docs/safeareaview

Mục tiêu:

- Reuse Input Group
- Xử lý hiệu ứng borderColor với onFocus/onBlur
- Hoàn thiện base giao diện

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

#78. Hide/Show Password

1.Hide/Show Password
Sử dụng **secureTextEntry**
https://reactnative.dev/docs/textinput#securetextentry

Sử dụng icons: https://icons.expo.fyi/Index

import FontAwesome5 from '@expo/vector-icons/FontAwesome5';

<FontAwesome5 name="**eye**" size={24} color="black" />

<FontAwesome5 name="**eye-slash**" size={24} color="black" />

Position vời React Native:
https://reactnative.dev/docs/layout-props#position

Default: **relative** (vị trí so với vị trí gốc của nó)
Absolute: vị trí so với cha gần nhất của nó (relative)
https://www.w3schools.com/css/css_positioning.asp

2.State hóa form data
// Nhấn submit lấy được data ở trên form

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

#79. Gọi API với React Native

Lưu ý: cần phải đảm bảo rằng bạn đã setup thành công backend (đã kích hoạt bản quyền thành công tại chapter 7)

Bước 1: chạy source code backend
Kiểm tra với localhost có vào đc backend hay không ?
http://localhost:8080/

Bước 2: Cài đặt thư viện
https://www.npmjs.com/package/axios

npm i --save-exact axios@1.7.5

Bước 3: Test API
//setup env for expo
https://docs.expo.dev/guides/environment-variables/

Truy cập: http://localhost:8080/

Nếu bạn dùng điện thoại của chính bạn để test, xem video #168 để biết cách kết nối từ điện thoại tới localhost

Lưu ý về đường link localhost cho máy ảo android/ios
https://stackoverflow.com/a/66195215
.\adb devices

Trên android, sẽ localhost là chạy trong máy ảo, vì vậy:
Sử dụng 10.0.2.2 thay vì localhost

Check code đang chạy trên android hay ios:
https://reactnative.dev/docs/platform#os

C:\Users\your-computer-username\AppData\Local\Android\Sdk\platform-tools
.\adb -s emulator-5554 reverse tcp:8080 tcp:8080

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

#80. API Sign Up

Cần đảm bảo rằng, bạn đã chạy thành công video #69, đã cấu hình phần gửi email
Video #70, setup thành công web admin

Bước 1: Test gửi email với với Postman
(Chạy với câu lệnh **npm start** hoặc **npm run dev**)

Chạy dự án backend
Chạy dự án frontend admin web (đăng nhập: admin@gmail.com/123456)

Logic của luồng register:

- Giao diện react native sẽ gọi API backend để tạo mới tài khoản

- Nếu tạo mới thành công, backend sẽ:
  Trả ra id của user tạo mới
  Tài khoản ở trạng thái chưa kích hoạt
  Gửi code kích hoạt tới email đăng ký

- React Native sẽ cần redirect tới screen nhập mã code xác nhận

Bước 2: gọi api với axios
//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

#81. Cách Debug với React Native (Extra)
Để bắt đầu video này, đừng quên chạy dự án backend bạn nhé

Tài liệu: https://reactnative.dev/docs/debugging
https://docs.expo.dev/debugging/tools/

Debug được sử dụng hiệu quả nhất khi:

- Bạn muốn xem code chạy từng dòng một
- Bạn muốn check giá trị của biến số một cách linh hoạt
- Code của bạn bị lỗi, và bạn không biết nó sai ở đâu

1.Basic
Sử dụng câu lệnh **console.log**
Ưu điểm: đơn giản, dễ dùng
Nhược điểm: đối với các biến số có giá trị phức tạp (object), hoặc không thể xem code chạy từng dòng một

2.Cách debug hay dùng nhất

Developer menu:
Nếu bạn test = điện thoại của bạn, “lắc điện thoại” (shake) để mở devtool

Debug http request:
https://docs.expo.dev/debugging/tools/#debugging-with-chrome-devtools

- Cách mở debug
- Cách đặt mở file cần debug (ctrl + P hoặc debugger)
- Cách đặt breakpoint
- Xem code chạy từng dòng và check giá trị của biến

Lưu ý: react native caching data
Demo với api bên ngoài, status 304

Về source code video này:
Các file thay đổi tại video, xem tại đây
Source code ứng với video này, xem tại đây

#82. Cấu hình Axios Instance

Để bắt đầu video này, đừng quên chạy dự án backend bạn nhé

Mục đích: tái sử dụng code và giúp code có thể bảo trì được.

Instance là bản sao của axios, giúp bạn cấu hình nó. Hãy tưởng tượng, bạn sử dụng class của java, bạn muốn thao tác với class, cần tạo object (đối tượng cụ thể), thì instance ở đây chính là "object của java"

Một trường hợp hay gặp nhất là frontend gọi tới nhiều server khác nhau, ví dụ:

- Server A (https://serverA.com) => bạn cấu hình instance A
- Server B (https://serverB.com) => bạn cấu hình instance B

Về axios: https://github.com/axios/axios

Tại github, nhấn phím Ctrl + F để search

Cấu hình instance:
https://github.com/axios/axios?tab=readme-ov-file#creating-an-instance

```
const instance = axios.create({
  baseURL: 'https://some-domain.com/api/',
  timeout: 1000,
  headers: {'X-Custom-Header': 'foobar'}
});
```

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

#83. Cấu hình Axios Interceptor

Để bắt đầu video này, đừng quên chạy dự án backend bạn nhé

Mục đích: tái sử dụng code và giúp code có thể bảo trì được.

1.Cấu hình Axios Interceptor
Interceptor được sinh ra, để giúp bạn can thiệp vào request và response

Với request: trước khi gọi API, bạn có thể “modify” request, như thêm access token vào header chẳng hạn

Với reponse, trước khi client nhận được phản hồi, bạn có thể format data theo định dạng mong muốn

//cấu hình interceptor
https://github.com/axios/axios?tab=readme-ov-file#interceptors

```
// Add a request interceptor
axios.interceptors.request.use(function (config) {
    // Do something before request is sent
    return config;
  }, function (error) {
    // Do something with request error
    return Promise.reject(error);
  });

// Add a response interceptor
axios.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;
  }, function (error) {
    // Any status codes that falls outside the range of 2xx cause this function to trigger
    // Do something with response error
    return Promise.reject(error);
  });
```

2.Cấu hình Typescript với Axios

//định nghĩa global type
https://bobbyhadz.com/blog/typescript-make-types-global#declare-global-types-in-typescript

```
export {};

declare global {
  interface Employee {
//
  }
 }
```

//setup typescript for axios

```
// https://github.com/axios/axios/issues/1510#issuecomment-448201698
import axios from 'axios';
declare module 'axios' {
   export interface AxiosResponse<T = any> extends Promise<T> { }
}


export interface IBackendRes<T> {
   error?: string | string[];
   message: string;
   statusCode: number | string;
   data?: T;
}
```

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

#84. Hiển Thị Thông Báo Lỗi

Để bắt đầu video này, đừng quên chạy dự án backend bạn nhé

1.Hiển thị thông báo lỗi
Sử dụng toast: https://docs.expo.dev/ui-programming/react-native-toast/

https://github.com/magicismight/react-native-root-toast

Cài đặt thư viện:
- Áp dụng khi sử dụng SDK 51:
npm i --save-exact react-native-root-toast@3.6.0

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

#85. Design Verify Code Screen

Tham khảo file design giao diện tại đây

Tài liệu:
https://www.npmjs.com/package/react-native-otp-textinput

https://github.com/naveenvignesh5/react-native-otp-textinput

Cài đặt thư viện:
- Áp dụng khi sử dụng SDK 51:
npm i --save-exact react-native-otp-textinput@1.1.6

// @ts-ignore:next-line

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

#86. API Verify Account (Part 1)

Để bắt đầu video này, đừng quên chạy dự án backend + frontend bạn nhé

//fix lỗi import image với typescript
https://github.com/microsoft/TypeScript/issues/55019

//input autofocus

//sử dụng overlay với activity indicator
Ẩn keyboard nếu nhập 6 ký tự
https://stackoverflow.com/questions/44046500/how-do-i-overlay-activityindicator-in-react-native

Tải file overlay tại đây (#86)
//xử lý sự kiện onChange để gọi API, **chưa sử dụng ref**

```
  const handleCellTextChange = async (text: string, i: number) => {
    console.log(">>> check text: ", text, " and i = ", i)
  }
```

//gọi API verify account (lưu ý đang hardcode email)
nếu có lỗi, hiển thị thông báo lỗi (làm basic)
thành công, redirect về login

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

#87. API Verify Account (Part 2)
//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

#88. API Resend Code

Để bắt đầu video này, đừng quên chạy dự án backend + frontend bạn nhé

Gọi API resend code (cần truyền vào email của người dùng)

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

#89. Tổng kết về chapter

Ưu điểm:

- Hoàn thiện tính năng signup (kết hợp gửi email)
- Design UI/UX

Nhược điểm: chưa tối ưu hóa về form, điều này sẽ được làm tại chapter tiếp theo

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 9: Module Login

Thực hiện chức năng đăng ký tài khoản người dùng

#90. Tổng quan về chapter

- Hoàn thiện logic Login

- Sử dụng Formik để tối ưu hóa hiệu năng (uncontrolled component), đồng thời validate dữ liệu tại frontend (trước khi gọi API)

#91. Bài Tập Login Screen
(Bài tập này sẽ được chữa tại #92)

Yêu cầu:
Tham khảo file design giao diện tại đây

Từ màn hình welcome, nhấn nút “Start with email” sẽ chuyển qua màn hình login
Design giao diện tương tự màn hình signup

Quản lý state với React, khi submit form, lấy được email/password
Lưu ý: chưa cần validate dữ liệu tại phía frontend

Gợi ý cách làm:
Bước 1: Khai báo Stack để ẩn header của màn hình login

Bước 2: Tạo bố cục (style nếu cần thiết), tương tự màn hình signup

Bước 3: Quản lý input với state của React

Xử lý sự kiện onPress của button, console.log ra giá trị của input email/password

#92. Chữa Bài Tập Login 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

#93. Bài Tập API Login

Để bắt đầu video này, đừng quên chạy dự án backend + frontend bạn nhé

Logic của luồng đăng nhập:

- Frontend React Native gửi username/password vào API backend.
  Lưu ý là tên trường đặt là username, không phải là email nhé.
  Trong quá trình code, có thể sử dụng Chrome Dev Tool để check payload khi gọi API (kết hợp test API với Postman)
- Backend sẽ kiểm tra thông tin và trả ra phản hồi cho frontend
- Nếu như có thông báo lỗi, hiển thị thông báo lỗi
  Có thể console.log phản hồi của API trong trường hợp lỗi để nắm rõ data

- **Nếu thông báo lỗi là "tài khoản chưa được kích hoạt", ứng với mã lỗi statusCode = 400**, cần chuyển hướng người dùng về trang verify-code
- Nếu login thành công, chuyển hướng người dùng về trang /home

  (Để có tài khoản active/inactive, bạn có thể sử dụng giao diện web admin để update user)

Gợi ý cách làm:

Bước 1: gọi API

Bước 2: xử lý kết quả trả ra của api

- Nếu không có lỗi, redirect người dùng về homepage
- Nếu có lỗi xảy ra:
  + Lỗi sai mật khẩu/email, chỉ hiển thị thông báo

  + **Lỗi tài khoản chưa kích hoạt (ứng với mã lỗi statusCode = 400), xử lý như sau:**
    Mục tiêu là tái sử dụng lại màn hình verify-code (do đã có tính năng verify code và resend code)
    => cần truyền thêm tham số trước khi redirect sang màn hình verify-code
    Ví dụ: router.replace({
    pathname: "/(auth)/verify",
    params: { email: email, **isLogin: 1**}
    })

    Dựa vào tham số trên, để sau khi verify-code thành công, với màn hình login, sẽ redirect về homepage (còn màn hình register, redirect về login)

#94. Chữa Bài Tập API Login
//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

#95. Hiển Thị Loading (Extra)
Để bắt đầu video này, đừng quên chạy dự án backend bạn nhé

Mục tiêu: tăng trải nghiệm của người dùng (UX)

1.Tạo hiệu ứng “delay”
Trong thực tế, khi bạn gọi API giữa các hệ thống khác nhau, sẽ có độ trễ của phản hồi (delay)

Minh họa api với header “delay” (test với postman và sử dụng debug devtool)
Lưu ý: Hiệu ứng delay là do backend đã code sẵn, không phải tự nhiên mà có :v

Set header cho axios: https://stackoverflow.com/a/45581882

2.Sử dụng Activity Indicator
https://reactnative.dev/docs/activityindicator

Sử dụng disabled pressable : https://reactnative.dev/docs/pressable#disabled

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

#96. Giới thiệu về Formik

1.Vấn đề còn tồn đọng
Frontend chưa hề validate dữ liệu trước khi submit API.

Một ứng dụng thực tế, cần validate dữ liệu cả frontend lẫn backend (an toàn nhất).
Nếu chỉ validate tại backend, backend sẽ tiếp nhận request không cần thiết (các request không hợp lệ), với hệ thống có nhiều lượt truy cập, dễ dẫn tới tốn kém nguồn tài nguyên server (và tài nguyên là tiền - money)

Với frontend, sau khi validate, cần hiển thị thông báo lỗi.
Vấn đề gì sẽ xảy ra nếu có quá nhiều input (case này ít gặp với react native, cơ mà rất hay gặp khi làm reactjs)

2. Controlled component vs uncontrolled component
Controlled : kiểm soát, sử dụng state của React

Uncontrolled: không kiểm soát, lấy data bằng cách dùng Ref và DOM

3. Giải pháp đề ra
Sử dụng các thư viện để đơn giản hóa quá trình, đồng thời tối ưu hóa hiệu năng

Cài đặt thư viện formik:

npm i --save-exact formik@2.4.6

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

Về source code video này:
Các file thay đổi tại video, xem tại đây
Source code ứng với video này, xem tai đây
Video hướng dẫn cách download source sau khi kết thúc video này, xem tai đây

#97. Sử dụng Formik với React Native

1.Tích hợp vào React Native
https://formik.org/docs/guides/react-native

```
<Formik
   initialValues={{ email: '', password: '' }}
   onSubmit={values => console.log("check values = ", values)}
>
   {({ handleChange, handleBlur, handleSubmit, values }) => (
      <View style={{ margin: 10 }}>
         <Text>Email</Text>
         <TextInput
            style={{ borderWidth: 1, borderColor: "#ccc" }}
            onChangeText={handleChange('email')}
            onBlur={handleBlur('email')}
            value={values.email}
         />
         <View style={{ marginVertical: 10 }}></View>
         <Text>Password</Text>
         <TextInput
            style={{ borderWidth: 1, borderColor: "#ccc" }}
            onChangeText={handleChange('password')}
            onBlur={handleBlur('password')}
            value={values.password}
         />
         <View style={{ marginVertical: 10 }}></View>

         <Button onPress={handleSubmit as any} title="Submit" />
      </View>
   )}
</Formik>
```

2.Tích hợp validate

Yup vs joi vs zod

https://formik.org/docs/guides/validation

Cài đặt thư viện:

npm i --save-exact yup@1.4.0

```
 const SignupSchema = Yup.object().shape({
   firstName: Yup.string()
     .min(2, 'Too Short!')
     .max(50, 'Too Long!')
     .required('Required'),
   lastName: Yup.string()
     .min(2, 'Too Short!')
     .max(50, 'Too Long!')
     .required('Required'),
   email: Yup.string().email('Invalid email').required('Required'),
 });
```

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

#98. Login với Formik
//todo
https://github.com/stackworx/formik-mui/issues/172

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

#99. Bài Tập Signup với Formik
//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

#100. Tổng kết về chapter
//todo

Chapter 10: Module Homepage
Design giao diện trang chủ và xem chi tiết nhà hàng/quán ăn

#101. Tổng quan về chapter

Các kiến thức sẽ học trong chương học này:

- Phân tích layout của ứng dụng Shopee Food
- Design layout với sticky Header
- Design base giao diện HomePage (hardcode data)

#102. Hướng Dẫn Cài App Trên Máy Ảo (Extra)
Mục tiêu: Cài đặt app ShopeeFood để lấy ý tưởng design

#103. Phân Tích Layout ứng dụng

Mục tiêu: tích sticky header, dựng base layout

Sticky search
Banner
List Category

Các category nổi bật

Clone source code tại đây

Design list category:
https://stackoverflow.com/questions/45939823/react-native-horizontal-flatlist-with-multiple-rows

#104. Tạo Base Layout Home 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

#105. Sử dụng Carousel

Tài liệu:
https://www.npmjs.com/package/react-native-reanimated-carousel

https://docs.expo.dev/versions/latest/sdk/view-pager/

Mục tiêu:
- Tích hợp thư viện carousel
- Slider với pagination

Bước 1:
Cài đặt thư viện:

npm i --save-exact react-native-reanimated-carousel@4.0.0-canary.14

https://react-native-reanimated-carousel.vercel.app/

Bước 2: Tích hợp ví dụ
https://react-native-reanimated-carousel.vercel.app/usage

https://github.com/dohooo/react-native-reanimated-carousel/issues/606

Bước 3: cấu hình **GestureHandlerRootView**
Ví dụ mình sử dụng trong video:
https://docs.swmansion.com/react-native-gesture-handler/docs/#apps

Về source code video này:
Các file thay đổi tại video, xem tại đây
Source code ứng với video này, xem tai đây
Video hướng dẫn cách download source sau khi kết thúc video này, xem tai đây

```
import * as React from "react";
import { Dimensions, Text, View } from "react-native";
import { useSharedValue } from "react-native-reanimated";
import Carousel, {
  ICarouselInstance,
  Pagination,
} from "react-native-reanimated-carousel";

const data = [...new Array(6).keys()];
const width = Dimensions.get("window").width;

function App() {
  const ref = React.useRef<ICarouselInstance>(null);
  const progress = useSharedValue<number>(0);

  const onPressPagination = (index: number) => {
    ref.current?.scrollTo({
      /**
       * Calculate the difference between the current index and the target index
       * to ensure that the carousel scrolls to the nearest index
       */
      count: index - progress.value,
      animated: true,
    });
  };

  return (
    <View style={{ flex: 1 }}>
      <Carousel
        ref={ref}
        width={width}
        height={width / 2}
        data={data}
        onProgressChange={progress}
        renderItem={({ index }) => (
          <View
            style={{
              flex: 1,
              borderWidth: 1,
              justifyContent: "center",
            }}
          >
            <Text style={{ textAlign: "center", fontSize: 30 }}>{index}</Text>
          </View>
        )}
      />

      <Pagination.Basic
        progress={progress}
        data={data}
        dotStyle={{ backgroundColor: "rgba(0,0,0,0.2)", borderRadius: 50 }}
        containerStyle={{ gap: 5, marginTop: 10 }}
        onPress={onPressPagination}
      />
    </View>
  );
}

export default App;
```

#106. Bài Tập Hoàn Thiện Banner
//todo

Download hình ảnh tại đây

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

#107. Bài Tập Hoàn Thiện List Category
//todo
Download hình ảnh tại đây

```
const data1 = [
   { key: 1, name: "Hot Deal", source: require("@/assets/icons/flash-deals.png") },
   { key: 2, name: "Quán Ngon", source: require("@/assets/icons/nice-shop.png") },
   { key: 3, name: "Tích Điểm", source: require("@/assets/icons/points.png") },
   { key: 4, name: "Ngọt Xỉu", source: require("@/assets/icons/rice.png") },
   { key: 5, name: "Quán Tiền Bối", source: require("@/assets/icons/noodles.png") },
   { key: 6, name: "Bún, Mì, Phở", source: require("@/assets/icons/bun-pho.png") },
   { key: 7, name: "BBQ", source: require("@/assets/icons/bbq.png") },
   { key: 8, name: "Fast Food", source: require("@/assets/icons/fastfood.png") },
   { key: 9, name: "Pizza", source: require("@/assets/icons/Pizza.png") },
   { key: 10, name: "Burger", source: require("@/assets/icons/burger.png") },
   { key: 11, name: "Sống Khỏe", source: require("@/assets/icons/egg-cucmber.png") },
   { key: 12, name: "Giảm 50k", source: require("@/assets/icons/moi-moi.png") },
   { key: 13, name: "99k Off", source: require("@/assets/icons/fried-chicken.png") },
   { key: 14, name: "No Bụng", source: require("@/assets/icons/korean-food.png") },
   { key: 15, name: "Freeship", source: require("@/assets/icons/Steak.png") },
   { key: 16, name: "Deal 0Đ", source: require("@/assets/icons/tomato.png") },
   { key: 17, name: "Món 1Đ", source: require("@/assets/icons/elipse.png") },
   { key: 18, name: "Ăn chiều", source: require("@/assets/icons/chowmein.png") },
   { key: 19, name: "Combo 199k", source: require("@/assets/icons/Notif.png") },
   { key: 20, name: "Milk Tea", source: require("@/assets/icons/salad.png") },
]
```

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

#108. Bài Tập Hoàn Thiện Header
//bonus: chỉnh background
https://github.com/react-navigation/react-navigation/issues/9811#issuecomment-901724207

https://github.com/expo/expo/blob/main/templates/expo-template-tabs/app/_layout.tsx#L52C1-L53C1

Về source code video này:
Các file thay đổi tại video, xem tại đây
Source code ứng với video này, xem tai đây
Video hướng dẫn cách download source sau khi kết thúc video này, xem tại đây

#109. Tạo Base Category Nổi Bật
//todo
```
const data = [
   { key: 1, name: "Top Quán Rating 5* tuần này", ref: "" },
   { key: 2, name: "Quán Mới Lên Sàn", ref: "" },
   { key: 3, name: "Ăn Thỏa Thích, Freeship 0Đ", ref: "" },
]
```

Về source code video này:
Các file thay đổi tại video, xem tại đây
Source code ứng với video này, xem tai đây
Video hướng dẫn cách download source sau khi kết thúc video này, xem tại đây

#110. Bài Tập Design Bottom Tabs
https://reactnavigation.org/docs/tab-based-navigation#customizing-the-appearance

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

#111. Tổng Kết về chapter
//todo

Chapter 11: Module Restaurant
Chia sẻ data giữa các component và xem chi tiết cửa hàng/quán ăn

#112. Tổng quan về chapter

- Customize splash screen/app icon

- Lưu thông tin người dùng sau khi đã đăng nhập:
    + Tab account
    + Tắt app mở lại vẫn còn thông tin và không cần đăng nhập lại

- Hiển thị danh sách collections

- Xem chi tiết cửa hàng:
    + Sử dụng Section List kết hợp với Flatlist

    + Sử dụng animation (sticky header)

    + Sử dụng skeleton

#113. Sử dụng React Context

1.Vấn đề tồn đọng

Cần chia sẻ data giữa các component (mà không dùng passing props từ cha sang con)

Ví dụ: login thành công, làm sao để hiển thị thông tin user tại components

Ví dụ mobile có tồn tại "localStorage" tương tự khi làm website, là**m sao để data thay đổi, component "re-render" ?**

2.Setup React Context
Lưu ý: cú pháp này áp dụng với React 18 trở xuống, có thể cú pháp của React 19 sẽ hơi khác một xíu

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 tai đây
Video hướng dẫn cách download source sau khi kết thúc video này, xem tại đây

#114. Hiển thị thông tin user login

Để bắt đầu video này, đừng quên chạy dự án backend bạn nhé

Hiển thị thông tin user đăng nhập tại tab account

File design giao diện, tham khảo tại đây

Hiển thị avatar:
http://localhost:8080/images/avatar/default-user.png

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

#115. Access Token & Async Storage

1.Làm sao để người login duy nhất chỉ 1 lần ?

Lưu ý: React Native là frontend, không nên lưu trữ thông tin của người dùng tại frontend, và các hệ thống thực tế cũng không làm vậy. Cần có backend để xử lý vấn đề trên

Logic hiện tại:

- User sử dụng username/password để đăng nhập.

- React Native gọi tới backend để xác thực thông tin người dùng.
  Nếu thông tin hợp lệ, backend trả ra thông tin. React Native lưu thông tin này vào React Context (memory)

- Mỗi lần tắt ứng dụng, memory được giải phóng (tương tự F5/refresh website), thông tin lưu trữ của React Context bị mất => mất thông tin của user

Giải pháp đề ra: cần phải có cơ chế gì đấy mà không yêu cầu người dùng nhập username/password
=> sử dụng access token

2. Access Token
Về cách tạo ra access token, bạn vui lòng tham khảo backend tại đây

Access Token dùng để "định danh" người dùng là ai (mà không cung cấp username/password)

Mỗi 1 request từ frontend gửi lên backend, backend sẽ "giải mã" token đấy để biết được rằng ai đang sử dụng nó.

Để giải quyết vấn đề user chỉ login 1 lần thì:

- Sau khi login thành công, backend sẽ trả ra access token cho frontend. Frontend sẽ cần lưu trữ lại thông tin này

- Khi cần biết ai là người dùng đang đăng nhập, frontend sẽ cần gọi tới backend, nhờ "giải mã" thông tin chứa trong token.

3. Lưu vào async storage

Với React Native, sử dụng Async Storage (tương tự LocalStorage khi sử dụng website)

Cài đặt thư viện
https://docs.expo.dev/versions/latest/sdk/async-storage/
https://www.npmjs.com/package/@react-native-async-storage/async-storage

- Áp dụng khi sử dụng SDK 51:
npm i --save-exact @react-native-async-storage/async-storage@1.23.1

https://react-native-async-storage.github.io/async-storage/docs/usage

//check async storage

```
const printAsyncStorage = () => {
  AsyncStorage.getAllKeys((err, keys) => {
    AsyncStorage.multiGet(keys!, (error, stores) => {
      let asyncStorage: any = {}
      stores?.map((result, i, store) => {
        asyncStorage[store[i][0]] = store[i][1]
      });
      console.log(JSON.stringify(asyncStorage, null, 2));
    });
  });
};
```

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

#116. API Get Account

Để bắt đầu video này, đừng quên chạy dự án backend bạn nhé.

Cần chạy backend để thực hiện tất cả các video còn lại của khóa học.

Mục tiêu: tắt app, mở lại ứng dụng, user (có access token), sẽ không cần đăng nhập lại

Bước 1: gán access token vào header request

Lấy access token (lưu tại async storage)
https://react-native-async-storage.github.io/async-storage/docs/api#getitem

Gán vào header axios

Bước 2: gọi API Get Account
//nơi nào dùng để gọi API ?

Bước 3: cập nhật React Context
//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

#117. Sử Dụng Splash Screen

Để bắt đầu video này, đừng quên chạy dự án backend bạn nhé

Tài liệu:
https://docs.expo.dev/versions/latest/sdk/splash-screen/

Bước 1: Cấu trúc lại rootPage

Tách code hiện tại, thành màn hình: (auth)/welcome.tsx

Logic khi vào rootPage sẽ là: nếu người dùng đã đăng nhập, redirect về trang chủ. Ngược lại, redirect về màn hình login

Bước 2: Splash Screen

Cài đặt thư viện:
- Áp dụng khi sử dụng SDK 51:
npm i --save-exact expo-splash-screen@0.27.5

Splash Screen là logic hiển thị "màn hình chờ" trước khi người dùng sử dụng ứng dụng, ví dụ fetch API, fetch fonts...

=> sử dụng rootPage cho việc trên

//clear cache:
https://stackoverflow.com/a/51998712

"start": "expo start -c",

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

#118. Custom Splash Screen & App Icon

Tài liệu: https://docs.expo.dev/develop/user-interface/splash-screen-and-app-icon/

Download file hình ảnh tại đây

Splash-screen: hiển thị khi chờ app loading data (fonts, api…)

App-icons: hiển thị trên app store và ngoài màn hình (nếu user download về)

//cần clear cache để nhận icons mới
//clear cache:
https://stackoverflow.com/a/51998712

"start": "expo start -c",

//giải pháp đề ra: rename icons là xong

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

#119. Design Collections

Download file hình ảnh tại đây

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

#120. API Hiển Thị Collections

Để bắt đầu video này, đừng quên chạy dự án backend bạn nhé

//todo
Nguyên tắc hiển thị hình ảnh (đang làm tại backend)
url-backend/images/tên-loại-sử-dụng/tên-image

Hiển thị restaurant:
http://localhost:8080/images/restaurant/**restaurant_name**
//todo

//lưu ý: cần tạo import data cho collections
Download file collections tại đây

https://stackoverflow.com/questions/55487695/react-native-text-component-using-number-of-lines

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

#121. Giao Diện Xem Chi Tiết Restaurant

Cài đặt thư viện:
npm i --save-exact @faker-js/faker@9.0.1 debounce@2.1.1
react-native-tabs-section-list@1.0.1

Tải source code để thực hành tại đây

//Giải thích về cấu trúc của component

- Header + image
- Info
- Slide menu
- Section list

//tổng quan về animated

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

#122. Section List (Part 1)

//zIndex và position absolute (về nút back và heart)
//file section.list.basic.tsx

//sử dụng SectionList

//useRef hook

//check viewableItems

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

#123. Section List (Part 2)

//file section.list.scroll.tsx

//sử dụng SectionList với Flatlist

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

#124. Section List (Part 3)

//file section.list.library.tsx

Tham khảo:
https://github.com/facebook/react-native/issues/29547

//sử dụng thư viện
https://github.com/netguru/sticky-parallax-header

https://github.com/bogoslavskiy/react-native-tabs-section-list

//đọc source code của thư viện

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

#125. API Xem Chi Tiết Restaurant (Part 1)
Tài liệu:
https://docs.expo.dev/router/create-pages/#dynamic-routes

//sử dụng dynamic routes

//gọi api lấy thông tin của restaurant

//hiển thị tên/image của restaurant

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

#126. API Xem Chi Tiết Restaurant (Part 2)

```
//process data
 interface IMenu {
      _id: string;
      restaurant: string;
      title: string;
      createdAt: Date;
      updatedAt: Date;
      menuItem: IMenuItem[]
   }

   interface IMenuItem {
      _id: string;
      menu: string;
      title: string;
      description: string;
      basePrice: number,
      image: string;
      options: {
         title: string;
         description: string;
         additionalPrice: number;
      }[],
      createdAt: Date;
      updatedAt: Date;
   }




export const processDataRestaurantMenu = (restaurant: IRestaurant | null) => {
   if (!restaurant) return [];
   return restaurant?.menu?.map((menu, index) => {
      return {
         index,
         key: menu._id,
         title: menu.title,
         data: menu.menuItem
      }
   })
}
```

//hiển thị menu và items
Test với id = **66d5756d7d50bae39992e916**

//check type của sectionList
https://stackoverflow.com/a/65464102

//hoàn thiện giao diện
Icons sử dụng:
https://icons.expo.fyi/Index/AntDesign/plussquare

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

#127. Sử Dụng Skeleton (Extra)

1.Fix bug giao diện
Format giá tiền
https://stackoverflow.com/a/66343347

```
export const currencyFormatter = (value: any) => {
   const options = {
      significantDigits: 2,
      thousandsSeparator: '.',
      decimalSeparator: ',',
      symbol: 'đ'
   }

   if (typeof value !== 'number') value = 0.0
   value = value.toFixed(options.significantDigits)

   const [currency, decimal] = value.split('.')
   return `${currency.replace(
      /\B(?=(\d{3})+(?!\d))/g,
      options.thousandsSeparator
   )}${options.symbol}`
}
```

2. Sử dụng Skeleton
//hướng dẫn cách search với Expo

//File ảnh skeleton của Shopee, tải tại đây

//Tìm hiểu nhanh về SVG, tham khảo nhanh tại đây:
https://www.w3schools.com/graphics/svg_intro.asp

Với máy ảo. Nhấn phím Ctrl + S để capture nhanh màn hình

Cài đặt thư viện:
https://github.com/danilowoz/react-content-loader
- Áp dụng khi sử dụng SDK 51:
npm i --save-exact react-content-loader@7.0.2 react-native-svg@15.7.1

```
//tạo fake delay

   return axios.get<IBackendRes<IRestaurant>>(url, {
      headers: { delay: 3000 }
   });



//phần collection

// const { height: sHeight, width: sWidth } = Dimensions.get('window');
<ContentLoader
   speed={2}
   width={sWidth}
   height={230}
   // viewBox="0 0 700 150"
   backgroundColor="#f3f3f3"
   foregroundColor="#ecebeb"
   style={{ width: '100%' }}
>
   <Rect x="10" y="10" rx="5" ry="5" width={150} height="200" />
   <Rect x="170" y="10" rx="5" ry="5" width={150} height="200" />
   <Rect x="330" y="10" rx="5" ry="5" width={150} height="200" />
</ContentLoader>
```

Về source code video này:
Các file thay đổi tại video, xem tại đây
Source code ứng với video này, xem tai đây
Video hướng dẫn cách download source sau khi kết thúc video này, xem tai đây

```
// const { height: sHeight, width: sWidth } = Dimensions.get('window');
<ContentLoader
   speed={2}
   width={700}
   height={sHeight}
   // viewBox="0 0 700 150"
   backgroundColor="#f3f3f3"
   foregroundColor="#ecebeb"
   style={{ width: '100%' }}
>
   <Rect x="0" y="0" rx="3" ry="3" width={sWidth} height="120" />

   <Rect x="10" y="140" rx="10" ry="10" width={sWidth - 50} height="20" />
   <Rect x="10" y="170" rx="10" ry="10" width={sWidth - 150} height="20" />

   <Rect x="10" y="220" rx="5" ry="5" width={100} height="100" />
   <Rect x="130" y="220" rx="10" ry="10" width={150} height="20" />
   <Rect x="130" y="250" rx="10" ry="10" width={100} height="20" />
   <Rect x="130" y="280" rx="10" ry="10" width={200} height="20" />

   <Rect x="10" y="340" rx="5" ry="5" width={100} height="100" />
   <Rect x="130" y="340" rx="10" ry="10" width={150} height="20" />
   <Rect x="130" y="370" rx="10" ry="10" width={100} height="20" />
   <Rect x="130" y="400" rx="10" ry="10" width={200} height="20" />

   <Rect x="10" y="460" rx="5" ry="5" width={100} height="100" />
   <Rect x="130" y="460" rx="10" ry="10" width={150} height="20" />
   <Rect x="130" y="490" rx="10" ry="10" width={100} height="20" />
   <Rect x="130" y="520" rx="10" ry="10" width={200} height="20" />

   <Rect x="10" y="580" rx="5" ry="5" width={100} height="100" />
   <Rect x="130" y="580" rx="10" ry="10" width={150} height="20" />
   <Rect x="130" y="610" rx="10" ry="10" width={100} height="20" />
   <Rect x="130" y="640" rx="10" ry="10" width={200} height="20" />

   <Rect x="10" y="700" rx="5" ry="5" width={100} height="100" />
   <Rect x="130" y="700" rx="10" ry="10" width={150} height="20" />
   <Rect x="130" y="730" rx="10" ry="10" width={100} height="20" />
   <Rect x="130" y="760" rx="10" ry="10" width={200} height="20" />

</ContentLoader>
```

#128. Tổng kết về chapter

//hướng dẫn thêm mới data cho restaurant

Truy cập: https://shopeefood.vn/

Chọn quán ăn

Thêm thông tin hình ảnh và text/price

//todo

Chapter 12: Module Product

Design giao diện order sản phẩm của nhà hàng/quán ăn

#129. Tổng quan về chapter

Các tính năng chính:

- Tính năng đặt hàng:
    + Thêm/sửa sản phẩm
    + Sử dụng Modal
- Tính năng order history

#130. Error Boundary (Extra)
Tài liệu: https://docs.expo.dev/router/error-handling/
Mục tiêu:
//handle no internet connection.
Check trường hợp chưa chạy backend

Lưu ý: đây là xử lý lỗi (nếu có), giúp ứng dụng không bị crash.

Việc check người dùng "có kết nối mạng hay không" (network connections), tham khảo: https://docs.expo.dev/versions/latest/sdk/network/

1.Về Error Boundary
https://react.dev/reference/react/Component#catching-rendering-errors-with-an-error-boundary

https://legacy.reactjs.org/docs/error-boundaries.html#how-about-event-handlers

2. Sử dụng với Expo
https://github.com/facebook/react/issues/14981#issuecomment-468460187

```
<SafeAreaView style={{ flex: 1 }}>
  <View style={{ flex: 1, paddingHorizontal: 10, gap: 15 }}>
    <View style={{
      backgroundColor: "#333", padding: 10,
      borderRadius: 3, gap: 10
    }}>
      <Text style={{ color: "red", fontSize: 20 }}>
        Something went wrong
      </Text>
      <Text style={{ color: "#fff" }}>{error.message}</Text>
    </View>
    <Button title="Try Again ?" onPress={retry} />
  </View>
</SafeAreaView>
```

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

#131. Phân tích tính năng mua hàng

Yêu cầu: bạn cần có tài khoản và đăng nhập vào Shopee Food
Chưa làm các animation liên quan về giỏ hàng trong khóa học này

1. Quy trình thực hiện

Bạn chọn quán ăn => chọn món ăn

Nếu món ăn không có options (ví dụ size, color…) => tăng số lượng là 1, hiển thị giỏ hàng tại bottom

Nếu món ăn có options, hiển thị modal các option => tăng số lượng 1, hiển thị giỏ hàng tại bottom

Mỗi 1 cửa hàng là một giỏ hàng riêng (tính theo đơn vận chuyển)

2. Logic xử lý

Lưu thông tin giỏ hàng vào React Context (để chia sẻ giữa các màn hình khác nhau)

Lưu dưới dạng object, vì 1 giỏ hàng, chứa nhiều đơn hàng (ứng với các cửa hàng khác nhau)

Carts : {
“Restaurant-1-id”: { … },
“Restaurant-2-id”: { … },
}

#132. Bài Tập Design Giỏ Hàng/Tăng Giảm Số Lượng

Mục tiêu:
Design giao diện Footer và số lượng giỏ hàng (nút tăng/giảm)

Expo Icons: https://icons.expo.fyi/Index

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

#133. Data Giỏ Hàng (Part 1)
Lưu vào React Context

Cấu trúc data giỏ hàng : lưu theo id của hàng và id của sản phẩm

//tăng, giảm, hiển thị số lượng từng sản phẩm

```
interface ICart {
    [key: string]: {
        sum: number;
        quantity: number;
        items: {
            [key: string]: {
                quantity: number;
                data: IMenuItem
            }
        }
    }
}
```

cart: ICart | Record<string, never>;

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

#134. Data Giỏ Hàng (Part 2)

//todo
//tăng giảm số lượng item
//hiển thị tổng số tiền tại footer

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

#135. Sử Dụng Modal

Tài liệu: https://docs.expo.dev/router/advanced/modals/

//fix lỗi flicker giao diện
Set background => “white” thay vì “transparent”

//sử dụng animation

```
<Animated.View
  entering={FadeIn}
  style={{
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#00000040',
  }}
>
  {/* Dismiss modal when pressing outside */}
  <Link href={'/'} asChild>
    <Pressable style={StyleSheet.absoluteFill} />
  </Link>
  <Animated.View
    entering={SlideInDown}
    style={{
      width: '90%',
      height: '80%',
      alignItems: 'center',
      justifyContent: 'center',
      backgroundColor: 'white',
    }}
  >
    <Text style={{ fontWeight: 'bold', marginBottom: 10 }}>Modal Screen</Text>
    <Link href="/">
      <Text>← Go back</Text>
    </Link>
  </Animated.View>
</Animated.View>
```

Về source code video này:
Các file thay đổi tại video, xem tại đây
Source code ứng với video này, xem tai đây
Video hướng dẫn cách download source sau khi kết thúc video này, xem tại đây

#136. Phân Tích Modal Options
//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

#137. Design Modal Options Create
//todo
Source code để bắt đầu video này, tải tại đây

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

#138. Hoàn Thiện Modal Options Create
//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

#139. Modal Options Update
//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

#140. Bài Tập Order 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

#141. Place Orders API

//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

#142. Bài Tập Hiển Thị Đơn Hàng đã đặt
//todo

```
interface IOrderHistory {
      _id: string;
      restaurant: IRestaurant;
      user: string;
      status: string;
      totalPrice: number;
      totalQuantity: number;
      orderTime: Date;
      detail: {
         image: string;
         title: string;
         option: string;
         price: number;
         quantity: number;
      }[]
      createdAt: Date;
      updatedAt: Date;
   }
```

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

#143. Tổng kết về chapter
//todo

Chapter 13: Luyện Tập
Thực hành kiến thức thông qua các bài tập về React Native để hoàn thiện dự án

#144. Tổng quan về chapter
//todo

#145. Bài Tập Design Account Screen

- Cập nhật thông tin
- Thay đổi mật khẩu
- Ngôn ngữ
- Về ứng dụng
  Log out

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

#146. Code Refactoring
//todo

Về source code video này:
Các file thay đổi tại video, xem tại đây
Source code ứng với video này, xem tai đây
Video hướng dẫn cách download source sau khi kết thúc video này, xem tai đây

#147. Chức Năng LogOut

1. Cơ chế khi sử dụng access token

Access Token dùng để định danh người dùng, và “thuộc về client” sử dụng.

Logout có bị hết hạn không ?

Logout không bị hết hạn. Nếu người dùng vẫn giữ access token, vẫn sử dụng được.

Giải pháp đề ra, là set thời hạn sử dụng ngắn xuống. Ví dụ 30 phút, 1h, 1 ngày.

Khi hết hạn, sử dụng refresh token để lấy lại access token

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

#148. Keyboard Avoiding View (Extra)

Tài liệu: https://docs.expo.dev/guides/keyboard-handling/

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

#149. Bài Tập Cập Nhật Thông Tin User

Update validate với formik:
https://formik.org/docs/guides/validation#when-does-validation-run

Disabled button submit: https://stackoverflow.com/a/59676720

//API update user, sử dụng: **PATCH /api/v1/users**

//update xong, cần cập nhật react 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

#150. Bài Tập Change Password

API change password, sử dụng: **POST /api/v1/users/password**

Validate password match với yup: https://stackoverflow.com/a/72975771

Reset formik with ref: https://stackoverflow.com/a/66301797

Khai báo type cho ref:
https://github.com/jaredpalmer/formik/issues/2290#issuecomment-852946205

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

#151. Bài Tập Forgot Password

Bước 1 : tạo screen **request.password**

Yêu cầu: điền vào email và gọi API để gửi code xác nhận qua email

API sử dụng: **POST /api/v1/auth/retry-password**

API gọi thành công, chuyển qua screen tại bước 2

Truyền thêm thông tin email khi navigate/replace (mục đích là sử dụng tại bước 2)

Bước 2: tạo screen thực hiện thay đổi mật khẩu: **forgot.password**

//nếu làm resend-code (tương tự màn hình kích hoạt tài khoản)
API sử dụng để resend email chính là API tại bước 1 **POST /api/v1/auth/retry-password**

API sử dụng để thực hiện thay đổi mật khẩu: **POST /api/v1/auth/forgot-password**

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

#152. Bài Tập Like/Dislike a Restaurant

1.Like/Dislike a restaurant
//component: sticky.header.tsx
Api fetch restaurant by id, trả thêm thuộc tính: isLike: boolean

Sử dụng API : **POST /api/v1/likes**

2.Hiển thị danh sách cửa hàng đã thích

Sử dụng API: **GET /api/v1/likes?current=1&pageSize=10**

//chấp nhận lỗi, data không cập nhật (do component đã mount)

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

#153. Bài Tập Pull to Refresh

Tài liệu:
https://reactnative.dev/docs/refreshcontrol

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

#154. Sử Dụng Custom Font (Extra)

Tài liệu:
https://docs.expo.dev/versions/latest/sdk/font/

https://docs.expo.dev/develop/user-interface/fonts/

https://fonts.google.com/
Download fonts sử dụng trong video tại đây

React Native sử dụng font mặc định của android/ios
https://stackoverflow.com/a/57746853

Cài đặt thư viện:
- Áp dụng khi sử dụng SDK 51:
npm i --save-exact expo-font@12.0.10

//app.json
```
  "plugins": [
    "expo-router",
    "expo-font"
  ],
```

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

#155. Bài Tập Thanh Search Header

Logic xử lý:

1.Tại homepage, nhấn vào thanh search, điều hướng qua màn hình tìm kiếm
=> cần tạo thêm (auth)/search.tsx

2. Design layout theo ý tưởng sau
Bố cục gồm 2 phần, phần header để search, và phần body hiển thị kết quả

Check điều kiện từ ô input:

- Nếu input rỗng, hiển thị kết quả gợi ý (mặc định) (sử dụng flatlist, hardcode data)

- Nếu input khác rỗng, gọi API để hiển thị kết quả
  Ở đây, làm basic, là tìm kiếm cửa hàng (shopee food là tìm kiếm theo tên món ăn)

  Nếu click vào kết quả, chuyển hướng tới cửa hàng

API sử dụng: (lưu ý cần test api để biết cách dùng và kết quả trả về)
/api/v1/restaurants?current=1&pageSize=10&name=/**keyword**/i

Ví dụ: /api/v1/restaurants?current=1&pageSize=10&name=/sữa/i
Sẽ hiển thị các cửa hàng có tên, chứa từ “sữa”

Có thể dùng debounce để hạn chế việc gọi API quá nhiều khi onChange
https://dev.to/otamnitram/throttling-and-debouncing-avoiding-unnecessary-api-calls-2god

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

#156. Bài Tập Hiển Thị Tất Cả Restaurants

//sử dụng load more
Tài liệu: https://reactnative.dev/docs/0.74/flatlist#virtualizedlist-props

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

#157. Bài Tập Hiển Thị Popup Sale
//todo
Vào trang chủ, hiển thị popup

Download file hình ảnh sử dụng trong video tại đây

//lỗi useEffect không navigate:
https://github.com/react-navigation/react-navigation/issues/10803

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

#158. Tổng kết về chapter
//todo

Chapter 14: Tổng kết
Tổng kết các kiến thức đã học và hướng phát triển tiếp theo cho dự án

#159. Nhận Xét Về Dự Án Thực Hành

Ưu điểm:

- Nắm vững các component quan trọng/cốt lõi của React Native:
  View, Text, Button, Image
  TextInput, Pressable
  SectionList, FlatList, SectionList

- Thực hành để nắm vững các kiến thức trên thông qua dự án với tính thực tiễn cao.

Nhược điểm:

- Các component chưa đề cập (sẽ làm trong khóa React Native Advance):
  Xử lý Camera
  Xử lý Location
  Xử lý Notification

//chưa làm login với google/github vì yêu cầu cần phải build app

#160. Cách Upgrade ứng dụng Expo (SDK 51)
Tài liệu: https://docs.expo.dev/workflow/upgrading-expo-sdk-walkthrough/
Lưu ý 1: framework expo (trung bình 1 năm release 1 version của SDK)
Lưu ý 2: Khóa học này sử dụng SDK 51, sau này ra version mới hơn (52, 53…) sẽ được mình làm thêm video hướng dẫn upgrade sau.

Khi upgrade, cần nâng cấp từng version một, tránh breaking change.

Ví dụ: SDK của bạn v49, trong khi version mới nhất là v52. Bạn sẽ làm:
Update v49 -> v50. Kiểm tra xem có ok không ?
Update v50 -> v51. Kiểm tra xem có ok không ?
Update v51 -> v52. Kiểm tra xem có ok không ?

Việc update trực tiếp từ v49 -> v52 sẽ có rủi ro. Vì vậy hạn chế tối đa việc làm này, chỉ nên upgrade từng version một

1.Hướng dẫn cách upgrade thư viện
Đọc các thay đổi của SDK khi có version mới (hoặc so sánh giữa các version) tại đây:
https://docs.expo.dev/workflow/upgrading-expo-sdk-walkthrough/#sdk-changelogs

Bước 1: Sử dụng GIT để hạn chế tối đa rủi ro
Bạn nên checkout sang 1 branch code khác cho việc upgrade. Vì nếu có lỗi, sẽ không ảnh hưởng tới code hiện tại.

Bước 2: Cần xác định version SDK bạn cần nâng cấp
Lưu ý: nâng cấp từng version một
Tại thời điểm mình quay video này, chưa có SDK v52, vì vậy, version mới nhất là v51
Câu lệnh sử dụng: **npm install expo@51**

Bước 3: upgrade các thư viện phụ thuộc vào expo
npx expo install --fix

Bước 4: cài đặt thư viện với câu lệnh **npm i**

Bước 5: Test dự án để xem kết quả upgrade có gây ra bug hay không ?

Lưu ý: với các thư viện không phụ thuộc vào Expo, bạn cần update thủ công (check version rồi cài đặt)

#161. Cách Tự Code Project React Native của bạn

Lý do mình yêu cầu kéo code từ github, là để đảm bảo môi trường của bạn và video thực hành là giống hệt nhau, hạn chế tối đa bug có thể xảy ra.

Khi bạn tự thực hành, bug đâu thì fix đó, vì fix bug giúp bạn lên trình, giống quá trình mình quay video bị bug đấy.
Bug => fix => upgrade level

Điểm khác biệt giữa việc bây giờ bạn thực hành project (video #161), và việc ở những video đầu tiên bạn kéo code từ git của mình (video #22),
là bây giờ bạn đã có kiến thức nền tảng về React Native rồi.

Bạn cũng đã có kỹ năng search google, debug cũng như keyword để search (nếu có bug xảy ra)

Code mà không có bug mới là lạ.ahihi

Cách để tự code (từ đầu) dự án React Native của chính bạn

Bước 1: Tạo nhanh dự án expo với template sử dụng **expo router (typescript)**
https://docs.expo.dev/more/create-expo/

npx create-expo-app@latest --template default

Bước 2: Cài đặt các thư hiện của expo thông qua npx expo
Expo sẽ tự động tìm version phù hợp với SDK bạn sử dụng

Ví dụ: tại video #154, sử dụng: **npm i --save-exact expo-font@12.0.10**
Thì sử dụng với expo:
npx expo install expo-font

Bước 3: Về design giao diện
Tham khảo các file design giao diện trên figma (hoặc tự design giao diện)

Bước 4: Test ứng dụng React Native
Sử dụng Expo Go, hoặc build ra dự án react native, hoặc kết nối debug thông qua chính điện thoại của bạn

Bước 5: Thực hành dự án với backend

Mình khuyến khích các bạn nên học backend để có thể viết API theo ý thích
Tham khảo lộ trình backend của hoidanit tại đây

Nếu bạn muốn có API miễn phí, hoặc API fake, có thể tham khảo:
https://github.com/typicode/json-server
https://jsonplaceholder.typicode.com/

Lưu ý: hàng miễn phí, sẽ không được “ngon như ý bạn”, vì bạn “không kiểm soát nó”.

Vì vậy, để làm được các ứng dụng ngon (ví dụ như khóa học mình hướng dẫn), cần phải học backend các bạn nhé.

#162. Prebuild Dự Án Expo (Part 1)
Tài liệu: https://docs.expo.dev/workflow/prebuild/

Mục tiêu của video này là giúp bạn hiểu, tại sao cần Prebuild (khi đã có Expo Go)

1.Tại sao lại gọi là Prebuild (không phải là build), và tại sao cần (khi đã có Expo Go)

Prebuild là quá trình tạo ra native code. Cụ thể hơn, từ code dự án react native, chuyển thành code kotlin (android) và swift (ios)

Tương tự như việc bạn sử dụng React Native CLI, prebuild của Expo, cũng sẽ **tạo ra 2 folder là android và ios.**

Sao không gọi prebuild là build ?
Từ build thường ám chỉ sản phẩm cuối cùng (production) (ví dụ với android là file APK, ios là IPA)
Prebuild của Expo tạo ra folder android/ios => chỉ được gọi là “pre” (quá trình sơ khai)

Tại sao cần Prebuild khi đã có Expo Go ?

Expo Go là ứng dụng tạo ra môi trường sandbox giúp bạn test nhanh ứng dụng Android/IOS mà không cần cài đặt thêm gì khác (ví dụ như cấu hình android/ios)

Ưu điểm: tạo nhanh dự án (tiết kiệm tối đa thời gian cho việc cấu hình)

Nhược điểm: chỉ có thể tích hợp các package có sẵn trong SDK
https://docs.expo.dev/develop/development-builds/introduction/

Hướng dẫn sử dụng search của react native để biết thư viện hỗ trợ expo go hay không ?
https://reactnative.directory/

Ví dụ như không chạy pre-build, không test được google login.
https://docs.expo.dev/guides/google-authentication/

#163. Prebuild Dự Án Expo (Part 2)

Mục tiêu của video này, là thực hiện quá trình prebuild ứng dụng Expo

Lưu ý: hướng dẫn của video này sử dụng Android Studio (mục tiêu build file APK)
Trên windows không thể cấu hình và build ứng dụng của iOS

1. Cài đặt expo dev client
https://docs.expo.dev/versions/latest/sdk/dev-client/

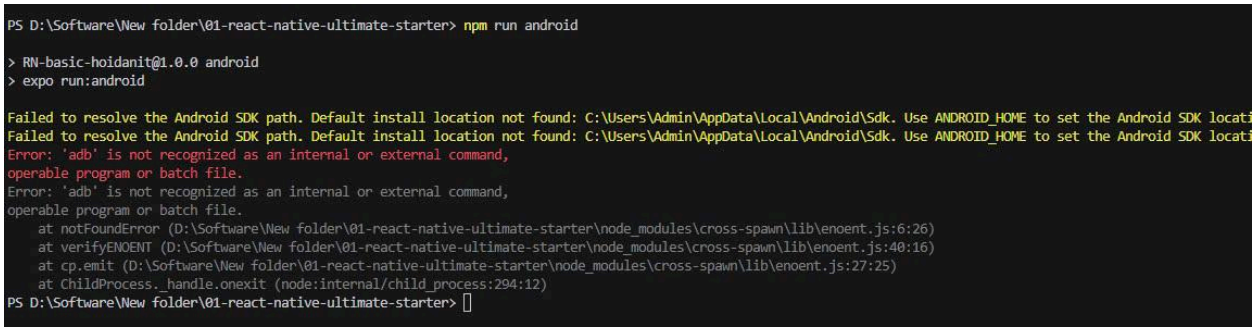
Sử dụng câu lệnh: **npx expo install expo-dev-client expo-system-ui**

Cài dev client để hỗ trợ tốt hơn trong quá trình debug ứng dụng Expo (với chế độ prebuild)

2. Setup môi trường Android (đã làm tại video #16.2)
Tham khảo: (sử dụng VPN để xem)
https://medium.com/@prasadkatkade008/how-to-set-up-expo-dev-client-a-complete-guide-2024-be898a519ec1

Nếu bạn không cấu hình step 2 này, sẽ gặp lỗi như sau:
Failed to resolve the Android SDK path. Use ANDROID_HOME to set the Android SDK location.

Error: 'adb' is not recognized as an internal or external command,
operable program or batch file.



3. Chạy dự án với chế độ prebuild

Bước 1: build native code với câu lệnh prebuild
npx expo prebuild

Output của câu lệnh trên, là trong source code của bạn, sẽ có thêm folder android/ios (native code)
(trên windows chỉ tạo ra folder android)

Lưu ý: thêm folder android/ios vào .gitignore

Bước 2: chạy dự án thực hành
//check câu lệnh tại file package.json

Bản chất của quá trình này, là tạo ra 1 version “giống hệt” như Expo Go, bạn vẫn có hot reloading, debug…

Bước 3: khi nào bạn cần chạy lại câu lệnh pre-build ?

Mỗi lần bạn cài đặt thêm thư viện mới, nên chạy lại bước 2.. Phần lớn thời gian còn lại, là sử dụng đúng bước 3 để chạy dự án.

#164. Build File APK cho ứng dụng

Lưu ý 1 : chi tiết về build Android/IOS sẽ được mình hướng dẫn tại khóa học tiếp theo

Lưu ý 2: bạn nên xem hết video để có một bức tranh tổng quan, sau đấy hãy thực hành, như vậy sẽ tiết kiệm được thời gian (vì mỗi lần build là cần chờ code thực thi)

Có 2 các chính để build ứng dụng Expo

Cách 1: **build local**, sử dụng chính máy tính của bạn (là cách làm trong video này)
Ưu điểm: miễn phí 100%
Nhược điểm: bạn cần tốn thời gian cấu hình, fix bug (nếu có)

Nơi chứa file apk build:
android\app\build\outputs\apk\release

Cách 2: sử dụng dịch vụ của Expo, hỗ trợ build android/ios out of the box
https://docs.expo.dev/build/introduction/

Ưu điểm: tối ưu hóa cho việc build android/ios
Nhược điểm: pay as you go (có hỗ trợ gói dùng để demo - free)

Nhận xét về bundle size (build size)
https://reactnative.dev/showcase

Quá trình build thực hiện trong video:
Bước 1: chọn nơi lưu thư mục code
Với đường link quá dài, hoặc chứa ký tự đặc biệt (ví dụ đặt tiếng việt) sẽ build failed.

Bạn clone/copy source code (không copy thư mục node_modules và android) sang nơi khác, đảm bảo là nơi chứa có tên ngắn gọn và không chứa ký tự đặc biệt

Bước 2: update file .env
Cần update đường link backend tại file .env, ở đây sử dụng forward port, chi tiết mình hướng dẫn tại #168

Bước 3: Prebuild dự án

Chạy lần lượt 2 câu lệnh sau:

npm i

npx expo prebuild

Bước 4: Mở dự án bằng Android Studio

Mở thư mục **android** được tạo ra từ bước 3

Tiến hành build file apk

Nơi chứa file apk build:
android\app\build\outputs\apk\release

#165. What's next ?

1.Tự thực hành dự án React Native của bạn
Mục đích: để chuyển hóa (hiểu sâu hơn) về các kiến thức đã học

Cách 1: Nếu bạn có khả năng tự viết backend của bạn
Clone các ứng dụng (app) tương tự cách mình làm trong khóa học này

Cách 2: Nếu bạn chưa viết được backend

Có thể code lại dự án này, với một số gợi ý thay đổi sau:
Nếu bạn để ý, giao diện Tiki hay shopee, chưa cần đăng nhập vẫn xem được sản phẩm, chỉ khi nào mua sản phẩm, mới cần đăng nhập

Bạn có thể sửa logic như vậy để làm.

Mình khuyến khích các bạn nên học backend để có thể viết API theo ý thích
Tham khảo lộ trình backend của hoidanit tại đây

2.Học thêm các kiến thức mở rộng

Sử dụng UI với React Native
Về làm UI, tham khảo list thư viện tại đây:
https://docs.expo.dev/ui-programming/user-interface-libraries/

Sử dụng top tab
https://reactnativepro.dev/posts/expo-router-top-tabs

Authentication với Expo (protected route)

Sử dụng các component khác của expo như : camera, location, notification…

Quá trình build react native (prebuild), rồi ESA (build file, deploy store, update version…)

Chapter 15: Cách Test Ứng Dụng React Native (Extra)

Hướng dẫn chi tiết các test ứng dụng React Native phụ thuộc vào cấu hình máy tính và môi trường sử dụng

#166. Tại sao chapter này ra đời ?

Code mobile vất vả hơn so với việc làm lập trình website, vì nó yêu cầu từ phần cứng tới phần mềm.

Ví dụ: code android => sử dụng Android Studio
Code iOS => sử dụng XCode (chỉ có trên MacOS)

Chapter này ra đời, giúp bạn có thể theo dõi và thực hành theo khóa học, không phân biệt bạn sử dụng Android/Macos.

Các vấn đề cần giải quyết (nếu bạn gặp vấn đề nào, theo dõi video đấy để biết cách giải quyết):

Vấn đề 1: #167. Cài đặt Expo Go Chỉ Hỗ Trợ SDK version mới nhất - Làm sao để chạy dự án với SDK version cũ ?

Vấn đề 2: #168. Sử dụng Expo Go trên điện thoại cá nhân, làm sao để kết nối tới backend localhost ?

#167. Cài đặt Expo Go Chỉ Hỗ Trợ SDK version mới nhất
Khi bạn cài Expo Go từ CH Play (Android) và App Store (iOS), mặc định Expo Go chỉ hỗ trợ version mới nhất của Expo SDK.

Vấn đề xảy ra là tại thời điểm mình quay video (mình sử dụng SDK 51), theo thời gian, SDK 52, 53 … ra đời.

Như vậy, bạn sẽ không thể chạy được Expo Go hỗ trợ SDK 51 như video hướng dẫn.

Cách khắc phục, có 2 cách:

Cách 1 (Khuyến khích làm cách này) : tạo custom Expo Go bằng chế độ prebuild

Lưu ý 1: phạm vi áp dụng, là máy tính bạn cần chạy được Android Studio (hoặc XCode), và bạn **không thể quét mã QR** để chạy ứng dụng.

(Chỉ áp dụng cho điện thoại Android) : nếu máy tính bạn "quá yếu", không thể chạy Android Studio (XCode), bạn có thể kết nối điện thoại vào máy tính , và tự google cách bật chế độ debug của điện thoại (như vậy bạn sẽ có một máy ảo mà không cần cài đặt Android Studio)

Lưu ý 2: Cần cấu hình Java và Android path, tham khảo video #16.2

Demo này tương ứng với video #22 (cài đặt dự án thực hành)
Bước 1: cài đặt expo dev client
Sử dụng lần lượt 2 câu lệnh sau:

npm i

npx expo install expo-dev-client expo-system-ui

Bước 2: build native code với câu lệnh prebuild
npx expo prebuild

Output của câu lệnh trên, là trong source code của bạn, sẽ có thêm folder android/ios (native code)
(trên windows chỉ tạo ra folder android)
Lưu ý: thêm folder android/ios vào .gitignore

Bước 3: chạy dự án thực hành
//check câu lệnh tại file package.json
npm run android

Bản chất của quá trình này, là tạo ra 1 version "giống hệt" như Expo Go, bạn vẫn có hot reloading, debug…

Bước 4: khi nào bạn cần chạy lại câu lệnh pre-build ?
Mỗi lần bạn cài đặt thêm thư viện mới, chạy 2 câu lệnh bên dưới.

npx expo prebuild --clean
npm run android

Cách 2: chỉ thực hiện được, nếu bạn sở hữu điện thoại android/hoặc chạy máy ảo Android (simulator)

Cách làm này sẽ đảm bảo 100% quá trình bạn thực hiện và video là giống nhau (do cùng môi trường sử dụng)

Truy cập đường link sau:
https://expo.dev/go

Mục đích ở đây, là download file APK ứng với version cũ, sau đấy cài đặt.

Chọn version SDK 51, download và sử dụng (thay vì tải từ CH Play). Cách làm này chỉ áp dụng với môi trường Android.

Lưu ý: nên cài đặt trên máy ảo, và không nên cài trực tiếp trên điện thoại của bạn

#168. Expo Go trên điện thoại cá nhân kết nối tới backend localhost ?

Tài liệu: https://code.visualstudio.com/docs/editor/port-forwarding

Mục tiêu: chuyển đường link localhost thành url thực tế có thể truy cập ngoài internet

Bạn chỉ cần làm theo video này, nếu như bạn test ứng dụng React Native (Expo) với điện thoại cá nhân (kể từ chapter 7 - dự án thực hành)

Bước 1: Chạy dự án thực hành backend
Kết quả của bước này, là chúng ta có dự án backend chạy tại
http://localhost:8080/

Bước 2: Sử dụng forward port với VSCode

Để làm điều này, bạn cần có tài khoản github

Bước 3: chuyển chế độ từ Private sang Public

Bước 4: update file .env của dự án với đường link backend vừa tạo

Bước 5: Test thành quả

#169. Cách Update Backend (nếu cần thiết) ?

Mục đích: nếu bạn đã kích hoạt thành công backend, và source code backend của mình update (ví dụ fix bug), thì bạn cần xử lý như thế nào ?

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.

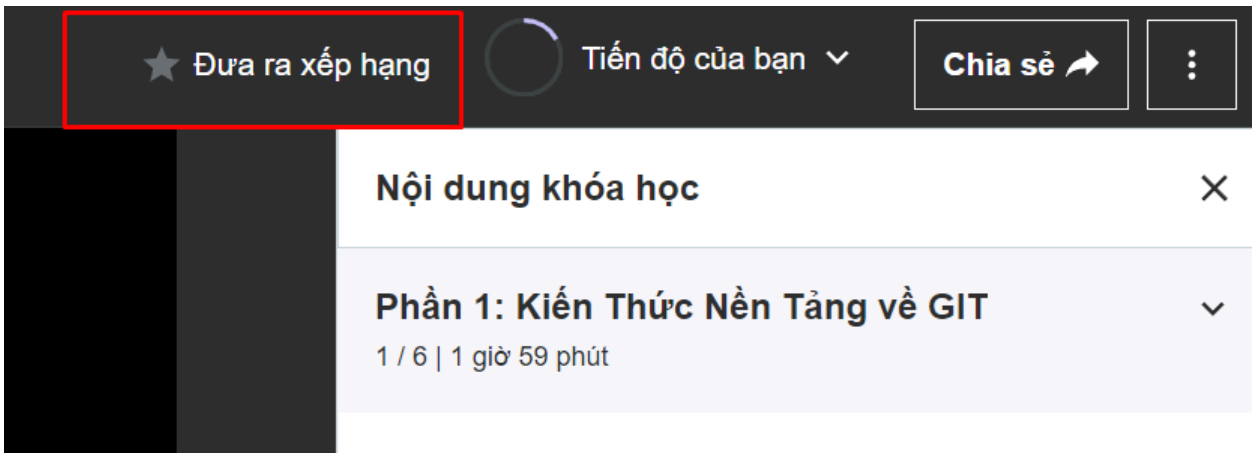
Mình cũng luôn sẵn sàng trả lời câu hỏi của bạn - hãy gửi tin nhắn trực tiếp cho mình bất cứ lúc nào.

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

Tạo review đánh giá:

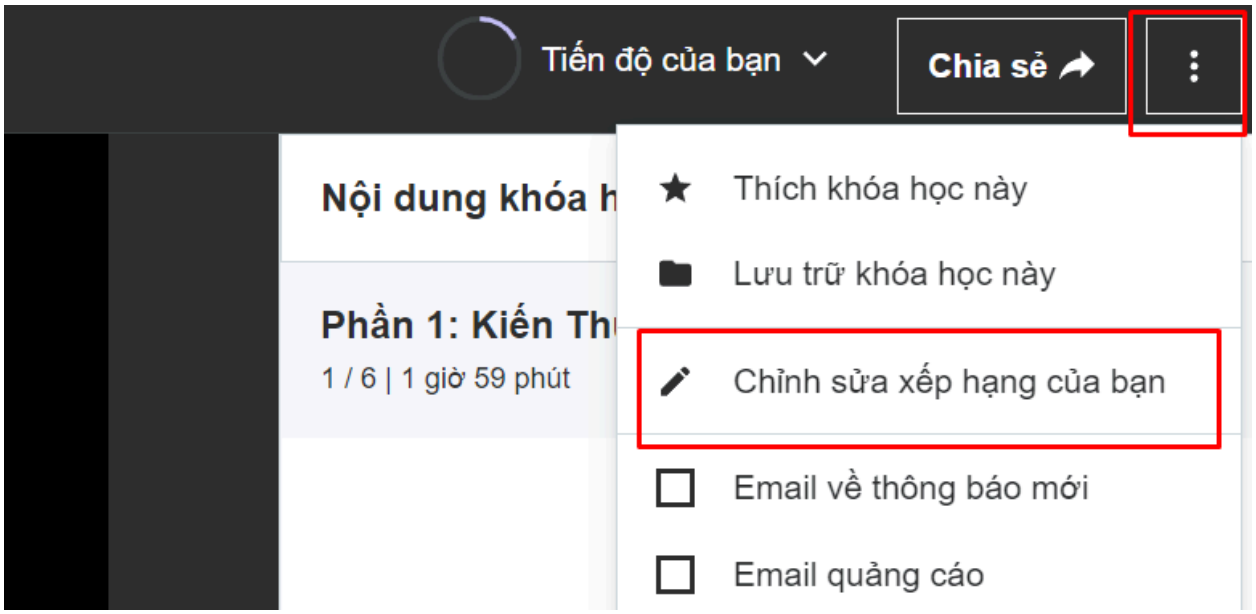
Nhấn vào “đưa ra xếp hạng”. Sau đấy rating (star) và viết đánh giá chia sẻ trải nghiệm.

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 150+ video về sử dụng React Native dành cho front-end mobile 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 vs React Native.

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)