

(49) ● Let's Build Twitter with SwiftUI (iOS 15, Xcode 13, Firebase, SwiftUI 3.0) - YouTube

Main subtitle

what's going on youtube welcome back to the channel it's stefan here from app stuff and i have some absolutely mind-blowing content for you guys today and i hope you're as excited about it as i am man i'm excited to bring this to you guys we are going to be building a twitter clone from scratch that's front end back end design everything you name it we're building it today we got six plus hours of content we're building it with swift ui and ios 15. it's going to be a lot of fun you guys are going to learn a lot and it's coming your way from me 100 free so i hope you guys are pumped up let's go ahead and check out what we're going to be building on that screen right there so let's just go through a quick demo of the application before we get started guys you notice that we open up to this absolutely gorgeous user feed it displays a list of all the tweets that users have made on the application we're pulling this information in from a real

Second subtitle

chuyện gì đang xảy ra trên youtube chào mừng bạn quay trở lại kênh đó là stefan ở đây từ ứng dụng công cụ và tôi có một số hoàn toàn nội dung hấp dẫn cho các bạn ngày hôm nay và tôi hy vọng bạn cũng hào hứng với nó như tôi là đàn ông, tôi rất vui khi mang cái này đến các bạn, chúng ta sẽ xây dựng một twitter sao chép từ đầu đó là phía trước kết thúc thiết kế kết thúc mọi thứ bạn đặt tên nó chúng tôi đang xây dựng nó hôm nay chúng tôi có sáu cộng với hàng giờ nội dung chúng tôi đang xây dựng nó với swift ui và ios 15. nó sẽ vui vẻ nhé các bạn học hỏi rất nhiều và nó sẽ đến theo cách của bạn từ tôi 100 miễn phí vì vậy tôi hy vọng các bạn được bơm lên hãy tiếp tục và kiểm tra những gì chúng ta sẽ được xây dựng trên màn hình đó ngay tại đó vì vậy chúng ta hãy đi qua một bản demo nhanh ứng dụng trước khi chúng tôi bắt đầu các bạn chú ý rằng chúng tôi cởi mở với điều này người dùng hoàn toàn tuyệt đẹp cho nó ăn hiển thị danh sách tất cả các tweet mà người dùng đã thực hiện trên ứng dụng chúng tôi lấy thông tin này từ một thực tế

Main subtitle

back end and we're going to be going over how to do all of that stuff in the course we can also like tweets or

interact with them really so you notice i can like a tweet here you notice that

these tweets were previously liked so

we're saving that information in the database which is awesome i can also unlike my tweet and it updates my ui

very smoothly

and that is looking absolutely incredible and you guys will notice that

i also have this beautiful side menu here that is identical to what you see in the twitter app i have a couple different options that i can click on i

also display some real life user data

about the currently logged in user so let's just go ahead and check out the user profile now you notice that this

once again looks exactly like twitter it displays some real life data that's

coming from a back end about the currently logged in user as we saw

before and what's my favorite part about this profile is guys is this awesome like filter bar that we see here that

displays different types of tweets the

Second subtitle

kết thúc và chúng ta sẽ đi về cách làm tất cả những thứ đó trong tất nhiên chúng ta cũng có thể thích các tweet hoặc

tương tác với họ thực sự để bạn chú ý tôi có thể thích một tweet ở đây bạn chú ý rằng

những tweet này trước đây đã được thích như vậy

chúng tôi đang lưu thông tin đó trong cơ sở dữ liệu thật tuyệt vời tôi cũng có thể không giống như tweet của tôi và nó cập nhật giao diện người dùng của tôi

rất suôn sẻ

và đó là tìm kiếm hoàn toàn đáng kinh ngạc và các bạn sẽ nhận thấy rằng

tôi cũng có menu bên đẹp này ở đây giống hệt với những gì bạn thấy trong ứng dụng twitter tôi có một vài các tùy chọn khác nhau mà tôi có thể nhấp vào tôi

cũng hiển thị một số dữ liệu người dùng thực tế

về người dùng hiện đang đăng nhập chúng ta hãy tiếp tục và kiểm tra

hồ sơ người dùng bây giờ bạn nhận thấy rằng điều này

một lần nữa trông giống hệt như twitter nó hiển thị một số dữ liệu thực tế cuộc sống đó là

đến từ một back-end về

người dùng hiện đang đăng nhập như chúng ta đã thấy

trước đây và phần yêu thích của tôi là gì hồ sơ này là mọi người thật tuyệt vời giống như thanh bộ lọc mà chúng ta thấy ở đây

hiển thị các loại tweet khác nhau

Main subtitle

user has made or interacted with so for

example clicking on this likes tab

displays tweets that this user has liked

going back to tweets just displays the

list of tweets this user has made

themselves so that looks incredible

we also have this beautiful search page

that allows us to search for different users on the app so for example i can

filter my users down just like that

click on it and it takes me over to that user's profile and then i can also you

know let's go over like lewis hamilton's

page and it displays real data about lewis hamilton his unique tweets the

tweets that he has liked and this is all looking absolutely phenomenal and the last piece of the main interface guys is going to be our ability to click on this

like floating button here you'll notice that it's um

what's it called it doesn't move as we

scroll through this list so i click on it and it takes me to this page where i can actually upload a tweet and have that get displayed in my user feed so

last up is we can log out of the app

and it will log us out and take us to

Second subtitle

người dùng đã thực hiện hoặc tương tác với như vậy cho

ví dụ nhấp vào tab thích này

hiển thị các tweet mà người dùng này đã thích

quay lại tweet chỉ hiển thị

danh sách các tweet mà người dùng này đã thực hiện

chính họ để trông đáng kinh ngạc

chúng tôi cũng có trang tìm kiếm tuyệt đẹp này

cho phép chúng tôi tìm kiếm các

người dùng trên ứng dụng, ví dụ như tôi có thể

lọc người dùng của tôi xuống giống như vậy

nhấp vào nó và nó sẽ đưa tôi đến đó

hồ sơ của người dùng và sau đó tôi cũng có thể bạn

biết chúng ta hãy đi qua như lewis hamilton's

trang và nó hiển thị dữ liệu thực về lewis hamilton những dòng tweet độc đáo của anh ấy

tweet mà anh ấy đã thích và đây là tất cả trông hoàn toàn phi thường và phần cuối cùng của giao diện chính là sẽ là khả năng của chúng tôi để nhấp vào đây

như nút nổi ở đây bạn sẽ nhận thấy đó là ừm

nó được gọi là gì nó không di chuyển khi chúng ta

cuộn qua danh sách này để tôi nhấp vào nó và nó đưa tôi đến trang này, nơi tôi thực sự có thể tải lên một tweet và có được hiển thị trong nguồn cấp dữ liệu người dùng của tôi để

cuối cùng là chúng ta có thể đăng xuất khỏi ứng dụng

và nó sẽ đăng xuất chúng tôi và đưa chúng

Main subtitle

this beautiful login page where we can actually log in as different users and also we have this registration page where we can sign different users up and upload information about you know creating a user that's what we're going to be building guys i'm super pumped i know you're super pumped let's go ahead and get started make sure you like and subscribe to the channel if you want to see more content like this and check out the links in the description for the source code to this application and other courses that i have online so let's go ahead and now finally get started with xcode guys okay guys so i have xcode opened up in front of me here and i created a new project if you guys don't know how to do that open up xcode go to file new and select project and it should bring you to the screen anyway we're going to create this app here so make sure you have ios selected as your tab and then we're going to select this app option and i'm going to name this twitter swift ui youtube

Second subtitle

tôi đến trang đăng nhập tuyệt đẹp này, nơi chúng ta có thể thực sự đăng nhập với tư cách người dùng khác nhau và chúng tôi cũng có trang đăng ký này nơi chúng tôi có thể đăng ký những người dùng khác nhau và tải lên thông tin về bạn biết tạo một người dùng đó là những gì chúng ta sẽ làm để được xây dựng guys tôi siêu bơm tôi biết bạn siêu bơm, hãy tiếp tục và bắt đầu đảm bảo bạn thích và đăng ký kênh nếu bạn muốn xem thêm nội dung như thế này và kiểm tra các liên kết trong phần mô tả cho mã nguồn cho ứng dụng này và các khóa học khác mà tôi có trực tuyến vì vậy hãy tiếp tục và bây giờ cuối cùng cũng có được bắt đầu với xcode guys được rồi các bạn vì vậy tôi đã mở xcode trước mặt tôi ở đây và tôi đã tạo một cái mới dự án nếu các bạn không biết làm mở xcode đi tới tệp mới và chọn dự án và nó sẽ mang lại cho bạn lên màn hình dù sao chúng ta cũng sẽ tạo ứng dụng này ở đây để chắc chắn rằng bạn có ios được chọn làm tab của bạn và sau đó chúng tôi sẽ chọn tùy chọn ứng dụng này và tôi sẽ đặt tên cho cái này twitter nhanh giao diện người dùng youtube

Main subtitle

okay and name this whatever you guys want and make sure that you have swift ui selected as the user interface and swift as the language and this stuff should be fine

we don't need core data and we don't

need to include tests so we should be

good there just hit next and go ahead and create your project

and to start this off guys before we actually get started with any code i actually like to organize my project directory here so it makes navigating through my code a lot easier so we're

going to create some folders over in this project directory here to get started so go ahead and select the first

folder in your project not the app

project name but this folder name and we're going to create a new group so go ahead just select new group and you guys will notice it creates a folder for

you there we're actually going to do this four times a shortcut to do it is

holding down command option and hitting n

and then just select your project folder

again and just do that a couple more times so you should have these four groups we're gonna name the first one

Second subtitle

được rồi và đặt tên cái này là gì các bạn muốn và đảm bảo rằng bạn có nhanh chóng ui được chọn làm giao diện người dùng và nhanh như ngôn ngữ và những thứ này sẽ ổn thôi

chúng tôi không cần dữ liệu cốt lõi và chúng tôi không

cần bao gồm các bài kiểm tra vì vậy chúng ta nên

tốt ở đó chỉ cần nhấn tiếp theo và tiếp tục và tạo dự án của bạn

và để bắt đầu điều này trước khi chúng tôi thực sự bắt đầu với bất kỳ mã nào tôi thực sự muốn tổ chức dự án của tôi thư mục ở đây để nó điều hướng thông qua mã của tôi dễ dàng hơn rất nhiều vì vậy chúng tôi

sẽ tạo một số thư mục trong thư mục dự án này ở đây để có được đã bắt đầu, vì vậy hãy tiếp tục và chọn cái đầu tiên

thư mục trong dự án của bạn không phải ứng dụng

tên dự án nhưng tên thư mục này và chúng tôi sẽ tạo một nhóm mới để tiếp tục chỉ cần chọn nhóm mới và bạn mọi người sẽ nhận thấy nó tạo một thư mục cho

bạn ở đó chúng tôi thực sự sẽ làm điều này bốn lần một phím tắt để làm điều đó là

giữ tùy chọn lệnh và nhấn

N

và sau đó chỉ cần chọn thư mục dự án của bạn

một lần nữa và chỉ cần làm điều đó một vài lần nữa

lần vì vậy bạn nên có bốn

nhóm chúng ta sẽ đặt tên cho nhóm đầu tiên

Main subtitle

extensions

we're gonna name the second one model
the third one we're going to call it
service

and the fourth one we're going to call
core

and within the core folder we're going
to create two groups there as well so
once again you can right click and hit

new group

and this one is going to be called
components

and then right click on core again and

create another group and we're going to
call this guy feed

okay and i actually like to drag
feed

let's see

i like that to be below um the
components guy so let's talk about this

structure really quick before we get

started with the code guys you guys are
probably used to seeing me organize this
as model view view model

but i recently started uh changing the
way i organize my directory to organize

it by feature instead of like
architecture components so
now this this folder for the feed will

contain everything about the feed
whereas before i would you know create a

Second subtitle

phần mở rộng

chúng ta sẽ đặt tên cho mô hình thứ hai
cái thứ ba chúng ta sẽ gọi nó là
dịch vụ

và cái thứ tư chúng ta sẽ gọi
cốt lõi

và trong thư mục lõi, chúng ta sẽ
để tạo hai nhóm ở đó cũng như vậy
một lần nữa bạn có thể nhấp chuột phải và
nhấn

nhóm mới

và cái này sẽ được gọi là
các thành phần

và sau đó nhấp chuột phải vào lõi một lần
nữa và

tạo một nhóm khác và chúng tôi sẽ
gọi anh chàng này thức ăn
được rồi và tôi thực sự thích kéo
cho ăn

hãy xem nào

tôi thích điều đó ở dưới um
anh chàng thành phần vì vậy hãy nói về
điều này

cấu trúc thực sự nhanh chóng trước khi
chúng tôi nhận được

bắt đầu với mã các bạn
có lẽ đã từng thấy tôi tổ chức việc này
dưới dạng xem mô hình xem mô hình
nhưng gần đây tôi đã bắt đầu thay đổi
cách tôi tổ chức thư mục của mình để tổ
chức

nó theo tính năng thay vì thích
thành phần kiến trúc như vậy
bây giờ thư mục này cho nguồn cấp dữ liệu
sẽ

chứa mọi thứ về nguồn cấp dữ liệu
trong khi trước đây tôi muốn bạn biết tạo
một

Main subtitle

folder for models view models and views
and once your app gets a lot of features
in it it actually gets kind of annoying
to navigate through all that stuff so
this is a little bit more efficient to
organize your folder structure by
feature instead of architecture
component like model view view model
anyway within feed we're going to create
another folder or two more folders
actually
so we're going to call this guy view
models and then we're going to call this
guy
view
so
this is what i was talking about right
now every time we want to go to
something associated with the feed we
know we just go to this feed folder and
it will be right there okay so let's
actually get started with some code now
so we're going to go
into this feed view folder and we're
going to create a new file it's going to
be a swift ui view and we're going to
hit next and we're going to call this
feed
view awesome

Second subtitle

thư mục cho các mô hình xem mô hình và
chế độ xem
và một khi ứng dụng của bạn có nhiều tính
năng
trong đó nó thực sự gây phiền nhiễu
để điều hướng qua tất cả những thứ đó
đây là một chút hiệu quả hơn để
tổ chức cấu trúc thư mục của bạn bằng
cách
tính năng thay vì kiến trúc
thành phần như chế độ xem mô hình chế độ
xem mô hình
dù sao trong nguồn cấp dữ liệu, chúng tôi
sẽ tạo
một thư mục khác hoặc hai thư mục nữa
Thực ra
vì vậy chúng ta sẽ gọi anh chàng này là
view
mô hình và sau đó chúng ta sẽ gọi đây là
chàng trai
xem
Vì thế
đây là những gì tôi đã nói về đúng
bây giờ mỗi khi chúng tôi muốn đi đến
một cái gì đó liên quan đến nguồn cấp dữ
liệu chúng tôi
biết rằng chúng tôi chỉ cần vào thư mục
nguồn cấp dữ liệu này và
nó sẽ ở ngay đó được thôi, vì vậy hãy
thực sự bắt đầu với một số mã bây giờ
vì vậy chúng ta sẽ đi
vào thư mục xem nguồn cấp dữ liệu này và
chúng tôi
sẽ tạo một tệp mới, nó sẽ
là một chế độ xem ui nhanh chóng và chúng
tôi sẽ
nhấn tiếp theo và chúng tôi sẽ gọi đây là
cho ăn
xem tuyệt vời

Main subtitle

okay so now let's just go ahead and hit resume guys i actually like to build this on an iphone 11 pro and let's just wait for this project to show up awesome so now we're seeing our preview showing up here and it has this text called uh that says hello world um just to make sure everything's updating let's just delete this and say hello and we should notice our preview updates as well awesome so we're good to go so let's actually start creating this feed view so we notice here guys that we have a list of information so this is actually going to be a list of tweets so we're going to replace this text that we see here with something called a scroll view and within that scroll view we're going to have something called a lazy v stack and let's just add some text sorry we're going to add a for each loop here and let's just go like zero up to 20

Second subtitle

được rồi, bây giờ chúng ta hãy tiếp tục và đánh sơ yếu lý lịch các bạn tôi thực sự thích xây dựng cái này trên iphone 11 pro Và chúng ta hãy chờ đợi cho dự án này để hiển thị lên tuyệt vời vì vậy bây giờ chúng tôi đang nhìn thấy của chúng tôi bản xem trước hiển thị ở đây và nó có cái này tin nhắn có tên uh nói xin chào thế giới ừm chỉ để đảm bảo mọi thứ đang cập nhật, hãy xóa cái này và nói Xin chào và chúng tôi sẽ nhận thấy các bản cập nhật xem trước của chúng tôi cũng tuyệt vời vì vậy chúng tôi rất tốt để đi vì vậy hãy thực sự bắt đầu tạo chế độ xem nguồn cấp dữ liệu này vì vậy chúng tôi nhận thấy ở đây các bạn rằng chúng tôi có một danh sách thông tin vì vậy đây thực sự là sẽ là một danh sách các tweet Vì thế chúng tôi sẽ thay thế văn bản này mà chúng tôi xem ở đây với một thứ gọi là cuộn xem và trong chế độ xem cuộn đó, chúng tôi sẽ để có một thứ gọi là lazy v stack và chúng ta hãy thêm một số văn bản xin lỗi, chúng tôi sẽ thêm một cho mỗi vòng lặp ở đây và chúng ta hãy đi như số không lên đến 20

Main subtitle

id backslash dot self and then we're going to open up our brackets and we are just going to say underscore in and now let's just go ahead and add some text here that says like tweet row view and we should see a bunch of these options or these text components coming up so that looks pretty good um we noticed that if we were to hit play on the preview that we could actually like scroll this stuff so this is the beginnings of what our feed view is going to look like we just have to replace um this text component with an actual tweet row view so let's go ahead and start by creating that now so in your components folder guys go ahead and create another folder and we're going to call this tweets and the reason we're creating or adding this tweet stuff into the components folder is because we're going to be reusing this tweet cell in a couple different places in our app and that's what the components folder is going to be for so for example i could go to my profile and notice that i'm reusing these tweet cells here

Second subtitle

id dấu gạch chéo ngược dấu chấm tự và sau đó chúng tôi sẽ mở dấu ngoặc của chúng tôi và chúng ta sẽ chỉ nói gạch dưới trong và bây giờ chúng ta hãy tiếp tục và thêm một số văn bản ở đây có nội dung như tweet xem hàng và chúng ta sẽ thấy một loạt những thứ này tùy chọn hoặc các thành phần văn bản sắp tới lên trông khá đẹp ừm chúng tôi nhận thấy rằng nếu chúng tôi nhấn play trên bản xem trước mà chúng tôi thực sự có thể như cuộn những thứ này vì vậy đây là khởi đầu của chế độ xem nguồn cấp dữ liệu của chúng tôi là gì sẽ có vẻ như chúng ta chỉ cần phải thay thế ừm thành phần văn bản này với một thực tế chế độ xem hàng tweet, vì vậy hãy tiếp tục và bắt đầu bằng cách tạo nó ngay bây giờ để trong của bạn thư mục thành phần đi trước và tạo một thư mục khác và chúng tôi sẽ gọi đây là tweet và lý do chúng tôi đang tạo hoặc thêm nội dung tweet này vào các thành phần thư mục là bởi vì chúng ta sẽ trở thành sử dụng lại tế bào tweet này trong một vài những nơi khác nhau trong ứng dụng của chúng tôi và đó là thư mục thành phần sẽ là gì ví dụ như vậy tôi có thể vào hồ sơ của mình và nhận thấy rằng tôi đang sử dụng lại các ô tweet này ở đây

Main subtitle

right and it also gets reused in this like section this tweet section so we

definitely want this to be a reusable component

so in tweets go ahead and create something called it's a swift ui view

and we're going to call this tweet row view

and let's start by resuming on our preview

just to make sure that we can see this

stuff showing up as we build it looks good

now uh let's go back to this to make

sure we can see what we want it to look

like okay so

we're gonna have like an image uh some text about the user the text that represents the tweet caption and then we're gonna have this like row of action

buttons so let's just go ahead and get started so this is all gonna be contained within a v stack here guys so within that v stack we're going to create an h stack

the alignment is going to be dot top spacing is going to be 12.

and then we're going to have just like a circle

we're going to replace the circle with an actual image later on

we're going to say dot frame

Second subtitle

đúng và nó cũng được sử dụng lại trong này thích phần này phần tweet này vì vậy chúng tôi

chắc chắn muốn cái này có thể tái sử dụng thành phần

vì vậy trong các tweet, hãy tiếp tục và tạo một cái gì đó được gọi là một giao diện người dùng nhanh chóng

và chúng ta sẽ gọi dòng tweet này xem

và hãy bắt đầu bằng cách đang tiếp tục trên bản xem trước của chúng tôi

chỉ để đảm bảo rằng chúng ta có thể thấy điều này

những thứ hiển thị khi chúng tôi xây dựng Tốt

bây giờ chúng ta hãy quay trở lại vấn đề này để thực hiện

chắc chắn chúng ta có thể thấy những gì chúng ta muốn

thích được vậy

chúng ta sẽ có một hình ảnh uh some văn bản về người dùng văn bản đó đại diện cho chú thích tweet và sau đó chúng ta sẽ có điều này giống như hàng hành động

các nút vì vậy hãy tiếp tục và nhận bắt đầu vì vậy đây là tất cả sẽ được chứa trong ngăn xếp v ở đây các bạn trong ngăn xếp v đó, chúng ta sẽ tạo ngăn xếp h

sự liên kết sẽ là dot top khoảng cách sẽ là 12.

và sau đó chúng ta sẽ có giống như một vòng tròn chúng ta sẽ thay thế vòng tròn bằng một hình ảnh thực tế sau này chúng ta sẽ nói khung chấm

Main subtitle

width and height is 56 by 56 we don't

need the alignment

so that looks better

and then we're just going to say we could give it a foreground color of like color

dot

system

blue okay

and then after that guys we're gonna have a v stack within this h stack so

let me explain what's going on here so

the overall like parent container is a v

stack because we have things vertically

stacked on top of each other and then here in this h stack

we are going to have the image and then all of this stuff over here

but this these components here the user

info and the caption that's its own

vertical stack so let me just go ahead

and add some comments here

so we can sort of

make this clear so this is going to be

user info

and tweet

caption

and this h stack is going to be

uh

profile image

plus

user info

Second subtitle

chiều rộng và chiều cao là 56 x 56 chúng tôi không

cần sự liên kết

như vậy trông đẹp hơn

và sau đó chúng ta sẽ nói rằng chúng ta có thể cho nó một màu nền trước giống như màu sắc

dấu chấm

hệ thống

màu xanh được

và sau đó các bạn chúng ta sẽ

có một ngăn xếp v trong ngăn xếp h này vì vậy

hãy để tôi giải thích những gì đang xảy ra ở đây

tổng thể giống như vùng chứa chính là một v

ngăn xếp vì chúng ta có những thứ theo chiều dọc

xếp chồng lên nhau rồi

ở đây trong ngăn xếp h này

chúng ta sẽ có hình ảnh và sau đó

tất cả những thứ này ở đây

nhưng đây là những thành phần ở đây người dùng

thông tin và chú thích đó là của riêng mình

ngăn xếp dọc vì vậy hãy để tôi tiếp tục

và thêm một số ý kiến ở đây

vì vậy chúng ta có thể sắp xếp

làm cho điều này rõ ràng vì vậy điều này sẽ được

thông tin người dùng

và tweet

đầu đề

và ngăn xếp h này sẽ là

uh

hình ảnh hồ sơ cá nhân

thêm

thông tin người dùng

Main subtitle

plus tweet

okay

so now let's go into this v stack we're

going to have a text component it's
going to just say bruce wayne
and we're going to give it a font of

let's see dot sub headline and then dot
bold and then the
foreground color we don't need to change
and then we're just going to say we want

a text

and that's going to be at
batman

oh wait this is actually an h stack
sorry guys so
cut that out and this is going to be in
an h stack

and this is the user info

and then below that is going to be tweet
caption

so paste that back in there

okay so that's looking pretty good then
on batman you're going to say foreground
color is dot gray

and then the font is dot caption

and you notice that makes it nice and
small there

and then we are going to have another
text component that is going to say uh
just like

49

weeks or like we could say 2w for two

weeks

and then we're just gonna copy and paste

Second subtitle

cộng với tweet

được rồi

Vì vậy, bây giờ chúng ta hãy đi vào ngăn
xếp v này, chúng ta

nó sẽ có một thành phần văn bản
sẽ chỉ nói bruce wayne

và chúng tôi sẽ cung cấp cho nó một phông
chữ

xem chấm sub tiêu đề rồi chấm

in đậm và sau đó là

màu nền trước chúng ta không cần thay đổi
và sau đó chúng ta sẽ nói rằng chúng ta
muốn

một văn bản

và đó sẽ là lúc

người đợi

chờ đã, đây thực sự là một ngăn xếp h
xin lỗi các bạn vì vậy

cắt cái đó ra và cái này sẽ ở trong
một ngăn xếp h

và đây là thông tin người dùng

và sau đó bên dưới sẽ là dòng tweet
đầu đề

vì vậy hãy dán lại vào đó

được rồi vậy là trông khá ổn rồi

trên batman bạn sẽ nói tiền cảnh

màu sắc là chấm xám

và sau đó phông chữ là chú thích dấu chấm

và bạn nhận thấy điều đó làm cho nó đẹp và
nhỏ ở đó

và sau đó chúng ta sẽ có một cái khác

thành phần văn bản sẽ nói uh

giống như

49

tuần hoặc giống như chúng ta có thể nói 2w
cho hai

tuần

và sau đó chúng ta sẽ sao chép và dán

Main subtitle

these two things

um

to modify

our font right there and that looks

pretty good and then here we're gonna go outside of this h stack okay and that's because now

we're we're outside of this horizontal horizontally stacked group of components

and now we want to add the tweet caption

so

let me go below my tweet caption comment

and just say text

i

believe

in harvey dent

and then we're going to say font dot sub headline

dot multi-line text alignment

is that leading so what this

line of code is going to do guys is make

it so that if our tweet is multiple lines

then it's going to make sure that it aligns to the leading edge

in as opposed to being centered or trailing

okay but we have an issue here we notice

that this does not line up where we want

it to

so on this v stack

we're going to go uh and say alignment

Second subtitle

hai điều này

ô

để sửa đổi

phông chữ của chúng tôi ngay tại đó và có vẻ như

khá tốt và sau đó chúng ta sẽ đi bên ngoài ngăn xếp h này được và đó là bởi vì bây giờ chúng ta ở ngoài chiều ngang này nhóm các thành phần xếp chồng lên nhau theo chiều ngang và bây giờ chúng tôi muốn thêm chú thích tweet

Vì thế

hãy để tôi đi bên dưới bình luận chú thích tweet của tôi

và chỉ nói văn bản

Tôi

tin tưởng

trong harvey

và sau đó chúng ta sẽ nói font dot sub tiêu đề

chấm căn chỉnh văn bản nhiều dòng đó có phải là hàng đầu vậy cái này là gì dòng mã sẽ làm mọi người đang làm nó để nếu tweet của chúng tôi là nhiều dòng

sau đó nó sẽ đảm bảo rằng nó thẳng hàng với cạnh hàng đầu

trái ngược với việc tập trung hoặc dấu vết

được rồi nhưng chúng tôi có một vấn đề ở đây chúng tôi nhận thấy

rằng điều này không xếp hàng ở nơi chúng tôi muốn

nó để

vì vậy trên ngăn xếp v này

chúng ta sẽ đi uh và nói thẳng hàng

Main subtitle

is dot leading

and we're going to give it some spacing

of 4 pixels as well

okay so that looks pretty good guys
it's looking just like we have in our
completed application

then we just need this uh

group of buttons down below

okay so

let's go ahead and create that now and
let me just put a comment here for

action buttons and we can see here that

this is

contained within that overall sort of
parent v stack so it's going to go below
all of this stuff right so this is this

h stack here

let's see that's our h stack yeah so
that would have been a mistake we want
to make sure we're outside the h stack

right

and that we're in the parent v stack so

that it goes below this stuff so you
guys would notice if you highlight the h
stack there it's all this and then the v

stack

is going to be something else so
we could just go here and say like text

action buttons and we notice that it

shows up below all of that stuff which

Second subtitle

đang dẫn đầu

và chúng tôi sẽ cung cấp cho nó một số
khoảng cách

của 4 pixel là tốt

được rồi, trông khá ổn đây các bạn

nó trông giống như chúng ta có trong

Đơn hoàn thành

sau đó chúng ta chỉ cần điều này uh

nhóm nút bên dưới

được thôi, nếu vậy

hãy tiếp tục và tạo ra nó ngay bây giờ và

hãy để tôi chỉ cần đặt một bình luận ở đây
cho

các nút hành động và chúng ta có thể thấy
ở đây

đây là

chứa trong loại tổng thể của

cha mẹ v stack vì vậy nó sẽ đi bên dưới

tất cả những thứ này đúng vì vậy đây là cái
này

h ngăn xếp ở đây

hãy xem đó là ngăn xếp h của chúng ta

đó sẽ là một sai lầm chúng ta muốn

để đảm bảo rằng chúng tôi ở bên ngoài
ngăn xếp h

Phải

và rằng chúng tôi đang ở trong ngăn xếp
cha v

rằng nó đi bên dưới những thứ này để bạn
mọi người sẽ chú ý nếu bạn làm nổi bật h
ngăn xếp ở đó là tất cả những thứ này và
sau đó là v

cây rơm

sẽ là một cái gì đó khác vì vậy

chúng ta chỉ có thể vào đây và nói thích văn
bản

các nút hành động và chúng tôi nhận thấy
rằng nó

hiển thị bên dưới tất cả những thứ mà

Main subtitle

is great

but okay let's create this uh group of action buttons now so this is going to be you guys might have guessed it an h stack because it's horizontally stacked as a

group of buttons

and we're going to create a button and it's going to be action and label just go ahead and hit enter there action we're not going to have anything just yet

we can say action goes here and then the label is just going to be a pretty simple button so i'm going to say or we can just make it an image actually it's going to be system name and the first one is going to be bubble dot left then we're going to say dot font

dot sub headline

dot

and that should be good

cool

and then we're gonna copy and paste this

a couple times guys

uh so we should have four buttons right and after the first button we're gonna add a spacer after the next one we're going to add a spacer after the next one we're going to add a spacer

Second subtitle

tuyệt

nhưng được rồi, hãy tạo cái này uh nhóm các nút hành động bây giờ vì vậy đây sẽ là các bạn có thể đã đoán nó là một ngăn xếp h bởi vì nó được xếp chồng lên nhau theo chiều ngang như một nhóm nút và chúng ta sẽ tạo một nút và nó sẽ là hành động và nhãn chỉ cần tiếp tục và nhấn enter ở đó hành động chúng ta sẽ không có gì vừa mới

chúng ta có thể nói hành động diễn ra ở đây và sau đó nhãn sẽ chỉ là một nút khá đơn giản vì vậy tôi sẽ nói hoặc chúng ta có thể làm nó là một hình ảnh thực sự nó sẽ là tên hệ thống và cái đầu tiên sẽ là bong bóng chấm trái

sau đó chúng ta sẽ nói phông chữ dấu chấm

chấm tiêu đề phụ

dấu chấm

và điều đó sẽ tốt

mát mẻ

và sau đó chúng ta sẽ sao chép và dán cái này

vài lần nhé mọi người

uh vậy chúng ta nên có bốn nút phải không và sau cái nút đầu tiên, chúng ta sẽ thêm một miếng đệm

sau cái tiếp theo, chúng ta sẽ thêm một miếng đệm

sau cái tiếp theo, chúng ta sẽ thêm một miếng đệm

Main subtitle

and you guys will notice that that
equally spaces out all of our buttons so

that looks really good now we just need
to change the image names and adjust our

spacing here

so

the second one is going to be
arrow.2.square path
and if you guys are wondering where i'm

getting these images from it's from this
app called sf symbols if you just google

it you should be able to download it and
it just gives you an awesome
set of icons that you can access through

xcode

so all these names that you're seeing

here so like if i typed in arrow.2
that's where i get this
arrow.2.squarepath which is what we see

right here as like a retweet button so
that's pretty cool definitely recommend

getting that app it's super helpful
anyway let's keep going
the next one is going to be heart
and the next one is going to be
a bookmark
awesome

okay so now we're just going to want to
give this uh let's see this v stack i

believe

Second subtitle

và các bạn sẽ nhận thấy rằng
cách đều nhau ra khỏi tất cả các nút của
chúng tôi để
trông thực sự tốt bây giờ chúng ta chỉ cần
để thay đổi tên hình ảnh và điều chỉnh của
chúng tôi
khoảng cách ở đây
Vì thế
cái thứ hai sẽ là
mũi tên.2.đường dẫn hình vuông
và nếu các bạn đang thắc mắc tôi đang ở
đâu
lấy những hình ảnh này từ nó là từ cái này
ứng dụng được gọi là biểu tượng sf nếu bạn
chỉ google
nó bạn sẽ có thể tải về nó và
nó chỉ mang đến cho bạn một điều tuyệt vời
tập hợp các biểu tượng mà bạn có thể truy
cập thông qua
xcode
vì vậy tất cả những cái tên mà bạn đang
nhìn thấy
ở đây giống như nếu tôi gõ vào mũi tên.2
đó là nơi tôi nhận được điều này
arrow.2.squarepath đó là những gì chúng ta
thấy
ngay tại đây giống như một nút retweet vậy
điều đó khá tuyệt, chắc chắn khuyên bạn
nên
nhận được ứng dụng đó, nó rất hữu ích
dù sao chúng ta hãy tiếp tục đi
cái tiếp theo sẽ là trái tim
và cái tiếp theo sẽ là
một dấu trang
tuyệt vời
được rồi, bây giờ chúng ta sẽ chỉ muốn
đưa cái này uh chúng ta hãy xem cái v stack
này
tin tưởng

Main subtitle

so i'm actually going to fold this up
 guys because it's pretty
 uh it's a lot of code
 so just go you can double click on this

h stack

and command click on that last bracket
 and just hit fold and it'll fold it up
 nicely there so that your codes a little
 bit more readable
 and on this v stack let's see i just
 want to add some padding
 and that looks pretty good
 we do want more space there though

so let's add some padding on this h
 stack

right that looks way better and then we
 want this guy right here to be aligned

to the left

so on my v stack my parent v stack here

um let's see if we can just go up to the
 top and make the alignment.leading
 and awesome that looks great so
 this is uh pretty much it for our tweet
 cell guys so now we should be able to go

back to our feed view

and drop this tweet cell into or sorry
 tweet row view
 into
 our feed and then hit resume

and let's see how that looks
 uh sometimes you gotta hit resume a
 couple times okay so that's looking

Second subtitle

vì vậy tôi thực sự sẽ gấp cái này lại
 mọi người vì nó đẹp
 uh nó rất nhiều mã
 vì vậy chỉ cần đi, bạn có thể nhấp đúp chuột
 vào đây
 ngăn xếp h
 và lệnh nhấp vào dấu ngoặc cuối cùng đó
 và chỉ cần nhấn gấp và nó sẽ gấp lại
 độc đáo ở đó để mã của bạn một chút
 dễ đọc hơn một chút
 và trên ngăn xếp v này, hãy xem tôi chỉ
 muốn thêm một số phần đệm
 và điều đó có vẻ khá tốt
 mặc dù chúng tôi muốn có nhiều không gian
 hơn ở đó
 vì vậy hãy thêm một số phần đệm vào h này
 cây rơm
 đúng là có vẻ tốt hơn và sau đó chúng tôi
 muốn anh chàng này ngay tại đây được liên
 kết
 Qua bên trái
 vì vậy trên ngăn xếp v của tôi, ngăn xếp v
 của cha mẹ tôi ở đây
 ừm, hãy xem liệu chúng ta có thể đi lên
 top và căn chỉnh.leading
 và tuyệt vời thật tuyệt
 uh đây là khá nhiều cho tweet của chúng tôi
 các bạn di động vì vậy bây giờ chúng ta sẽ
 có thể đi
 quay lại chế độ xem nguồn cấp dữ liệu của
 chúng tôi
 và thả ô tweet này vào hoặc xin lỗi
 chế độ xem hàng tweet
 vào trong
 nguồn cấp dữ liệu của chúng tôi và sau đó
 nhấn tiếp tục
 và hãy xem nó trông như thế nào
 uh đôi khi bạn phải nhấn tiếp tục một
 vài lần được rồi vì vậy đó là tìm kiếm

Main subtitle

pretty pretty pretty good uh we just need a divider back in that view so let's go ahead and add that back in our tweet row view um at the bottom of your v stack just go here and say divider and nice that has a divider showing up um something about dividers that i don't like is that like they um get cut off like with the padding so let's see if we take that padding off yeah that looks a little better so we're gonna take the padding off of the the end there and just add our divider and back in our feed view we can go here and say dot padding and i think that should fix our problem so resume it one more time okay it does the same thing which isn't the worst thing in the world right it doesn't look terrible if it's really bothersome we can come back and fix that but this is looking pretty good guys so let's actually go ahead and run this

Second subtitle

khá khá khá tốt uh chúng tôi chỉ cần một dải phân cách trở lại trong chế độ xem đó hãy tiếp tục và thêm nó trở lại vào chế độ xem hàng tweet ừm ở phía dưới ngăn xếp v của bạn chỉ cần vào đây và nói dải phân cách và thật tuyệt khi có một dải phân cách hiển thị ừm có điều gì đó về dải phân cách mà tôi không biết giống như vậy giống họ ừm bị cắt như với phần đệm vì vậy hãy xem liệu chúng ta có bỏ phần đệm đó ra không vâng, điều đó có vẻ tốt hơn một chút vì vậy chúng tôi sẽ lấy phần đệm ra kết thúc ở đó và chỉ cần thêm của chúng tôi bộ chia và quay lại chế độ xem nguồn cấp dữ liệu của chúng tôi chúng ta có thể vào đây và nói dấu chấm đệm và tôi nghĩ rằng nên khắc phục sự cố của chúng tôi vì vậy hãy tiếp tục nó một lần nữa được rồi nó làm điều tương tự đó không phải là điều tồi tệ nhất trên thế giới đúng là nó không có vẻ khủng khiếp nếu nó thực sự khó chịu chúng ta có thể đến quay lại và sửa lỗi đó nhưng điều này đang tìm kiếm các bạn khá tốt vì vậy chúng ta hãy thực sự đi trước và chạy cái này

Main subtitle

and see what it's looking like on the simulator
build succeeded so we're looking good

and
after this if this is looking good oh we actually need to drop our feed view in to the content view
so let's go to our twitter swift ui

youtube app file right right this is where uh the app actually launches replace content view with feed view

and we're going to build our tab view
after this guys okay so this is looking

awesome right
we have this nice scrollable list of tweets we don't have it populated with actual data yet and something i noticed

is we actually want these buttons to be gray so let's go back to our tweets view

or tweet row view
let's re-expand this guy or sorry
whoa where did my
okay

so
here we're just going to add a foreground color of dot gray

and then we can fold it back up
oops that was the z stack
so foreground color is dot gray

Second subtitle

và xem nó trông như thế nào trên giả lập
xây dựng thành công vì vậy chúng tôi đang tìm kiếm tốt

Và
sau này nếu điều này đang tìm kiếm tốt oh chúng tôi
thực sự cần bỏ chế độ xem nguồn cấp dữ liệu của chúng tôi trong
đến chế độ xem nội dung
Vì vậy, hãy truy cập twitter nhanh chóng của chúng tôi ui

tập tin ứng dụng youtube đúng rồi đây là ứng dụng thực sự khởi chạy ở đâu thay thế chế độ xem nội dung bằng chế độ xem nguồn cấp dữ liệu
và chúng tôi sẽ xây dựng chế độ xem tab của chúng tôi

sau những người này được rồi vì vậy đây là tìm kiếm
đúng tuyệt vời

chúng tôi có danh sách có thể cuộn đẹp này tweet chúng tôi không có nó phô biến với dữ liệu thực tế chưa và một cái gì đó tôi nhận thấy

là chúng tôi thực sự muốn các nút này được màu xám, vì vậy hãy quay lại chế độ xem tweet của chúng tôi

hoặc chế độ xem hàng tweet
hãy mở rộng lại anh chàng này hoặc xin lỗi ai đã làm của tôi ở đâu
được rồi

Vì thế
ở đây chúng ta sẽ chỉ thêm một màu nền chấm xám

và sau đó chúng ta có thể gấp nó lại rất tiếc đó là ngăn xếp z
vì vậy màu nền trước là chấm xám

Main subtitle

and let's fold it back up
and delete that padding comment
run it one more time
and that's looking really really good
guys

so

next thing i want to do is actually get

um this tab view showing up
so that this is like a lot more like the

main twitter interface that you're going

to see where you have this tab component
down at the bottom that allows you to
navigate between different views so

that's what we're going to be doing next
all right guys so let's create another
folder inside of our core folder so i'm

just going to right click hit new group
and then i'm going to call this
main tab

and i'm going to put the components
folder above that and inside main tab
we're going to create a new swift ui
view

and that is going to be called main tab
view

and this is where we're going to create
our main tab so let's just go ahead and

get right into it we're going to replace
text with a tab view
and we're going to select this option
that has a selection component or input
parameter

Second subtitle

và hãy gấp nó lại
và xóa bình luận đệm đó
chạy nó một lần nữa
và điều đó trông thực sự rất tốt
các bạn

Vì thế

điều tiếp theo tôi muốn làm là thực sự có
được
ừm, chế độ xem tab này đang hiển thị
để điều này giống như nhiều hơn nữa giống
như

giao diện twitter chính mà bạn đang sử
dụng
để xem nơi bạn có thành phần tab này
xuống ở phía dưới cho phép bạn
điều hướng giữa các chế độ xem khác nhau
để

đó là những gì chúng ta sẽ làm tiếp theo
được rồi các bạn, vậy hãy tạo một cái khác
thư mục bên trong thư mục cốt lõi của
chúng tôi vì vậy tôi
chỉ cần nhấp chuột phải vào nhóm mới
và sau đó tôi sẽ gọi cái này
tab chính

và tôi sẽ đặt các thành phần
thư mục phía trên đó và bên trong tab chính
chúng ta sẽ tạo một Swift ui mới
xem

và đó sẽ được gọi là tab chính
xem

và đây là nơi chúng ta sẽ tạo
tab chính của chúng tôi, vì vậy hãy tiếp tục
và

đi thẳng vào nó, chúng ta sẽ thay thế
văn bản với chế độ xem tab
và chúng tôi sẽ chọn tùy chọn này
có một thành phần lựa chọn hoặc đầu vào
tham số

Main subtitle

and this is going to help us keep track of our selected tab index so for example

if i if i select home

my selected tab index is the home tab or

the first tab

if i select my notifications tab

then that is going to be my new selective tab so we're going to have to create a state property to keep track of that selected index so we're going to say at state

private var selected index which is an integer

equals zero and we actually don't need to cast it we can just say equals zero

and we're going to pass in that dollar sign selected index here and then hit enter on the content

and the first thing we're going to create is our feed view

and then we're going to say dot on tap gesture

perform

self dot selected index equals zero

and then we're going to give it a dot tab item

and the label is just going to be an image with system name which is house and then we're going to give it a tag of 0 as well

and then we can just copy and paste this

a couple times guys

and we're just going to change the index

Second subtitle

và điều này sẽ giúp chúng tôi theo dõi của chỉ mục tab đã chọn của chúng tôi chẳng hạn

nếu tôi nếu tôi chọn nhà

chỉ mục tab đã chọn của tôi là tab trang chủ hoặc

tab đầu tiên

nếu tôi chọn tab thông báo của mình

thì đó sẽ là cái mới của tôi

tab chọn lọc vì vậy chúng ta sẽ phải tạo một tài sản nhà nước để theo dõi chỉ mục đã chọn đó, vì vậy chúng tôi sẽ nói ở trạng thái

private var đã chọn chỉ mục đó là một số nguyên

bằng không và chúng tôi thực sự không cần để truyền nó, chúng ta chỉ có thể nói bằng không

và chúng ta sẽ chuyển số đó la đó ký chỉ mục đã chọn ở đây và sau đó nhấn nhập vào nội dung

và điều đầu tiên chúng ta sẽ làm tạo là chế độ xem nguồn cấp dữ liệu của chúng tôi

và sau đó chúng ta sẽ nói dot on tap cử chỉ

trình diễn

tự chấm chỉ số đã chọn bằng không

và sau đó chúng ta sẽ cho nó một dấu chấm mục tab

và nhấn sẽ chỉ là một

hình ảnh với tên hệ thống là nhà

và sau đó chúng tôi sẽ cho nó một thẻ 0 cũng vậy

và sau đó chúng ta chỉ cần sao chép và dán cái này

vài lần nhé mọi người

và chúng ta sẽ thay đổi chỉ số

Main subtitle

property so it's going to be self.selected self.selectedIndexes1 tag is one two two three three and let's just go ahead and create our other uh tab views so right now we're just reusing the feed view a bunch of times we don't really want to do that we want to create our other views so let's hop up into our core folder and create some new folders for that so we're going to do command shift n command shift n command shift n and get your components folder above all that okay so this first group is going to be called uh we can call this explore this next one is going to be called notifications and this next one is going to be called uh messages and let's actually just go ahead and make one more guys and we're going to call it profile so let me get that above there and that looks pretty good so um inside of these guys uh we're the

Second subtitle

tài sản vì vậy nó sẽ là thẻ self.selected self.selectedIndexes1 là một hai hai ba ba và chúng ta hãy tiếp tục và tạo ra các lượt xem tab uh khác vì vậy ngay bây giờ chúng tôi chỉ sử dụng lại chế độ xem nguồn cấp dữ liệu lần chúng tôi không thực sự muốn làm điều đó chúng tôi muốn tạo các chế độ xem khác của chúng tôi vì vậy hãy nhảy vào thư mục cốt lõi của chúng tôi và tạo một số thư mục mới cho điều đó chúng ta sẽ thực hiện lệnh shift n lệnh shift n lệnh shift n và lấy thư mục thành phần của bạn trên tất cả cái đó được rồi, nhóm đầu tiên này sẽ là được gọi là uh chúng ta có thể gọi đây là khám phá cái tiếp theo này sẽ được gọi là thông báo và cái tiếp theo này sẽ được gọi là tin nhắn uh và chúng ta hãy thực sự tiếp tục và kiểm thêm một người nữa và chúng ta sẽ gọi nó là hồ sơ vì vậy hãy để tôi lấy nó ở trên đó và điều đó có vẻ khá tốt Vì thế ô bên trong những kẻ này uh chúng ta là

Main subtitle

profile is not going to be a tab
we're going to get to the profile a
different way we're just going to have
these four so let's go to explore

and we're going to create a new file
it's going to be a swift ui view and

we're going to call this explorer view

and we can replace this text with like

uh

explore view

and let's do the same thing in

notification so

swift ui view

notifications view

replace this with

notifications

and then our messages guy we're going to

create a new view for that as well swift

ui view

messages view

we're going to call this messages

now let's go back to our main tab folder

and we're going to replace this with
explorer view

we're going to replace this with

notifications view

and we're going to replace this last guy
with messages view

so let's go ahead and hit resume

just to see if our tab

view is showing up

the way we want it to

Second subtitle

hồ sơ sẽ không phải là một tab
chúng ta sẽ đến hồ sơ a
theo cách khác chúng ta sẽ có
bốn điều này vì vậy chúng ta hãy đi khám
phá

và chúng ta sẽ tạo một tệp mới

nó sẽ là một giao diện người dùng nhanh
chóng và

chúng ta sẽ gọi đây là chế độ xem thám
hiểm

và chúng ta có thể thay thế văn bản này
bằng like

uh

khám phá xem

và hãy làm điều tương tự trong

thông báo như vậy

giao diện người dùng nhanh

xem thông báo

thay thế cái này bằng

thông báo

và sau đó anh chàng tin nhắn của chúng tôi,
chúng tôi sẽ

tạo một chế độ xem mới cho điều đó cũng
nhanh chóng

giao diện người dùng

tin nhắn xem

chúng ta sẽ gọi đây là tin nhắn

bây giờ hãy quay lại thư mục tab chính của
chúng tôi

và chúng ta sẽ thay thế cái này bằng
chế độ xem thám hiểm

chúng ta sẽ thay thế cái này bằng

xem thông báo

và chúng ta sẽ thay thế người cuối cùng này
với chế độ xem tin nhắn

vì vậy hãy tiếp tục và nhắn tiếp tục

chỉ để xem nếu tab của chúng tôi

chế độ xem đang hiển thị

theo cách chúng ta muốn

Main subtitle

and that should be it okay so that's looking really good we just need to adjust our images so this one is going

to be for the explorer it's going to be a magnifying glass

this guy is going to be a bell and this

guy is going to be an envelope so that's it um you guys notice that it's pretty easy to create your tab view there um and let's go ahead and hit play on the simulator and let's select a new tab and we can see that it's linked to a

different view

so that's looking really really good guys

um

once again

that selection happens based on when we

change this selected index property so it's a state property so when our when

that property changes it reconfigures our view with the updated state and that

updated state is going to be a new index

so when i select the search item selected index is now going to be changed to one and then it's going to go okay hey explore view is the in is the view that you have linked to index one

and then if i select notifications the selected index property is now two and

Second subtitle

và điều đó sẽ ổn thôi vì vậy đó là trông thực sự tốt, chúng ta chỉ cần điều chỉnh hình ảnh của chúng tôi để hình ảnh này sẽ diễn ra

dành cho nhà thám hiểm nó sẽ là kính lúp

anh chàng này sẽ là một cái chuông và cái này

chàng trai sẽ trở thành một phong bì vậy là xong ừm các bạn để ý đó thật dễ dàng để tạo chế độ xem tab của bạn có um và chúng ta hãy tiếp tục và nhấn play trên trình giả lập và hãy chọn một cái mới tab và chúng ta có thể thấy rằng nó được liên kết với một

góc nhìn khác

vì vậy điều đó trông thực sự rất tốt các bạn

ô

một lần nữa

sự lựa chọn đó xảy ra dựa trên thời điểm chúng ta

thay đổi thuộc tính chỉ mục đã chọn này để đó là tài sản của nhà nước vì vậy khi chúng tôi

thuộc tính đó thay đổi nó cấu hình lại quan điểm của chúng tôi với trạng thái được cập nhật và điều đó

trạng thái được cập nhật sẽ là một chỉ mục mới

vì vậy khi tôi chọn mục tìm kiếm chỉ mục đã chọn bây giờ sẽ là thay đổi thành một và sau đó nó sẽ đi được rồi, khám phá chế độ xem là trong là chế độ xem mà bạn đã liên kết với chỉ mục một

và sau đó nếu tôi chọn thông báo thuộc tính chỉ mục đã chọn hiện là hai và

Main subtitle

then it's like okay in your tab view
what view is linked to the index of two
and then it says hey that's your notifications view that's why we see notifications popping up there so the last thing for us to do in uh this chapter is to go ahead and make the main view of our app this main tab view so so let's just go ahead and replace this with main tab view and run the application to make sure that it's looking good on our simulator and we can see here that this is looking great so that's going to wrap it up for this section of the course guys in the next one we are going to go ahead and um get started with building out the profile so get excited for that all right guys so this is the profile view that we see in front of us here it's pretty much an exact replica of the actual twitter application we're going to start building this now so let's go ahead and jump right into it hop up to this profile folder that's within our core folder and let's actually go ahead and create two folders in here guys we're going to create one for view

Second subtitle

sau đó mọi thứ sẽ ổn trong chế độ xem tab của bạn chế độ xem nào được liên kết với chỉ mục của hai và sau đó nó nói này, đó là của bạn chế độ xem thông báo đó là lý do tại sao chúng tôi thấy thông báo bật lên ở đó để điều cuối cùng để chúng ta làm trong chương này là đi trước và làm cho chế độ xem chính của ứng dụng của chúng tôi, chế độ xem tab chính này Vì vậy, hãy tiếp tục và thay thế cái này với chế độ xem tab chính và chạy ứng dụng để đảm bảo rằng nó trông đẹp trên trình mô phỏng của chúng tôi và chúng ta có thể thấy ở đây rằng điều này đang tìm kiếm tuyệt vời vì vậy sẽ kết thúc nó cho cái này phần tiếp theo của khóa học một trong những chúng ta sẽ tiếp tục và um bắt đầu với xây dựng hồ sơ rất phán khích vì điều đó được rồi các bạn, đây là hồ sơ quan điểm mà chúng ta thấy trước mặt chúng ta ở đây nó gần như là một bản sao chính xác của ứng dụng twitter thực tế chúng ta sẽ để bắt đầu xây dựng cái này ngay bây giờ vì vậy hãy bắt đầu phía trước và nhảy ngay vào nó nhảy lên tới thư mục hồ sơ này nằm trong chúng tôi thư mục lõi và hãy thực sự tiếp tục và tạo hai thư mục ở đây nhé các bạn chúng ta sẽ tạo một cái để xem

Main subtitle

models

and one for

views

so within the views folder we're going to create a swift ui view and it's going

to be called profile view

and let's go ahead and start building it now first let's make sure our preview is

connected

and it's looking good so we see here guys that we're going to have just one big v stack okay it's a bunch of vertically stacked components

so we're going to make the parent container i like to call it a v stack and the first thing that we're going to have in this v stack is something called a z stack

so we have this background and on that background we have some things on top of

it so whenever you have things stacked

on top of each other you use a z stack for that so let's go ahead and create this z stack

we're going to say the alignment is dot bottom

leading

and here's our content we'll see where

this comes into play in a little bit okay

so we're going to say color

dot system blue and then we're going to

Second subtitle

người mẫu

và một cho

lượt xem

vì vậy trong thư mục lượt xem, chúng ta sẽ dễ tạo chế độ xem giao diện người dùng nhanh chóng và nó sẽ hoạt động

được gọi là xem hồ sơ

và hãy tiếp tục và bắt đầu xây dựng nó bây giờ, trước tiên hãy đảm bảo rằng bản xem trước của chúng tôi là

kết nối

và nó có vẻ tốt nên chúng tôi thấy ở đây các bạn rằng chúng ta sẽ chỉ có một ngăn xếp v lớn được rồi, đó là một loạt thành phần xếp chồng lên nhau theo chiều dọc

vì vậy chúng tôi sẽ làm cho cha mẹ container tôi muốn gọi nó là ngăn xếp v và điều đầu tiên mà chúng ta sẽ làm có trong ngăn xếp v này là một thứ gọi là một ngăn xếp z

vì vậy chúng tôi có nền này và trên đó nền tảng chúng tôi có một số thứ trên đầu trang

nó vì vậy bắt cứ khi nào bạn có những thứ xếp chồng lên nhau

chồng lên nhau, bạn sử dụng ngăn xếp z vì điều đó, hãy tiếp tục và tạo ra ngăn xếp z này

chúng ta sẽ nói căn chỉnh là dấu chấm đáy

hàng đầu

và đây là nội dung của chúng tôi, chúng tôi sẽ thấy nó

điều này đi vào chơi trong một chút được rồi

vì vậy chúng ta sẽ nói màu sắc

hệ thống chấm màu xanh lam và sau đó chúng ta sẽ

Main subtitle

say dot ignores safe area

so

that's going to make this whole background blue we don't exactly want

that right so we're going to give our z

stack a frame

so we're going to say frame we don't

need to give it a width

and the height is just going to be i think it's like let me check

um

oops

our height let's make it like 88

and we don't need the alignment

okay so you notice that it's all the way

down at the bottom here let's actually make it like 96 or something

um so

within this v stack just go ahead and add a spacer guys

and you guys can see that it pushes that up to the top and that's looking pretty good so that's like the background image of the twitter profile

okay so within this z stack we're going to have a circle we're going to replace this with an image later and

we're going to say dot frame

width and height of let's see

uh 72 by 72

and you guys notice that it gets shoved over to the bottom leading edge and

Second subtitle

nói dấu chấm bỏ qua khu vực an toàn

Vì thế

điều đó sẽ làm cho toàn bộ điều này

nền màu xanh chúng tôi không thực sự muốn

đúng vậy vì vậy chúng tôi sẽ cung cấp cho z của chúng tôi

xếp một khung

vì vậy chúng tôi sẽ nói khung chúng tôi không

cần phải cung cấp cho nó một chiều rộng và chiều cao sẽ là tôi

nghĩ rằng nó giống như để tôi kiểm tra ô

Ói

chiều cao của chúng ta hãy làm cho nó giống như 88

và chúng ta không cần căn chỉnh

được rồi để bạn nhận thấy rằng đó là tất cả các cách

ở dưới cùng ở đây chúng ta hãy thực sự làm cho nó giống như 96 hoặc một cái gì đó ừm vậy

trong ngăn xếp v này, hãy tiếp tục và thêm miếng đệm nhé mọi người

và các bạn có thể thấy rằng nó đã điều đó lên đến đỉnh và trông thật đẹp

tốt vì vậy nó giống như hình nền của hồ sơ twitter

được rồi, trong ngăn xếp z này, chúng ta sẽ để có một vòng tròn chúng ta sẽ thay thế cái này với một hình ảnh sau

Và

chúng ta sẽ nói khung chấm

chiều rộng và chiều cao của hãy xem

uh 72 x 72

và các bạn chú ý rằng nó bị đẩy qua cạnh dưới cùng và

Main subtitle

that's because of how we set our z stack

up so for example i could say dot top trailing and it would put that guy over here right so we want that on the bottom

leading edge

and that image is going to have some offset or

the circle rather so we're going to say offset and we're going to give it 16 by

12 of an offset

um

so that looks pretty good let's actually make this offset like 24 and that's how we can sort of get this profile image to be halfway in the background and halfway in the rest of the view right there so

that looks pretty good

um you guys notice we have this back button though

so we're actually going to wrap this circle within a v stack so command click

on that say embed in v stack and we're going to have a button here action label

and our label is going to be an image with system name arrow dot left

it's going to be dot resizable dot frame

width 20

by 16

no alignment

dot foreground color

dot white

Second subtitle

đó là do cách chúng tôi thiết lập ngăn xếp z của mình

up nên ví dụ tôi có thể nói dot top dấu vết và nó sẽ đưa anh chàng đó qua ở ngay đây vì vậy chúng tôi muốn điều đó ở phía dưới

cạnh hàng đầu

và hình ảnh đó sẽ có một số

bù đắp hoặc

vòng tròn thay vì vậy chúng ta sẽ nói

bù đắp và chúng tôi sẽ cung cấp cho nó 16 bởi

12 của một bù đắp

ô

Vì vậy, điều đó có vẻ khá tốt, hãy thực sự thực hiện phần bù này như 24 và đó là cách chúng ta có thể lấy hình ảnh hồ sơ này để được một nửa trong nền và một nửa trong phần còn lại của chế độ xem ngay tại đó

trông khá tốt

ừm các bạn để ý chúng tôi có cái này trở lại nút mặc dù

vì vậy chúng tôi thực sự sẽ bọc cái này khoanh tròn trong ngăn xếp v để nhấp vào lệnh

trên đó nói nhúng vào ngăn xếp v

và chúng ta sẽ có một nút ở đây nhãn hành động

và nhãn của chúng tôi sẽ là một hình ảnh với tên hệ thống dấu chấm mũi tên bên trái nó sẽ là khung chấm có thể thay đổi kích thước

chiều rộng 20

bởi 16

không căn chỉnh

chấm màu nền trước

chấm trắng

Main subtitle

dot offset and then we're gonna say 16 by 12.

and you guys will notice that that puts that arrow right there for us so that looks pretty good and that's just our back button right there um

so you guys can see here that we can just get go back to you know whatever

view we came from

so that looks pretty good right all right so

um

let's go ahead and continue guys so that this is our header

so i'm actually going to extract this view here

so i'm going to go down below and create an extension

for profile view

and i'm going to say var header view sum view

and then i'm going to just take this z stack and put it in there

so cut it put it there and then we're

just going to replace that with our header view component and the reason we do this guys

is because

uh this view contains a lot of different

components and you don't want this like

main sort of body to get too crowded um so i like to whenever that's happening just create an extension here

Second subtitle

dấu chấm bù và sau đó chúng ta sẽ nói 16 bởi 12.

và các bạn sẽ nhận thấy rằng điều đó đặt mũi tên đó ngay tại đó cho chúng tôi để trông khá tốt và đó chỉ là của chúng tôi nút quay lại ngay đó um

vì vậy các bạn có thể thấy ở đây chúng ta có thể

chỉ cần quay trở lại với bạn biết bắt cứ điều gì

xem chúng tôi đến từ

vì vậy có vẻ khá tốt phải không được rồi

ô

hãy tiếp tục và tiếp tục các bạn để đây là tiêu đề của chúng tôi

vì vậy tôi thực sự sẽ giải nén cái này xem ở đây

vì vậy tôi sẽ đi xuống bên dưới và tạo một phần mở rộng để xem hồ sơ

và tôi sẽ nói chế độ xem tiêu đề var xem tổng

và sau đó tôi sẽ lấy cái z này ngăn xếp và đặt nó ở đó

vì vậy hãy cắt nó đặt nó ở đó và sau đó chúng tôi

chỉ cần thay thế nó bằng của chúng tôi thành phần xem tiêu đề và lý do chúng tôi làm điều này các bạn

là vì

uh chế độ xem này chứa rất nhiều thứ khác nhau

các thành phần và bạn không muốn điều này giống như

loại cơ thể chính để trả nên quá đông đúc ừm vì vậy tôi thích bắt cứ khi nào đó là xảy ra chỉ cần tạo một phần mở rộng ở đây

Main subtitle

and extract some of those components to like variables
and just to keep the main body nice and clean
okay so that looks good so far
next we're going to build these action buttons
so

those are going to be an h stack
and in order to get it all the way to the right of the screen we're just going to add a spacer here from the get go
we're going to say image system name bell.badge and then we're going to give it a font of title 3.

we're going to say dot padding is 6
and then we're going to give it an overlay of a circle
dot stroke which is color dot gray and then comma line with 0.75

so let's just go ahead and hit resume and that's going to give us this nice

notification bell image right there so that's looking pretty good now we just need to add this uh other button here so within our h stack we're gonna add

another button or just the first one was an image
we're not actually going to be doing anything with this
this next one is going to be a tappable button in this course so we're going to say

Second subtitle

và trích xuất một số thành phần đó để giống như các biến
và chỉ để giữ cho thân chính đẹp
và sạch sẽ

được rồi vì vậy có vẻ tốt cho đến nay tiếp theo chúng ta sẽ xây dựng những hành động này

nút

Vì thế

chúng sẽ là một ngăn xếp h và để có được nó tất cả các cách để bên phải màn hình chúng ta sẽ đi để thêm một miếng đệm ở đây ngay từ đầu chúng ta sẽ nói tên hệ thống hình ảnh bell.badge và sau đó chúng tôi sẽ đưa ra nó là một phông chữ của tiêu đề 3.

chúng ta sẽ nói phần đệm dấu chấm là 6 và sau đó chúng tôi sẽ cung cấp cho nó một lớp phủ của một vòng tròn nét chấm có màu chấm xám và sau đó dấu phẩy với 0,75

vì vậy hãy tiếp tục và nhấn tiếp tục và điều đó sẽ mang lại cho chúng ta điều tốt đẹp này

hình chuông thông báo ngay đó điều đó trông khá tốt bây giờ chúng tôi chỉ cần thêm nút uh other này vào đây vì vậy trong ngăn xếp h của chúng tôi, chúng tôi sẽ thêm

một nút khác hoặc chỉ cái đầu tiên là một hình ảnh chúng ta sẽ không thực sự làm bất cứ điều gì với điều này cái tiếp theo này sẽ có thể gõ được nút trong khóa học này vì vậy chúng tôi sẽ nói

Main subtitle

dot

button which is action and label

and

leave the action blank for now then
we're just going to go here and say text
edit profile

dot font dot sub headline

and then we're going to make it bold
and you guys can see here that our edit

profile button is starting to show up we
just need to get the uh

like the modifiers right to get the

shape okay so we're going to say dot

frame width is 120

height is going to be 32

and then we're going to give it a
background i don't think we need to give

it a background actually we're going to
say dot overlay

and this is going to be a rounded
rectangle

and we want to open it up

and give it a

corner radius of 20

and then we're going to give it a stroke
of color.gray

lined with

0.75

okay so that's looking pretty good and
then we just need to give it a

foreground color of dot black

okay so that's looking pretty good right

Second subtitle

dấu chấm

nút đó là hành động và nhãn

Và

để trống hành động bây giờ sau đó
chúng ta sẽ đi đến đây và nói văn bản
chỉnh sửa hồ sơ
chấm phông chữ chấm tiêu đề phụ
và sau đó chúng ta sẽ làm cho nó đậm
và các bạn có thể thấy ở đây chỉnh sửa của
chúng tôi

nút hồ sơ đang bắt đầu hiển thị, chúng tôi
chỉ cần nhận được uh

giống như công cụ sửa đổi quyền để có
được

hình dạng ổn vì vậy chúng ta sẽ nói dấu
chấm

chiều rộng khung là 120

chiều cao sẽ là 32

và sau đó chúng tôi sẽ cung cấp cho nó một
bối cảnh tôi không nghĩ chúng ta cần phải
đưa ra

nó là một nền tảng thực sự chúng ta sẽ
nói lớp phủ dấu chấm

và đây sẽ là một vòng tròn

hình chữ nhật

và chúng tôi muốn mở nó ra

và cho nó một

bán kính góc 20

và sau đó chúng tôi sẽ cho nó một đột quy
màu sắc. màu xám

thẳng đường với

0,75

được rồi, điều đó trông khá tốt và

sau đó chúng ta chỉ cần cung cấp cho nó
một

màu nền trước của chấm đen

được rồi vì vậy điều đó trông khá tốt phải
không

Main subtitle

there guys

um we just need to give it some padding
and give this h stack some spacing right

so

up at the top go ahead and give your h
stack some spacing make it like 12 and

that just gives some more space between
these buttons and then
on this guy you can say

uh

dot padding is dot trailing
and that'll just give it some padding on

that trailing edge right there so this
is starting to come together pretty
nicely

once again let's extract this view
component so let's just cut it
and below our header view

variable we're going to make a variable
called action buttons
some view and just paste that stuff in

there

and then here we're going to just say
action buttons
cool

okay so the next thing we need is this

like user info view right
so it's gonna be the user's username

or sorry their full name their username
and their bio
so

Second subtitle

có các bạn

ừm, chúng ta chỉ cần cho nó một chút đệm
và cung cấp cho ngăn xếp h này một số
khoảng cách ngay

Vì thế

lên đầu hãy tiếp tục và cho h của bạn
xếp chồng một số khoảng cách làm cho nó
giống như 12 và

điều đó chỉ mang lại thêm không gian giữa
các nút này và sau đó

về anh chàng này bạn có thể nói
uh

đệm dấu chấm là dấu chấm
và điều đó sẽ chỉ cung cấp cho nó một số
phần đệm trên

cạnh sau đó ngay tại đó nên cái này
đang bắt đầu đến với nhau khá đẹp
độc đáo

một lần nữa hãy trích xuất chế độ xem này
thành phần vì vậy chúng ta hãy cắt nó
và bên dưới chế độ xem tiêu đề của chúng
tôi

biết chúng ta sẽ tạo một biến
được gọi là các nút hành động
một số chế độ xem và chỉ cần dán nội dung
đó vào

ở đó
và sau đó ở đây chúng ta sẽ chỉ nói
nút hành động
mát mẻ

được rồi, điều tiếp theo chúng ta cần là cái
này
thích xem thông tin người dùng ngay
vì vậy nó sẽ là tên người dùng của người
dùng

hoặc xin lỗi tên đầy đủ của họ tên người
dùng của họ
và sinh học của họ
Vì thế

Main subtitle

let's go here
 and create a v stack for that
 um first of all hit resume
 and okay so our preview is still looking
 good
 and this v stack is going to have an
 alignment guys
 of dot leading
 the spacing is going to be 4 and then
 just hit enter on the content oops
 let me just delete that content guy
 actually okay
 all right so now let's add our text it's
 going to say heath ledger
 and okay so we see that it's showing up
 right there so let's actually go to the
 parent v stack
 and make the alignment
 dot leading there as well
 and we'll notice that it gets shoved all
 the way over to the left which is great
 so let's actually wrap this text in an h
 stack because we want to be able to have
 this nice like verified badge there as
 well so command click on the text
 properties say embed in h stack
 and then we're going to first give this
 a font of title 2 dot bold
 and then we're going to say image system
 name
 check mark dot seal dot fill

Second subtitle

chúng ta hãy đi đây
 và tạo một ngăn xếp v cho điều đó
 ừm trước hết hãy nhấn vào sơ yếu lý lịch
 và được rồi, bản xem trước của chúng tôi
 vẫn đang tìm kiếm
 Tốt
 và ngăn xếp v này sẽ có một
 kẻ thẳng hàng
 của dấu chấm hàng đầu
 khoảng cách sẽ là 4 và sau đó
 chỉ cần nhấn enter vào nội dung rất tiếc
 để tôi xóa thẳng nội dung đó đi
 thực sự ổn
 được rồi, bây giờ hãy thêm văn bản của
 chúng ta
 sẽ nói sở cái sức khỏe
 và được rồi, vì vậy chúng tôi thấy rằng nó
 đang hiển thị
 ngay tại đó, vì vậy chúng ta hãy thực sự đi
 đến
 ngăn xếp cha mẹ v
 và thực hiện căn chỉnh
 dấu chấm dẫn đến đó là tốt
 và chúng tôi sẽ nhận thấy rằng nó bị đẩy tất
 cả
 con đường bên trái thật tuyệt
 vì vậy chúng ta hãy thực sự bọc văn bản
 này trong một h
 ngăn xếp vì chúng tôi muốn có thể có
 điều này tốt đẹp như huy hiệu đã được xác
 minh ở đó như
 cũng vậy lệnh bấm vào văn bản
 thuộc tính nói nhúng trong ngăn xếp h
 và sau đó trước tiên chúng ta sẽ đưa cái
 này
 một phông chữ của tiêu đề đậm 2 chấm
 và sau đó chúng ta sẽ nói hệ thống hình ảnh
 tên
 dấu kiểm dấu chấm dấu chấm điền

Main subtitle

dot font

or sorry about foreground color
of color parentheses dot system blue
okay so that's looking good now below
that h stack guys we're going to create
a text component and we're going to just
say at joker it's going to be font

dot sub headline

dot foregroundcolor.gray

so that looks good as well and then we
need the bio so we're going to say text
your mom's favorite

villain

knee slapper

all right so we got your mom's favorite

villain

and then we're going to say

font.subheadline.padding

is

dot vertical

okay so that looks pretty good

now uh on this whole v stack guys we

just want to give it a um

a padding of leading or sorry dot

horizontal

so now it's going to give it some nice

padding on the edges there

and that's looking pretty good

so now we want these two components here

so we can keep that within this v stack

um we can just say

h stack

and

Second subtitle

phông chữ chấm

hoặc xin lỗi về màu nền trước
dấu ngoặc đơn màu xanh lam
được rồi, bây giờ nó trông ổn bên dưới
chúng ta sẽ tạo ra ngăn xếp h đó
một thành phần văn bản và chúng ta sẽ chỉ
nói tại joker nó sẽ là phông chữ
chấm tiêu đề phụ

dấu chấm foregroundcolor.grey

vì vậy có vẻ tốt là tốt và sau đó chúng tôi
cần tiêu sử vì vậy chúng tôi sẽ nói văn bản
yêu thích của mẹ bạn

nhân vật phản diện

cái tát đầu gối

được rồi vì vậy chúng tôi đã có yêu thích
của mẹ bạn

nhân vật phản diện

và sau đó chúng ta sẽ nói

font.subheadline.padding

là

chấm dọc

được rồi vì vậy trông khá tốt

bây giờ uh trên toàn bộ ngăn xếp v này,
chúng tôi

chỉ muốn cho nó một um

một phần đậm của dấu chấm hàng đầu
hoặc xin lỗi

nằm ngang

vì vậy bây giờ nó sẽ cho nó một số tốt đẹp

đem trên các cạnh ở đó

và điều đó trông khá tốt

vì vậy bây giờ chúng tôi muốn hai thành
phần này ở đây

vì vậy chúng tôi có thể giữ nó trong ngăn
xếp v này

ừm chúng ta chỉ có thể nói

ngăn xếp h

Và

Main subtitle

each one of these guys it's going to be its own h stack right so this is like an image with text and then image with text so within that h stack we're going to have two h stacks and then i'm going to say image system name think that's matte pin dot and dot ellipse so that looks good and then we're gonna say text and we could say gotham comma new york and then we actually want to give this a foreground color of gray so let's just go do that on the whole age stack.foregroundcolors.com and then i believe we're going to want to give it a font of dot caption yeah so that looks pretty good and then this h stack is going to be pretty much the same the image name is just going to be link and then here we can just replace this with like you know like www.thejoker.com or something and it recognizes that as a link which is pretty cool then we can say spacer to spread them out

Second subtitle

mỗi người trong số những người này nó sẽ được ngăn xếp h của chính nó, vì vậy đây giống như một hình ảnh với văn bản và sau đó hình ảnh với văn bản vì vậy trong ngăn xếp h đó, chúng ta sẽ có hai ngăn xếp h và sau đó tôi sẽ nói hệ thống hình ảnh tên nghĩ rằng đó là pin mờ dấu chấm và dấu chấm elip vì vậy điều đó có vẻ tốt và sau đó chúng ta sẽ nói chữ và chúng ta có thể nói gotham dấu phẩy new york và sau đó chúng tôi thực sự muốn cung cấp cho điều này một màu nền trước của màu xám vì vậy chúng ta hãy làm điều đó trên toàn bộ ngăn xếp tuổi.foregroundcolors.com và sau đó tôi tin rằng chúng ta sẽ muốn để cung cấp cho nó một phông chữ của chú thích dấu chấm vâng vì vậy nó có vẻ khá tốt và sau đó ngăn xếp h này sẽ là khá giống với tên hình ảnh chỉ là liên kết và sau đó ở đây chúng ta chỉ có thể thay thế cái này với như bạn biết giống www.thejoker.com hay gì đó và nó nhận ra đó là một liên kết khá tuyệt thì chúng ta có thể nói miếng đệm cho trải chúng ra

Main subtitle

and that's looking pretty good right
 there guys
 um
 but we actually don't know if we want
 that to be a spacer we can just go here
 and say spacing is like 32 or something
 and then on each of these the spacing
 looks good
 um let's actually make that like 24.
 right so that's our little um
 like location and link view and then we
 need this like user stats view right
 here
 so once again we can stay within this
 v stack that we're working within
 right
 and we're going to have another h stack
 very similar to this
 so it's going to be an h stack with two
 h stacks inside of it
 so the first one is they're just gonna
 be two text components we're gonna say
 text
 and we could say like you know like uh
 this is going to be
 following we could say like 807
 and then we're going to give that a font
 of weight of bold
 so you can see here that it bolds the
 first thing and then here we can go here

Second subtitle

và điều đó trông khá tốt phải không
 có các bạn
 ô
 nhưng chúng tôi thực sự không biết nếu
 chúng tôi muốn
 đó là một miếng đệm chúng ta chỉ có thể đi
 đến đây
 và nói khoảng cách giống như 32 hoặc một
 cái gì đó
 và sau đó trên mỗi khoảng cách này
 có vẻ tốt
 ừm, hãy thực sự làm cho nó giống như 24.
 đúng vậy đó là um nhỏ của chúng ta
 như vị trí và xem liên kết và sau đó chúng
 tôi
 cần cái này như chế độ xem thống kê người
 dùng đúng không
 đây
 vì vậy một lần nữa chúng ta có thể ở trong
 này
 ngăn xếp v mà chúng tôi đang làm việc
 trong đó
 Phải
 và chúng ta sẽ có một ngăn xếp h khác
 rất giống với điều này
 vì vậy nó sẽ là một ngăn xếp h với hai
 h ngăn xếp bên trong nó
 vì vậy điều đầu tiên là họ sẽ chỉ
 là hai thành phần văn bản chúng ta sẽ nói
 chữ
 và chúng ta có thể nói như bạn biết như uh
 đây sẽ là
 sau chúng ta có thể nói như 807
 và sau đó chúng tôi sẽ cung cấp cho phông
 chữ đó
 trọng lượng đậm
 vì vậy bạn có thể thấy ở đây nó in đậm
 điều đầu tiên và sau đó ở đây chúng ta có
 thể đi ở đây

Main subtitle

and say
following
and give this a font of dot caption

and we could give this a font of let me
see when i made the font really quickly

one second guys
okay this is going to be dot font of
dot sub headline
bold and the h stack is going to have a
spacing of four
and then it's going to have
this guy is going to be dot
foregroundcolor.gray that following text
is going to be gray so that looks pretty
good let's copy and paste that to the
next age stack and here let's do like

you know 6.9 million followers or
something
and change this to
followers
right so that looks pretty good and then
we're going to want to give this guy
some vertical padding as well
just to spread that out nicely so that's

looking pretty darn good
the next part is going to be this fancy
little
tweet
filter view
so let's go ahead and see if we can get
that built and then we're going to be

Second subtitle

và nói
tiếp theo
và cung cấp cho phông chữ này chú thích
dấu chấm
và chúng ta có thể cung cấp cho tôi một
phông chữ let me
xem khi tôi tạo phông chữ thực sự nhanh
chóng
một giây các bạn
được rồi, đây sẽ là phông chữ chấm của
chấm tiêu đề phụ
in đậm và ngăn xếp h sẽ có một
khoảng cách của bốn
và sau đó nó sẽ có
anh chàng này sẽ được chấm
foregroundcolor.gray văn bản theo sau
sẽ có màu xám nên trông rất đẹp
tốt, hãy sao chép và dán nó vào
ngăn xếp tuổi tiếp theo và ở đây chúng ta
hãy làm như thế nào
bạn biết 6,9 triệu người theo dõi hoặc
thứ gì đó
và thay đổi điều này để
người theo dõi
đúng như vậy trông khá tốt và sau đó
chúng tôi sẽ muốn cho anh chàng này
một số đệm dọc là tốt
chỉ để lan truyền nó ra một cách độc đáo để
đó là
trông khá darn tốt
phản tiếp theo sẽ rất thú vị
nhỏ bé
tiếng riu ríu
chế độ xem bộ lọc
vì vậy hãy tiếp tục và xem liệu chúng ta có
thể nhận được
được xây dựng và sau đó chúng ta sẽ trở
thành

Main subtitle

able to reuse our tweet cells to drop them into this guy right here so that's going to be pretty cool and then we're going to use a view model eventually to be able to switch between the different types of tweets so that's going to be pretty cool as well all right so really quickly let's clean this up a little bit more we are going to go here and take this whole z stat or v stack sorry cut it and let's go down here and call that var user info details some view paste it in there user info details and you guys will notice that nothing should change about your your previewer the way it looks right so this is looking pretty clean right you guys can see how much easier it is to work within your body when you don't have all this massive amount of code in there right okay so anyway let's just keep going guys uh this this filter bar view is somewhat complex but it's not too bad it's actually going to be based off of like a data model

Second subtitle

có thể sử dụng lại các ô tweet của chúng tôi để thả họ vào anh chàng này ngay tại đây vì vậy đó là sẽ khá tuyệt và sau đó chúng ta sẽ sử dụng một mô hình xem cuối cùng để có thể chuyển đổi giữa các loại tweet khác nhau vì vậy đó là cũng sẽ khá tuyệt được rồi rất nhanh hãy làm sạch cái này thêm một chút nữa chúng ta sắp vào đây và lấy toàn bộ z stat hoặc v ngăn xếp xin lỗi cắt nó và chúng ta hãy đi xuống đây và gọi người dùng var đó thông tin chi tiết một số xem dán nó vào đó chi tiết thông tin người dùng và các bạn sẽ nhận thấy rằng không có gì nên thay đổi về bạn trình xem trước của bạn đúng như vậy cái này trông khá sạch phải không bạn mọi người có thể thấy nó dễ dàng hơn bao nhiêu làm việc trong cơ thể của bạn khi bạn không có tất cả số lượng mã khổng lồ này trong ở đó Phải được rồi, dù sao thì chúng ta hãy tiếp tục các bạn uh cái này chế độ xem thanh bộ lọc này hơi phức tạp nhưng nó không quá tệ, nó thực sự đang diễn ra được dựa trên giống như một mô hình dữ

Main subtitle

though so i want to create that first
before we go any further so in
this view models

um folder let's go ahead and create a
new file it's just going to be a swift
file and we're going to call this tweet
filter view model

and this is actually going to be an
enumeration guys so we're gonna say enum
tweet filter viewmodel

it's gonna be an integer type enum and
it's gonna conform to this case iterable
protocol i always have so much trouble
saying that i'll explain why in a little

bit but we're going to use this enum to
create all of these
cases for our tweet filter bar so we're

going to say case tweets case replies
case likes

and then i want to create a title
property for this so i'm going to say
var

title which is a string just say switch
self

we're going to say case.tweets return
tweets

case

dot replies return

oops

replies

case dot likes return

likes

Second subtitle

liệu

mặc dù vậy tôi muốn tạo cái đó trước
trước khi chúng ta đi xa hơn nữa vì vậy
trong

mô hình xem này

thư mục um hãy tiếp tục và tạo một
tập tin mới sẽ diễn ra nhanh chóng
tập tin và chúng tôi sẽ gọi tweet này
mô hình chế độ xem bộ lọc

và đây thực sự sẽ là một

kể liệt kê vì vậy chúng ta sẽ nói enum
chế độ xem bộ lọc tweet

nó sẽ là một kiểu số nguyên enum và
nó sẽ phù hợp với trường hợp này iterable
giao thức tôi luôn gặp rất nhiều rắc rối
nói rằng tôi sẽ giải thích tại sao trong một
chút

một chút nhưng chúng ta sẽ sử dụng enum
này để

tạo ra tất cả những thứ này

các trường hợp cho thanh bộ lọc tweet của
chúng tôi, vì vậy chúng tôi

sẽ nói trường hợp tweets trường hợp trả lời
trường hợp thích

và sau đó tôi muốn tạo một tiêu đề
tài sản cho điều này vì vậy tôi sẽ nói
var

tiêu đề là một chuỗi chỉ cần nói switch
bản thân

chúng ta sẽ nói case.tweets trả lại
tweet

trường hợp

câu trả lời chấm trả lại

Ói

trả lời

trường hợp chấm thích trả lại
thích

Main subtitle

okay so this is our view model here now

let's go ahead and see how we're going

to use this view model inside of the profile view to actually create this view here

and some of the logic that's associated with it

so we're going to create an h stack guys

and then we're going to use a for each loop

and we're going to select this option that says data and content

and we're actually going to be able to use this view model to loop through all of the cases of our enum

to then create a view for each one of

those cases so i'm going to say tweet filter.viewmodel.all cases and the reason i can say all cases is because i

conformed to that case iterable iterable protocol

so what this does is it allows me to tweet or treat

all of my enum cases like an array so

you can actually do a for loop on it

which is pretty cool i love this feature so we're going to say all cases comma

oh just hit enter on the content and we

actually need to give this an id so

Second subtitle

được rồi, đây là mô hình xem của chúng tôi ở đây bây giờ

chúng ta hãy tiếp tục và xem chúng ta đang đi như thế nào

để sử dụng mô hình xem này bên trong chế độ xem hồ sơ để thực sự tạo cái này xem ở đây

và một số logic liên quan với nó

vì vậy chúng ta sẽ tạo một ngăn xếp h nhé các bạn

và sau đó chúng ta sẽ sử dụng a cho mỗi vòng

và chúng tôi sẽ chọn tùy chọn này nói rằng dữ liệu và nội dung

và chúng tôi thực sự sẽ có thể sử dụng mô hình xem này để lặp qua tất cả trong số các trường hợp của enum của chúng tôi

để sau đó tạo chế độ xem cho từng người trong số

những trường hợp đó vì vậy tôi sẽ nói tweet bộ lọc.viewmodel.all trường hợp và lý do tôi có thể nói tất cả các trường hợp là bởi vì tôi

phù hợp với trường hợp đó iterable iterable giao thức

vì vậy những gì nó làm là nó cho phép tôi tweet hoặc điều trị

tất cả các trường hợp enum của tôi giống như một mảng

bạn thực sự có thể thực hiện một vòng lặp for trên đó

nó khá tuyệt, tôi thích tính năng này

vì vậy chúng ta sẽ nói tất cả các trường hợp dấu phẩy

oh chỉ cần nhấn enter vào nội dung và chúng tôi

thực sự cần phải cung cấp cho cái này một id để

Main subtitle

we're going to say id backslash dot raw value
 and then within this guy we're going to create a v stack we're going to have text
 and we actually need to after these brackets say option or let's go ahead and say view model in or i would let's just say option actually or item rather so say item in and then for this text component we can just say item.title so now it's going to go through each of the cases and give us back the title for each one of those cases so it should be tweet replies and likes so let's go ahead and hit resume just to see if that's actually working and we can see that that's uh actually using our data model that we created to create those options for us which is pretty cool so far so uh i'm not going to go too heavy into this concept here i want to just keep going with the ui i will do a brief explanation of it after we're done though with this section so

Second subtitle

chúng ta sẽ nói id dấu gạch chéo ngược dot raw giá trị và sau đó trong anh chàng này, chúng ta sẽ tạo một ngăn xếp v chúng ta sẽ có chữ và chúng ta thực sự cần phải sau những điều này dấu ngoặc đơn tùy chọn hoặc hãy tiếp tục và nói xem mô hình trong hoặc tôi sẽ chỉ nói tùy chọn Thực ra hoặc mục đúng hơn là nói mục trong và sau đó đối với thành phần văn bản này, chúng ta chỉ có thể nói danh mục vậy bây giờ nó sẽ đi qua từng các trường hợp và trả lại cho chúng tôi tiêu đề cho mỗi một trong những trường hợp đó nên nó phải là trả lời tweet và lượt thích, vậy hãy bắt đầu phía trước và nhấn tiếp tục chỉ để xem nếu nó thực sự làm việc và chúng ta có thể thấy rằng đó thực sự là uh sử dụng mô hình dữ liệu của chúng tôi mà chúng tôi đã tạo để tạo các tùy chọn đó cho chúng tôi, đó là khá mát mẻ cho đến nay vì vậy uh tôi sẽ không đi sâu vào khái niệm này ở đây tôi chỉ muốn tiếp tục với ui tôi sẽ làm một lời giải thích ngắn gọn về nó sau chúng tôi đã hoàn thành mặc dù với phần này Vì thế

Main subtitle

up at the top guys i'm actually going to want to keep track of a state property so state private var selected

option or filter rather and it's going

to equal

or sorry be of type tweet filter view

model and we're going to default it to tweets

so this state property is going to keep track of my selected filter

just like that

so every time i tap on one it's going to update that state property and then ultimately will change like the tweets that i'm looking at but we're going to do that later

and then there's some logic based on what the selected property is right so like the selected one you guys notice that it's a bolded text it has this underlined view and the other guys are like grayed out and they don't have this

selector view but then if i select it

then that changes and this changes to

the selected uh like view components or modifiers

so let's go ahead and do that we're going to say font dot sub headline dot font weight

and then we're going to say selected filter equals item

yes

Second subtitle

lên ở trên cùng, tôi thực sự sẽ muốn theo dõi một tài sản nhà nước vì vậy nhà nước tư nhân var đã chọn

tùy chọn hoặc bộ lọc thay vào đó và nó sẽ diễn ra

bằng nhau

hoặc xin lỗi thuộc loại chế độ xem bộ lọc tweet

mô hình và chúng tôi sẽ mặc định nó để tweet

vì vậy tài sản nhà nước này sẽ giữ theo dõi bộ lọc đã chọn của tôi cứ thế

vì vậy mỗi khi tôi nhấn vào một cái thì nó sẽ cập nhật thuộc tính trạng thái đó và sau đó cuối cùng sẽ thay đổi như các tweet mà tôi đang xem nhưng chúng ta sẽ làm điều đó sau

và sau đó có một số logic dựa trên tài sản được chọn là gì vậy like cái đã chọn các bạn để ý rằng đó là một văn bản in đậm nó có cái này gạch dưới xem và những kẻ khác là như chuyển sang màu xám và họ không có cái này

chế độ xem bộ chọn nhưng sau đó nếu tôi chọn nó

sau đó điều đó thay đổi và điều này thay đổi thành

uh đã chọn như các thành phần xem hoặc sửa đổi

vì vậy hãy tiếp tục và làm điều đó chúng ta sẽ nói phông chữ chấm tiêu đề phụ trọng lượng phông chữ chấm

và sau đó chúng ta sẽ nói đã chọn bộ lọc bằng mục

Đúng

Main subtitle

dot semi bold no dot regular
 so this is some logic here that's going
 to determine our
 um
 font weight right so you guys notice
 that the selected font weight is bold
 the rest of them are not so this is
 saying hey
 if the selected filter is equal to the
 particular item that we're looking at in
 the for loop then yes we're going to
 make it bold no we're going to make it
 regular so that's pretty cool right

there

then we're going to say dot
 foreground color and this is also going

to be based on the selected filter
 selected filter equals item yes dot
 black

oops

no dot gray

okay then on this v stack guys we're
 going to say on tap gesture perform
 with animation

dot ease in out

self dot selected filter equals item
 so just go ahead and build that
 and make sure your build succeeds so

we're looking good now let's go ahead

and add the rest of the stuff to this so
 we're going to say if selected filter
 equals item

then we're gonna have a capsule

Second subtitle

chấm nửa đậm không chấm thường
 vì vậy đây là một số logic ở đây sẽ diễn ra
 để xác định của chúng tôi
 ô
 font chữ cân phải nên các bạn để ý
 rằng trọng lượng phông chữ đã chọn là đậm
 phần còn lại của họ thì không nên đây là
 nói xin chào
 nếu bộ lọc đã chọn bằng với
 mục cụ thể mà chúng tôi đang xem xét trong
 vòng lặp for thì vâng, chúng ta sẽ
 làm cho nó đậm không, chúng tôi sẽ làm
 cho nó
 thường xuyên nên điều đó khá tuyệt phải
 không
 ở đó
 sau đó chúng ta sẽ nói dấu chấm
 màu nền trước và điều này cũng đang diễn
 ra
 dựa trên bộ lọc đã chọn
 bộ lọc đã chọn bằng mục có dấu chấm
 đen
 Ôi
 không có chấm xám
 được rồi, trên ngăn xếp v này, chúng ta
 sẽ nói khi nhấn cử chỉ thực hiện
 với hoạt hình
 chấm dễ dàng trong ra
 tự chấm bộ lọc được chọn bằng mục
 vì vậy hãy tiếp tục và xây dựng nó
 và đảm bảo quá trình xây dựng của bạn
 thành công
 Bây giờ chúng tôi đang tìm kiếm tốt, hãy
 tiếp tục
 và thêm phần còn lại của những thứ này để
 chúng ta sẽ nói nếu bộ lọc được chọn
 bằng mục
 sau đó chúng ta sẽ có một viên nang

Main subtitle

dot foreground color
of color
dot

or parenthesis dot system blue
foreground color

right
okay

so now let's resume it
and

right now we've defaulted our

oh that's crazy our selected filter to
tweet so that's why we noticed this

massive view under the tweet the tweets
option we also noticed that this is

bolded and these guys are not this is

black and these guys are gray so that's

looking good we just need to change the
frame of this guy so we're going to say

frame

height is 3

and that's going to look much better
right

and then we're going to say
here

in order to even this out we're going to
say else

and then we're just going to create this
capsule

again

but we're going to give it a clear color

for the other options

so you guys notice that the capsule is

Second subtitle

chấm màu nền trước
màu sắc
dấu chấm

hoặc hệ thống dấu ngoặc đơn màu xanh
màu nền

Phải
được rồi

vì vậy bây giờ chúng ta hãy tiếp tục nó
Và

ngay bây giờ chúng tôi đã mặc định của
chúng tôi

ồ thật điên rồ bộ lọc đã chọn của chúng tôi
tweet vì vậy đó là lý do tại sao chúng tôi
nhận thấy điều này

lượt xem lớn dưới tweet các tweet
tùy chọn chúng tôi cũng nhận thấy rằng đây
là

in đậm và những kẻ này không phải là đây
là

màu đen và những người này màu xám nên
đó là

nhìn tốt, chúng ta chỉ cần thay đổi
khung của anh chàng này vì vậy chúng tôi
sẽ nói

khung

chiều cao là 3

và điều đó sẽ trông tốt hơn nhiều

Phải

và sau đó chúng ta sẽ nói
đây

để giải quyết vấn đề này, chúng ta sẽ
nói khác đi

và sau đó chúng ta sẽ tạo ra cái này
viên con nhộng

lại

nhưng chúng tôi sẽ cung cấp cho nó một
màu sắc rõ ràng
cho các tùy chọn khác
vì vậy các bạn chú ý rằng viên nang là

Main subtitle

there for each option

but for the non-selected ones it's clear

and that's just sort of so we can get

everything on the same level because if
this capsule wasn't there these guys
would actually sit lower

right so i could comment that out

and it just messes everything up so like
giving everything a capsule and just
changing the background color just makes
everything a lot smoother there

okay so let's actually go ahead and hit
the play button and see if we can select
a filter

and you guys can see in the preview that

it's changing and that's really smooth
right

but we can make this a little bit

smoother you guys notice if i go from
tweet to likes it just kind of shows up
there whereas in the completed app

there's this nice like sliding animation
so we're going to use matched geometry

effects do that so

here just go ahead and say match
geometry effects id we can just make it

like filter

and then we need a namespace id so go up

to the top say at namespace bar
animation

and just pass in animation there

Second subtitle

ở đó cho mỗi tùy chọn

nhưng đối với những cái không được chọn
thì rõ ràng

và đó chỉ là một cách để chúng ta có thể
nhận được

mọi thứ ở cùng cấp độ bởi vì nếu
viên nang này không có những người này
thực sự sẽ ngồi thấp hơn

đúng để tôi có thể nhận xét điều đó

và nó chỉ làm mọi thứ rối tung lên như vậy
cho mọi thứ một viên nang và chỉ
thay đổi màu nền chỉ làm cho
mọi thứ mượt mà hơn rất nhiều ở đó
được rồi, hãy thực sự tiếp tục và đánh
nút phát và xem liệu chúng tôi có thể chọn
một bộ lọc

và các bạn có thể thấy trong bản xem trước
rằng

nó đang thay đổi và điều đó thực sự trơn tru
Phải

nhưng chúng ta có thể làm điều này một
chút

mượt hơn các bạn để ý nếu mình đi từ
tweet để thích nó chỉ hiển thị

ở đó trong khi trong ứng dụng đã hoàn
thành

có cái này đẹp như hoạt hình trượt
vì vậy chúng tôi sẽ sử dụng hình học phù
hợp

hiệu ứng làm điều đó như vậy

ở đây chỉ cần tiếp tục và nói trận đấu

id hiệu ứng hình học chúng ta có thể làm
cho nó

thích bộ lọc

và sau đó chúng tôi cần một id không gian
tên để đi lên

lên đầu nói ở thanh không gian tên
hoạt hình

và chỉ chuyển vào hoạt ảnh ở đó

Main subtitle

so now just go ahead and play that again
and let's go ahead and click on likes
huh

let me try again

okay so you guys notice now
that

it's sliding all the way over very
smoothly so that looks really really
good um one more thing you guys will
notice is that we do have like a divider
here so we're going to give this h stack
an overlay
of a divider
so we're going to say overlay
divider
then we're going to give it an offset of

x 0
y 16
and you guys will notice that that gives
us that nice divider there
um
and it just makes this look a little bit
cleaner and gives it more separation
so this is awesome guys that's looking
super super smooth that's coded really
cleanly
and uh you know our profile view is
really really starting to come together
here the last piece of the puzzle
is this list of tweets and guess what we
already have that and we're going to be
good coders here and reuse previously

Second subtitle

vì vậy bây giờ chỉ cần tiếp tục và chơi lại
và hãy tiếp tục và nhấp vào lượt thích
Huh
để tôi thử lại
được rồi để các bạn chú ý ngay bây giờ
cái đó
nó đang trượt hết cỡ
trơn tru để trông thực sự thực sự
tốt um một điều nữa các bạn sẽ
lưu ý là chúng tôi có giống như một dải
phân cách
ở đây vì vậy chúng tôi sẽ cung cấp ngăn
xếp h này
một lớp phủ
của một dải phân cách
vì vậy chúng ta sẽ nói lớp phủ
dải phân cách
sau đó chúng tôi sẽ cung cấp cho nó một
phản bù của

x 0
năm 16
và các bạn sẽ nhận thấy rằng điều đó mang
lại
chúng tôi mà dải phân cách tốt đẹp ở đó
ô
và nó chỉ làm cho điều này trông một chút
sạch hơn và cho nó tách biệt hơn
vì vậy đây là những người tuyệt vời đang
tìm kiếm
siêu siêu mượt được mã hóa thực sự
sạch sẽ
và uh bạn biết xem hồ sơ của chúng tôi là
thực sự thực sự bắt đầu đến với nhau
đây là mảnh ghép cuối cùng
đây là danh sách các tweet và đoán những
gì chúng tôi
đã có điều đó và chúng ta sẽ trở thành
lập trình viên giỏi ở đây và sử dụng lại trước
đây

Main subtitle

created components in order to get that guy to show up instead of recoding it all over again and having the same exact code in multiple places in our app and

basically being total newbie programmers we're not going to do that so

really quickly before we do that let's

go ahead and cut all this stuff out and create this uh

extracted like component

we're going to call it uh tweet filter bar

some view

and paste all that in there and go ahead here and say tweet filter bar

and then the last piece of the puzzle guys is

this uh scroll view so we're going to say scroll view

and then we're going to say lazy v stack and we can just do like a simple for

each loop zero we can do like up to nine

content um i believe we need to give it

an id backslash that self and we're just

gonna reuse that tweet row view just like that

oh yeah we need underscore in so let's go ahead and hit resume and we notice that that's showing up

Second subtitle

đã tạo các thành phần để có được điều đó anh chàng xuất hiện thay vì mã hóa lại một lần nữa và có cùng một chính xác mã ở nhiều nơi trong ứng dụng của chúng tôi và

về cơ bản là toàn bộ lập trình viên mới chúng tôi sẽ không làm điều đó

Vì thế

thực sự nhanh chóng trước khi chúng ta làm điều đó hãy

tiếp tục và cắt bỏ tất cả những thứ này ra và tạo cái này

uh

trích xuất như thành phần

chúng ta sẽ gọi nó là uh bộ lọc tweet quán ba

một số xem

và dán tất cả vào đó

và tiếp tục ở đây và nói bộ lọc tweet quán ba

và sau đó là mảnh ghép cuối cùng các bạn là

chế độ xem cuộn uh này vì vậy chúng tôi sẽ nói xem cuộn

và sau đó chúng ta sẽ nói lazy v stack và chúng ta chỉ có thể làm như một cách đơn giản cho

mỗi vòng lặp số 0 chúng ta có thể thực hiện tới chín

nội dung um tôi tin rằng chúng ta cần phải cung cấp cho nó

một dấu gạch chéo ngược id đó và chúng tôi chỉ

sẽ sử dụng lại chế độ xem hàng tweet đó cứ thế

Ồ vâng, chúng tôi cần gạch dưới trong vì vậy hãy tiếp tục và nhấn tiếp tục

và chúng tôi nhận thấy rằng điều đó đang hiển thị

Main subtitle

very nicely let's just go ahead and give our tweet row view some padding

awesome awesome awesome

so now you guys will notice that i can

actually

scroll through this list of tweets and that's looking super slick right so let's go ahead and just extract this view just like we did to the with the rest of our views here and that's gonna

wrap it up for our profile ui so i'm

gonna go down to the bottom here and call this tweets view some view

paste that in there

and then go back up to the top and say tweet view alright guys so that's going to wrap it up for the profile chapter or at least the ui for it in the next

part of this course we're going to be

building out this explore view so that we can search for users that exist in

the app and actually be able to click on them and have it take us over to their profile before we get into the rest of

the course so get excited for that guys

that's what's next let's do it

alright guys so let's get started with

this view here this should be pretty

Second subtitle

rất tuyệt, hãy tiếp tục và công hiến

hàng tweet của chúng tôi xem một số phần đệm

tuyệt vời tuyệt vời tuyệt vời

vì vậy bây giờ các bạn sẽ nhận thấy rằng tôi có thể

Thực ra

cuộn qua danh sách các tweet này và trông siêu bóng bẩy đúng như vậy hãy tiếp tục và giải nén cái này xem giống như chúng tôi đã làm với phần còn lại của quan điểm của chúng tôi ở đây và đó sẽ

kết thúc nó cho hồ sơ của chúng tôi ui vì vậy tôi

sẽ đi xuống phía dưới đây và gọi đây là chế độ xem tweet một số xem

dán cái đó vào đó

và sau đó quay trở lại đầu trang và nói tweet xem được rồi các bạn, vậy là xong để kết thúc nó cho chương hồ sơ

hoặc ít nhất là giao diện người dùng cho nó trong phần tiếp theo

một phần của khóa học này chúng ta sẽ trở thành

xây dựng chế độ xem khám phá này để chúng tôi có thể tìm kiếm người dùng tồn tại trong

ứng dụng và thực sự có thể nhấp vào họ và có nó đưa chúng ta đến với họ

hồ sơ cá nhân trước khi chúng tôi đi vào phần còn lại của

khóa học rất phán khích cho những người đó

đó là những gì tiếp theo chúng ta hãy làm điều đó

được rồi các bạn, vậy chúng ta hãy bắt đầu với

khung cảnh này ở đây sẽ rất đẹp

Main subtitle

quick so we're going to go into this explorer folder we already have an explorer view but we do want to create our two subfolders for views and view models so let's do that first views and then we're going to drag this explore view guy up into the view folder just like that awesome okay so the first thing we're going to need is or one of the things we're going to need is this sort of like user row view so we're going to put that in our components folder so i'm going to create a new group here call this users and i'm going to create a swift ui view and we're going to say user row view awesome so this is a pretty simple view guys it's just going to be an h stack and we're going to put a circle here for the image we're going to give it a frame of like i think it's going to be 48 by 48 we don't need the alignment and then we are going to within that h stack have a

Second subtitle

nhanh chóng vì vậy chúng ta sẽ đi sâu vào vấn đề này thư mục thám hiểm chúng tôi đã có một chế độ xem thám hiểm nhưng chúng tôi muốn tạo hai chúng ta các thư mục con để xem và xem các mô hình hãy làm điều đó trước lượt xem và sau đó chúng tôi sẽ kéo cái này khám phá xem thằng lén thành thư mục xem giống như tuyệt vời đó được rồi vì vậy điều đầu tiên chúng ta sẽ cần là hoặc một trong những thứ chúng ta sẽ cần đây có phải là kiểu xem hàng của người dùng không vậy chúng tôi sẽ đưa nó vào thư mục thành phần vì vậy tôi sẽ tạo một nhóm mới ở đây gọi người dùng này Và tôi sẽ tạo giao diện người dùng nhanh và chúng ta sẽ nói hàng người dùng xem tuyệt vời vì vậy đây là một cái nhìn khá đơn giản các bạn, nó sẽ chỉ là một ngăn xếp h và chúng ta sẽ đặt một vòng tròn ở đây cho hình ảnh chúng ta sẽ cung cấp cho nó một khung giống như tôi nghĩ nó sẽ là 48 bởi 48 chúng ta không cần sự liên kết và sau đó chúng tôi sẽ vào trong ngăn xếp h đó có một

Main subtitle

v stack let me just get my preview going here um and let's see we want to give that guy a foreground color too i think it defaults to black though so that's fine and then the v stack is going to be a text component that's just going to be like you know joker dot font dot sub headline and we're going to say dot bold and then text which is going to be heath ledger and then dot font dot sub headline and then it's going to be foreground color of dot gray and we want this v stack to be an alignment or have an alignment of dot leading right and then we also want to give it some spacing so let's say spacing is like four and then we want this to be all the way over to the left of our screen so we're gonna put a spacer um after that stuff to you know just fill out all that blank space and that looks pretty good to me then um we can just give this guy some

Second subtitle

v stack hãy để tôi xem bản xem trước của mình đây ừm và hãy xem chúng tôi muốn đưa ra điều đó anh chàng cũng có màu nền trước tôi nghĩ nó mặc định là màu đen Tốt rồi và sau đó ngăn xếp v sẽ là một thành phần văn bản sẽ trở thành như bạn biết joker chấm phông chữ chấm tiêu đề phụ và chúng ta sẽ nói dấu chấm đậm và sau đó chữ đó sẽ là số cái y tế và sau đó chấm phông chữ chấm tiêu đề phụ và sau đó nó sẽ là tiền cảnh màu chấm xám và chúng tôi muốn ngăn xếp v này là một căn chỉnh hoặc có sự liên kết của dấu chấm hàng đầu Phải và sau đó chúng tôi cũng muốn cung cấp cho nó một số khoảng cách vì vậy hãy nói khoảng cách giống như bốn và sau đó chúng tôi muốn điều này là tất cả các cách ở bên trái màn hình của chúng tôi để chúng tôi sẽ đặt một miếng đệm ừm sau những thứ đó để bạn biết điều vào tất cả chỗ trống đó và điều đó có vẻ khá tốt với tôi sau đó ừm, chúng ta có thể cho anh chàng này một ít

Main subtitle

padding uh

yeah just say padding

cool

um so that looks pretty good now we can

reuse this component back in our explore

view so let's hop back to our explorer

view

make sure our preview is connected

awesome

it's just going to be

a v stack

and then we're going to have uh

what is it called a scroll view with a

lazy v stack

we're gonna say four each zero up to

like 25 or something

id backslash dot self

open brackets underscore in

and guys we will be replacing this with

actual data as we go through the course

we just don't have that data yet

um so we need to actually connect the
back end using firestore which we're

going to do in a little bit

so let's just go ahead and resume on
that

and

let's actually pass in this user row

view

and see how that looks

okay so that's looking pretty good

Second subtitle

đêm uh

vâng chỉ cần nói phần đêm

mát mẻ

ừm để nó trông khá tốt bây giờ chúng ta có
thể

sử dụng lại thành phần này trong khám phá
của chúng tôi

xem vì vậy hãy quay lại nhà thám hiểm của
chúng tôi

xem

đảm bảo bản xem trước của chúng tôi được
kết nối

tuyệt vời

nó sẽ chỉ là

một ngăn xếp v

và sau đó chúng ta sẽ có uh

nó được gọi là chế độ xem cuộn với

ngăn xếp v lười biếng

chúng ta sẽ nói bốn mỗi số không cho đến

như 25 hay gì đó

id dấu gạch chéo ngược dấu chấm tự

mở ngoặc gạch dưới trong

và các bạn, chúng tôi sẽ thay thế cái này

bằng

dữ liệu thực tế khi chúng tôi trải qua khóa
học

chúng tôi chưa có dữ liệu đó

ừm vì vậy chúng ta cần thực sự kết nối

back end sử dụng firestore mà chúng tôi
đang

sẽ làm trong một chút

vì vậy chúng ta hãy tiếp tục và tiếp tục

cái đó

Và

hãy thực sự vượt qua hàng người
dùng này

xem

và xem nó trông như thế nào

được rồi vì vậy nó trông khá tốt

Main subtitle

um now guys what we want to do is actually make this something called a navigation link and it's going to have a destination and a label and the destination is going to be our profile view and the label is going to be our user row view just like that you guys will notice that it changed the color of everything so let's go back to the user row view and give this v stack uh a foreground color of dot black for the text and this guy will stay the same but i actually think it's all grayed out because we're not actually inside of a navigation view so i want us to do that as well because this won't actually work unless we're inside of a navigation view so what we can just do for now is wrap this whole thing inside of the navigation view and paste it all in there and you guys will notice that everything should now look just fine and we can actually give this like a

Second subtitle

ừm bây giờ những gì chúng ta muốn làm là thực sự làm cho cái này được gọi là liên kết điều hướng và nó sẽ có một điểm đến và một cái nhãn và đích đến sẽ là của chúng ta xem hồ sơ và nhãn sẽ là người dùng của chúng tôi xem hàng cứ thế các bạn sẽ nhận thấy rằng nó đã thay đổi màu sắc của mọi thứ vì vậy chúng ta hãy quay trở lại chế độ xem hàng của người dùng và cho ngăn xếp v này uh một màu nền trước của chấm đen cho chữ và anh chàng này sẽ giữ nguyên nhưng tôi thực sự nghĩ tất cả đều chuyển sang màu xám bởi vì chúng ta không thực sự bên trong chế độ xem điều hướng Vì thế tôi cũng muốn chúng ta làm điều đó bởi vì điều này sẽ không thực sự làm việc trừ khi chúng tôi đang ở trong chế độ xem điều hướng vì vậy những gì chúng ta có thể làm bây giờ là quấn toàn bộ điều này bên trong của chế độ xem điều hướng và dán tất cả vào đó và các bạn sẽ nhận thấy rằng mọi thứ bây giờ nên nhìn tốt và chúng ta thực sự có thể cho điều này giống như một

Main subtitle

navigation bar title as well so we can

say that navigation bar title
or just title is explore
and you guys will notice that our
navigation item comes up right there
and then we'll say dot navigation title

display mode and we can just say dot
inline

so now guys i'm going to click play on
this and click that and you guys will
notice that it takes me over to my
profile view
you guys will notice however that this

isn't doing anything when i click on
this button i have to like hold down and
swipe back

which isn't terrible but i do want to
fix that right now so let's go ahead and

do that so back in our profile view

i'm going to add something called an
environment variable here
so environment open parentheses
backslash dot presentation mode var mode

and let's scroll down to this header
view

and we have this button here for the our
arrow

we're going to say
mode.wrappedvalue.dismiss
and now let's hit resume on the preview

Second subtitle

tiêu đề thanh điều hướng cũng như để
chúng tôi có thể
nói rằng tiêu đề thanh điều hướng
hoặc chỉ tiêu đề là khám phá
và các bạn sẽ nhận thấy rằng chúng tôi
mục điều hướng xuất hiện ngay tại đó
và sau đó chúng tôi sẽ nói tiêu đề điều
hướng dấu chấm
chế độ hiển thị và chúng ta chỉ có thể nói
dấu chấm
nội tuyến
vì vậy bây giờ các bạn tôi sẽ nhấp vào chơi
trên
cái này và nhấp vào cái đó và các bạn sẽ
lưu ý rằng nó đưa tôi đến chỗ của tôi
xem hồ sơ
Tuy nhiên, các bạn sẽ nhận thấy rằng điều
này
không làm bất cứ điều gì khi tôi nhấp vào
nút này tôi phải nhấn giữ và
vuốt lại
điều đó không tệ nhưng tôi muốn
khắc phục điều đó ngay bây giờ vì vậy hãy
tiếp tục và
làm điều đó để trở lại trong chế độ xem hồ
sơ của chúng tôi
tôi sẽ thêm một thứ gọi là
biến môi trường ở đây
vì vậy môi trường mở ngoặc đơn
chế độ trình bày dấu gạch chéo ngược chế
độ var
và hãy cuộn xuống tiêu đề này
xem
và chúng tôi có nút này ở đây cho
mũi tên
chúng ta sẽ nói
mode.wrappedvalue.dismiss
và bây giờ hãy nhấn tiếp tục trên bản xem
trước

Main subtitle

let's go back to our explore view

and let's click on one of these guys and hit this

and you guys will notice now that it's

taking me back

over to my original view

so really quickly i actually just want to fix some of this padding here these rows are a little bit too spread out and there's a lot of white space in there so

let's just hop back into this user row

view

and we're going to remove the padding guy or rather we're just going to add

padding dot horizontal

and then dot padding

vertical we're just gonna do like

four

and

let's go back to

our

what's it called explore view i think

yep and resume and let's just look at these updates so that looks a little bit better to me i also want a little bit more spacing between the picture and the user info

so let's just go and add some spacing to

our h stack so we're going to say

spacing and let's do like 12.

Second subtitle

hãy quay lại chế độ xem khám phá của chúng tôi

và hãy nhấp vào một trong những người này và

đánh cái này

và bây giờ các bạn sẽ nhận thấy rằng đó là đưa tôi trở lại

qua chế độ xem ban đầu của tôi

rất nhanh tôi thực sự chỉ muốn

để sửa một số phần đệm này ở đây

các hàng hơi quá dàn trải

và có rất nhiều khoảng trống trong

Bởi vậy

chúng ta hãy quay trở lại hàng người dùng này

xem

và chúng tôi sẽ loại bỏ phần đệm

anh chàng hay đúng hơn là chúng ta sẽ thêm

dấu chấm đệm nằm ngang

và sau đó chấm đệm

theo chiều dọc, chúng ta sẽ làm như thế nào

bốn

Và

chúng ta hãy quay trở lại

của chúng tôi

tôi nghĩ nó được gọi là chế độ xem khám phá

vâng và tiếp tục và chúng ta hãy nhìn vào những cập nhật này để có vẻ một chút tốt hơn với tôi tôi cũng muốn một chút thêm khoảng cách giữa các hình ảnh và thông tin người dùng

vì vậy chúng ta hãy tiếp tục và thêm một số khoảng cách vào

ngăn xếp h của chúng tôi vì vậy chúng tôi sẽ nói

khoảng cách và hãy làm như 12.

Main subtitle

okay so that looks perfect guys so
 that's going to wrap it up for
 that component for now that you sort of
 user list view later on we're going to
 add a search bar and make those users
 searchable
 um but next up there's a couple more ui
 components that we have to build guys
 we're going to be building out this
 awesome side menu that you see here next
 and then we still have to do this uh
 tweet upload a tweet view as well as our
 login and sign up pages so
 side menu's next you guys are doing
 great so far i hope you're enjoying this
 course and the ui is really starting to
 come together then we're going to start
 adding some data in our back end so get
 excited for that guys
 alright guys so this is what the side
 menu is going to look like we're going
 to start by just building out the ui for
 this view and then work on the
 presentation logic after we're done
 building the ui so go up to your core
 folder we're going to create a new group
 and we're going to call this side menu

Second subtitle

được rồi, trông thật hoàn hảo nhé mọi
 người
 điều đó sẽ kết thúc nó cho
 thành phần đó bây giờ mà bạn sắp xếp
 chế độ xem danh sách người dùng sau này
 chúng ta sẽ
 thêm một thanh tìm kiếm và làm cho những
 người dùng đó
 có thể tìm kiếm
 ừm nhưng tiếp theo sẽ có thêm vài cái nữa
 ui
 các thành phần mà chúng ta phải xây dựng
 các chàng trai
 chúng ta sẽ xây dựng cái này
 menu bên tuyệt vời mà bạn thấy ở đây tiếp
 theo
 và sau đó chúng ta vẫn phải làm điều này
 uh
 tweet tải lên một lượt xem tweet cũng như
 của chúng tôi
 trang đăng nhập và đăng ký
 menu phụ tiếp theo các bạn đang làm
 tuyệt vời cho đến nay tôi hy vọng bạn thích
 điều này
 khóa học và giao diện người dùng đang
 thực sự bắt đầu
 đến với nhau sau đó chúng ta sẽ bắt đầu
 thêm một số dữ liệu vào phần cuối của
 chúng tôi để có được
 phần khích cho các chàng trai đó
 được rồi các bạn vì vậy đây là những gì bên
 thực đơn sẽ giống như chúng ta đang đi
 để bắt đầu bằng cách xây dựng giao diện
 người dùng cho
 chế độ xem này và sau đó làm việc trên
 logic trình bày sau khi chúng tôi hoàn thành
 xây dựng giao diện người dùng, vì vậy hãy
 đi đến cốt lõi của bạn
 thư mục chúng ta sẽ tạo một nhóm mới
 và chúng ta sẽ gọi menu phụ này

Main subtitle

and i'm going to put the components folder back above that and then here we're going to create two sub groups one for view model and one for views okay so we're gonna create our side menu view within this view folder so create a new file select swift ui view and we're going to call this side menu view and we're also going to create our side menu row view so go ahead and create another file called side menu row view and that's just going to be the like sort of row for each menu option and then we're also going to go here and create our side menu view model so that's going to be a swift file we're going to call it side menu view model and let's actually just go ahead and build the data model out first and it's going to be another enumeration um and here i'm going to explain the reason why i like enumeration so much for this guys but let's just go ahead and build it first we're going to call it side menu view model once again it's an integer enum and it conforms to the case

Second subtitle

và tôi sẽ đặt các thành phần thư mục phía trên đó và sau đó ở đây chúng ta sẽ tạo hai nhóm phụ một để xem mô hình và một cho lượt xem được thôi, nếu vậy chúng ta sẽ tạo chế độ xem menu bên trong thư mục chế độ xem này, vì vậy hãy tạo một thư mục mới tập tin chọn swift ui xem và chúng ta sẽ gọi menu phụ này xem và chúng tôi cũng sẽ tạo ra phe của chúng tôi chế độ xem hàng menu vì vậy hãy tiếp tục và tạo một tệp khác được gọi là chế độ xem hàng menu bên và điều đó sẽ giống như sắp xếp hàng cho mỗi tùy chọn menu và sau đó chúng ta cũng sẽ đi đến đây và tạo mô hình xem menu bên của chúng tôi để đó sẽ là một tập tin nhanh chóng chúng ta sẽ gọi nó là chế độ xem menu bên người mẫu và chúng ta hãy thực sự tiếp tục và xây dựng mô hình dữ liệu đầu tiên và nó sẽ là một phép liệt kê khác ô và ở đây tôi sẽ giải thích lý do tại sao tôi thích liệt kê rất nhiều cho điều này các bạn nhưng chúng ta hãy tiếp tục và xây dựng nó đầu tiên chúng ta sẽ gọi nó là bên thực đơn xem mô hình một lần nữa nó là một số nguyên enum và nó phù hợp với trường hợp

Main subtitle

interable protocol right so we're going to say case profile pretty much just mimic all these options you see here

case uh lists case bookmarks

and case logout

and then we're going to create a description

called it's going to be a string we're going to say switch south and here is just where we name

or give each option an actual string

title

so case profile

case dot lists

return lists i actually have to say

return up here

and then we're going to say

case.bookmarks return

book marks

and then case.log out return log out okay and one more thing guys we notice

that we also have an image associated

with each option so we're going to add that image name property to our view

model here as well

and see how we're going to be able to use that later so we're going to say var image name which is a string and we can pretty much just copy this switch statement and we're just going to change up the strings that we return so the first one is going to be person

Second subtitle

giao thức tương thích, vì vậy chúng tôi sẽ dễ dàng nói hồ sơ trường hợp khá nhiều chỉ bắt chước tất cả các tùy chọn bạn thấy ở đây

trường hợp uh liệt kê các dấu trang trường hợp

và trường hợp đăng xuất

và sau đó chúng ta sẽ tạo ra một

Sự miêu tả

được gọi là nó sẽ là một chuỗi chúng ta sẽ nói chuyển về phía nam và đây là chỉ là nơi chúng tôi đặt tên

hoặc cung cấp cho mỗi tùy chọn một chuỗi thực tế

tiêu đề

hồ sơ vụ án

trường hợp dot danh sách

trả lại danh sách tôi thực sự phải nói

trở lại đây

và sau đó chúng ta sẽ nói

case.bookmarks trả lại

danh sách

và sau đó case.log out quay lại đăng xuất

được rồi và một điều nữa các bạn chúng tôi lưu ý

rằng chúng tôi cũng có một hình ảnh liên quan

với mỗi tùy chọn vì vậy chúng tôi sẽ thêm thuộc tính tên hình ảnh đó vào chế độ xem của chúng tôi

người mẫu ở đây cũng vậy

và xem làm thế nào chúng ta sẽ có thể

sử dụng nó sau vì vậy chúng ta sẽ nói var tên hình ảnh là một chuỗi và chúng ta có thể gần như chỉ cần sao chép công tắc này

tuyên bố và chúng ta sẽ thay đổi

lên các chuỗi mà chúng tôi trả lại

vì vậy người đầu tiên sẽ là người

Main subtitle

next one list dot bullet

bookmarks is going to be bookmark
and then this one is going to be
arrow.left
dot square
cool
okay so let's go ahead and use this enum
um or actually rather just talk about
how it's used now we'll talk about it

later i promise

let's go ahead and start building out
this view really fast

so we're going to go to our side menu
view

and we want to start building out this

header first and then we're going to use

that data model to create all these

options here

so

first off guys we're going to uh make
this a v stack

and let me just get my code opened up
here

okay so first thing uh just create a
circle for our you know profile image

there

and let's get this preview connected
a little sip of tea for me

no beer this time

so we got our circle let's give it a
frame

Second subtitle

một danh sách tiếp theo dấu chấm đầu
dòng
dấu trang sẽ là dấu trang
và sau đó cái này sẽ là
mũi tên trái
chấm vuông
mát mẻ
được rồi, hãy tiếp tục và sử dụng enum này
um hoặc thực sự chỉ nói về
làm thế nào nó được sử dụng bây giờ
chúng ta sẽ nói về nó
sau này tôi hứa
hãy tiếp tục và bắt đầu xây dựng
lượt xem này thực sự nhanh
vì vậy chúng ta sẽ đi đến menu phụ của
chúng ta
xem
và chúng tôi muốn bắt đầu xây dựng điều
này
tiêu đề đầu tiên và sau đó chúng tôi sẽ sử
dụng
mô hình dữ liệu đó để tạo ra tất cả những
thứ này
tùy chọn ở đây
Vì thế
trước tiên chúng ta sẽ làm
đây là ngăn xếp v
và để tôi mở mã của mình
đây
được rồi, điều đầu tiên uh chỉ cần tạo một
vòng tròn cho hình ảnh hồ sơ bạn biết của
chúng tôi
ở đó
và hãy kết nối bản xem trước này
một chút trà cho tôi
lần này không có bia
vì vậy chúng tôi đã có vòng kết nối của
mình, hãy cho nó một
khung

Main subtitle

width and height 48 by 48
and we don't need that
okay
all right so then we're going to say so
we're going to have some text it's going
to be
bruce wayne
dot font dot uh
let's give it a font of headline
yep that looks good
then we're going to below that we're
going to say text at batman
and we're going to give this a font of
we can just go ahead and do caption i
think
oh cannot find oh dot caption cool
so that doesn't look too bad um
this v stack
is what we've got right here we want it
to be pushed over
uh to
the left
so we're going to make this alignment
dot leading that's not going to do
anything right now
or it is but we want it to eventually be
all the way over here we're going to
work on that in a second
first off let's give this a foreground
color of dot gray
for the username guy right there so
that's looking good

Second subtitle

chiều rộng và chiều cao 48 x 48
và chúng tôi không cần điều đó
được rồi
được rồi, vậy thì chúng ta sẽ nói như vậy
chúng tôi sẽ có một số văn bản nó sẽ đi
được
Lý Tiểu Long
phông chữ chấm chấm uh
hãy cung cấp cho nó một phông chữ tiêu đề
vâng cái đó có vẻ tốt
sau đó chúng ta sẽ đến bên dưới mà chúng
ta
sẽ nói văn bản tại batman
và chúng tôi sẽ cung cấp cho phông chữ
này
chúng ta có thể tiếp tục và tạo chú thích tôi
nghĩ
ò không thể tìm thấy chú thích dấu chấm
tuyệt vời
vậy trông cũng không tệ lắm ừm
ngăn xếp v này
là những gì chúng tôi có ngay tại đây, chúng
tôi muốn nó
bị đẩy qua
uh để
Bên trái
vì vậy chúng ta sẽ thực hiện sự liên kết này
dấu chấm hàng đầu sẽ không làm được
bất cứ điều gì ngay bây giờ
hoặc nó là nhưng chúng tôi muốn nó cuối
cùng là
tất cả các cách ở đây chúng ta sẽ
làm việc trên đó trong một giây
trước hết chúng ta hãy cung cấp cho nó một
tiền cảnh
màu chấm xám
cho anh chàng tên người dùng ngay tại đó
trông thật tuyệt

Main subtitle

and i'm actually going to wrap this inside of its own v stack guys because i want our my own set of spacing

on that

so i'm going to say spacing is like 4 so that looks good and then copy this alignment

and now that's looking pretty good and you guys notice that we have this same like stat view that we had back in

our profile view

so really quickly i'm actually going to create a reusable component for this

um so if you guys hop back to the profile and go to views profile view um let's see let's find that it should be

this guy right here

right so let's actually go ahead and create a reusable component for this

so we have a users section already so

let's go ahead and create a new file it's going to be a swift ui view

and we're going to call this user stats

view

and let's create that and just go back and we're going to cut all this or copy it

user stats view

go and paste that in there

and now what we can do guys is go back

Second subtitle

và tôi thực sự sẽ quấn cái này bên trong ngăn xếp v của chính nó bởi vì tôi muốn tập hợp khoảng cách của riêng mình

trong vấn đề này

vì vậy tôi sẽ nói khoảng cách giống như 4 để có vẻ tốt và sau đó sao chép này cẩn chỉnh

và bây giờ điều đó trông khá tốt

và các bạn để ý rằng chúng ta có cái này giống như chế độ xem thống kê mà chúng tôi đã quay lại

xem hồ sơ của chúng tôi

rất nhanh, tôi thực sự sẽ

tạo một thành phần có thể tái sử dụng cho việc này

ừm vậy nếu các bạn quay trở lại

hồ sơ và đi đến lượt xem xem hồ sơ

ừm hãy xem chúng ta thấy rằng nó nên là

anh chàng này ngay tại đây

đúng vậy, hãy thực sự tiếp tục và

tạo một thành phần có thể tái sử dụng cho việc này

vì vậy chúng tôi đã có một phần người dùng rồi

hãy tiếp tục và tạo một tệp mới

nó sẽ là một giao diện người dùng nhanh chóng

và chúng tôi sẽ gọi số liệu thống kê người dùng này

xem

và hãy tạo nó và chỉ cần quay lại

và chúng tôi sẽ cắt tất cả hoặc sao chép Nó

chế độ xem thống kê người dùng

đi và dán nó vào đó

và bây giờ những gì chúng ta có thể làm là quay trở lại

Main subtitle

to the side or sorry profile view
and let's actually just replace this
with our user stats view

and now we can go back to our side menu
view

and add in
that guy as well user stats view

and eventually we're going to make this
so that it's populated with real data
um but for now this is just sort of
dummy data

and we notice that that looks pretty
good

right we just want uh some spacing on

that guy so let's actually go ahead and
just give it some padding of like dot

vertical

right so that's looking pretty good
um and we can actually leave this as is

for now

um what we're going to do next is
actually build out this list so
let's go ahead and do that
i want us to wrap this v stack inside of

another v stack so go ahead and say
embed in v stack

and

that way this v stack acts as it's like

own container right we can see that

that's sort of like our header view

Second subtitle

sang một bên hoặc xin lỗi xem hồ sơ
và hãy thực sự thay thế cái này
với chế độ xem thống kê người dùng của
chúng tôi

và bây giờ chúng ta có thể quay lại menu
phụ của mình

xem

và thêm vào

anh chàng đó cũng xem số liệu thống kê
người dùng

và cuối cùng chúng ta sẽ làm điều này
để nó được phổ biến với dữ liệu thực
tử um nhưng bây giờ đây chỉ là một loại
dữ liệu giả

và chúng tôi nhận thấy rằng nó trông đẹp
Tốt

đúng rồi, chúng tôi chỉ muốn uh một số
khoảng cách trên

anh chàng đó vì vậy hãy thực sự tiếp tục và
chỉ cần cung cấp cho nó một số phần đệm
giống như dấu chấm

thẳng đứng

đúng như vậy trông khá tốt

um và chúng ta thực sự có thể để nó như
vậy

bây giờ

um những gì chúng ta sẽ làm tiếp theo là
thực sự xây dựng danh sách này vì vậy
hãy tiếp tục và làm điều đó

tôi muốn chúng ta bọc ngăn xếp v này bên
trong

ngăn xếp v khác vì vậy hãy tiếp tục và nói
nhưng vào ngăn xếp v

Và

theo cách đó ngăn xếp v này hoạt động
giống như

vùng chứa riêng, chúng ta có thể thấy điều
đó

điều đó giống như chế độ xem tiêu đề của
chúng tôi

Main subtitle

and now down here

we're going to create the actual like profile option list so we're going to

say for each and this is where our data

model comes in handy we're going to say side menu view model about all cases

oops

and go ahead and hit enter on that content

and don't forget to actually add your id backslash dot raw value

and here we can just say like option in

okay and this should actually be side menu view model that's my bad then within this guide we are going to create an h stack and that's going to be what each of these option views are so we're

going to create an image say system name and we're going to pass in the option.imagename

and then we're going to pass in a text

and we're going to pass in the option.title

you guys might have that as description i changed it to title i think that's a little bit easier to remember so if you want you guys can just go ahead and change the name of this property to title that way we have text option.title

and we can see here that this is like

the beginning of what our um

Second subtitle

và bây giờ xuống đây

chúng ta sẽ tạo ra thực tế như

danh sách tùy chọn hồ sơ vì vậy chúng tôi sẽ

nói cho mỗi và đây là nơi dữ liệu của chúng tôi

mô hình có ích chúng ta sẽ nói

menu bên xem mô hình về tất cả các trường hợp

Ói

và tiếp tục và nhấn enter vào đó nội dung

và dừng quên thực sự thêm id của bạn

dấu gạch chéo ngược dấu chấm giá trị thô

và ở đây chúng ta chỉ có thể nói tùy chọn like trong

được rồi và điều này thực sự nên được phụ mô hình xem menu đó là xấu của tôi sau đó trong hướng dẫn này, chúng tôi sẽ tạo một ngăn xếp h và đó sẽ là những gì mỗi chế độ xem tùy chọn này là vì vậy chúng tôi

sẽ tạo một hình ảnh nói tên hệ thống

và chúng ta sẽ vượt qua trong tùy chọn.tên hình ảnh

và sau đó chúng ta sẽ chuyển qua một văn bản

và chúng ta sẽ vượt qua trong tùy chọn.title

các bạn có thể có đó như mô tả

tôi đã đổi nó thành tiêu đề tôi nghĩ đó là một dễ nhớ hơn một chút vì vậy nếu bạn

muốn các bạn có thể tiếp tục và thay đổi tên của tài sản này để

tiêu đề theo cách đó chúng tôi có tùy chọn văn bản.title

và chúng ta có thể thấy ở đây rằng điều này giống như

sự khởi đầu của những gì um của chúng tôi

Main subtitle

our options should look like we're actually going to add a spacer here um to each one of them and hit resume and that's just going to move everything over to the left side of the screen for us which is going to be really nice okay and you notice it kind of lines everything up as well which is great this h stack guys we're going to give it a frame of height which is 40 to give everything a little bit more spacing so that's looking good and then we are going to give it some horizontal padding and then to get this guy over here we can just copy this leading alignment guy one more time and put that on the parent v stack right and then you notice that that gets pushed all the way over to the left as well and on that this guy right here this uh sort of header view v stack we're going to say padding dot leading and that's going to move that over nicely and then guys we need to get this moved

Second subtitle

các tùy chọn của chúng tôi sẽ giống như chúng tôi thực sự sẽ thêm một miếng đệm ở đây ừm với mỗi người trong số họ và nhấn tiếp tục và điều đó sẽ xảy ra di chuyển mọi thứ sang phía bên trái của màn hình cho chúng tôi sẽ là thực sự tốt đẹp được rồi và bạn nhận thấy đó là loại đường mọi thứ đều tốt, điều đó thật tuyệt ngăn xếp h này, chúng tôi sẽ cung cấp cho nó một khung có chiều cao là 40 để cung cấp mọi thứ thêm một chút khoảng cách như vậy trông thật tuyệt và sau đó chúng tôi sẽ cung cấp cho nó một số đệm ngang và sau đó để có được anh chàng này ở đây chúng ta chỉ có thể sao chép này anh chàng liên kết hàng đầu một lần nữa và đặt nó trên mẹ v ngăn xếp ngay và sau đó bạn chú ý mà được đầy tất cả các cách trên bên trái cũng vậy Và trong vấn đề này anh chàng này ngay tại đây uh đây là loại xem tiêu đề v ngăn xếp chúng ta sẽ nói phần đệm dấu chấm hàng đầu và điều đó sẽ chuyển nó qua độc đáo và sau đó các bạn, chúng ta cần di chuyển cái này

Main subtitle

up all the way to the top of the screen
so on your uh
parent v stack just go ahead and add a
spacer and that's going to shove
everything up to the top
and that's looking really good
we do want a little bit more spacing

between these options and we're going to

change the font and color a little bit
of them as well just to make it look a

little bit cleaner so the image we're
going to say dot font is dot headline
and that's going to make the image a

little nicer then we're going to say dot

foreground color

is dot gray

and then the text we're going to say dot

font dot sub headline

and that looks a little better

okay now

on the h stack guys go ahead and give it

a spacing parameter of 16.

and we're gonna add some padding to this

for each guy right here so go to the end

of that and say dot padding dot vertical

ooh uh

make that like four

yeah so that spaces our options out a

little bit better

and then on this v stack here the

Second subtitle

lên đến đỉnh màn hình

vậy bạn à

ngăn xếp cha v chỉ cần tiếp tục và thêm một
miếng đệm và điều đó sẽ xô đẩy

mọi thứ lên đến đỉnh

và điều đó trông thực sự tốt

chúng tôi muốn có thêm một chút khoảng
cách

giữa các tùy chọn này và chúng ta sẽ
thay đổi phông chữ và màu sắc một chút
của họ cũng như chỉ để làm cho nó trông
giống như một

sạch hơn một chút để hình ảnh chúng ta
sẽ nói dấu chấm là tiêu đề dấu chấm
và điều đó sẽ làm cho hình ảnh trở thành
một

đẹp hơn một chút thì chúng ta sẽ nói dấu
chấm

màu nền

là chấm xám

và sau đó là văn bản chúng ta sẽ nói dấu
chấm

phông chữ chấm tiêu đề phụ

và điều đó có vẻ tốt hơn một chút

được rồi

trên ngăn xếp h mọi người cứ tiếp tục và
cho nó

tham số khoảng cách là 16.

và chúng tôi sẽ thêm một số phần đệm vào
đây

cho mỗi chàng trai ngay tại đây vì vậy hãy đi
đến cuối

của điều đó và nói dấu chấm đệm dấu chấm
dọc

ồ ồ

làm cho nó giống như bốn

vâng để loại bỏ các tùy chọn của chúng tôi
ra một

tốt hơn một chút

và sau đó trên ngăn xếp v này ở đây

Main subtitle

overall item let's see if we can add

spacing of like 32

and that's just gonna yeah so let's actually remove that vertical padding and that'll just give it some spacing

here between the the header and sort of the option list so that's looking really good

so that's pretty much it for the ui on

the side menu next we're going to work on how to actually show that side menu from our main view but before we do that

i did promise you guys i would give you

a more in-depth explanation as to why i like to use enumerations as the view model to create these menu options that we see here we also saw something

similar with the profile filter bar so let me go ahead and do and explain that

really quickly so one of the main

reasons i like to use enumerations for situations like this is how easy they are to modify so for example let's say i wanted to remove an option i just do that right here so i remove the bookmarks case i'm going to build my code and it's going to give me an error in all the places i need to modify my code in order for it to work again so luckily for me it's all in one place

Second subtitle

mục tổng thể hãy xem liệu chúng ta có thể thêm

khoảng cách như 32

và điều đó sẽ đúng, vì vậy chúng ta hãy thực sự loại bỏ phần đậm dọc đó

và điều đó sẽ chỉ cung cấp cho nó một số khoảng cách

ở đây giữa tiêu đề và sắp xếp

danh sách tùy chọn để trông thực sự

Tốt

vì vậy đó là khá nhiều cho giao diện người dùng trên

menu phụ tiếp theo chúng ta sẽ làm việc về cách thực sự hiển thị menu bên đó

từ chế độ xem chính của chúng tôi nhưng trước khi chúng tôi làm điều đó

tôi đã hứa với các bạn tôi sẽ cung cấp cho các bạn

một lời giải thích sâu hơn về lý do tại sao tôi thích sử dụng bảng liệt kê làm chế độ xem mô hình để tạo các tùy chọn menu này chúng tôi thấy ở đây chúng tôi cũng thấy một cái gì đó

tương tự với thanh bộ lọc hồ sơ như vậy hãy để tôi tiếp tục và làm và giải thích điều đó

thực sự nhanh chóng vì vậy một trong những điều chính

lý do tôi thích sử dụng liệt kê cho những tình huống như thế này thật dễ dàng là để sửa đổi, ví dụ, giả sử tôi

muốn xóa một tùy chọn tôi chỉ cần làm

ngay tại đây vì vậy tôi loại bỏ

trường hợp dấu trang tôi sẽ xây dựng

mã và nó sẽ báo lỗi cho tôi

ở tất cả những nơi tôi cần sửa đổi

mã để nó hoạt động trở lại

may mắn cho tôi là tất cả ở một nơi

Main subtitle

so now if i go back to my side menu view

and i resume on my preview we're going

to notice that the bookmark option just goes away and the list automatically adjusts itself to

have that updated list of options right so the bookmark option is just gone and adding it back is as simple as you know adding a case you know you guys could add any number of cases to this and once you add another case so let's

say i added like uh

i don't know

super likes or something i don't know right um you guys will notice that it says hey switch has to be exhaustive so you have to add the missing case and

it pretty much just gives you a single source of truth for your data model for

your ui that's why i really like

enumerations here and we also get these

awesome like computed properties that we

can use

that are based on each one of the cases that we have so that's how we were able

to configure our data here all done

through one place or one sort of data structure which is this enumeration and we saw that with the tweet filter

Second subtitle

vì vậy bây giờ nếu tôi quay lại chế độ xem menu bên của mình

và tôi tiếp tục bắn xem trước của mình, chúng tôi sẽ đi

để lưu ý rằng tùy chọn dấu trang chỉ biến mất và danh sách tự động tự điều chỉnh để

có danh sách tùy chọn được cập nhật ngay vì vậy tùy chọn dấu trang đã biến mất và thêm nó trở lại đơn giản như bạn biết thêm một trường hợp bạn biết các bạn có thể thêm bất kỳ số lượng trường hợp này và một khi bạn thêm một trường hợp khác, vì vậy hãy

nói rằng tôi đã thêm như uh

Tôi không biết

siêu thích hay gì đó tôi không biết đúng rồi, các bạn sẽ nhận thấy điều đó nó nói rằng công tắc này phải đây đủ vì vậy bạn phải thêm trường hợp còn thiếu và

nó khá nhiều chỉ cung cấp cho bạn một nguồn sự thật cho mô hình dữ liệu của bạn cho

ui của bạn đó là lý do tại sao tôi thực sự thích

bảng liệt kê ở đây và chúng tôi cũng nhận được những

tuyệt vời như các thuộc tính được tính toán mà chúng tôi

có thể sử dụng

dựa trên từng trường hợp

rằng chúng tôi có nên đó là cách chúng tôi có thể

để định cấu hình dữ liệu của chúng tôi ở đây đã hoàn tất

qua một nơi hoặc một loại dữ liệu

cấu trúc đó là kiểu liệt kê này

và chúng tôi đã thấy điều đó với bộ lọc tweet

Main subtitle

view model as well it would be really easy for you guys to change these options around or add or remove options to them so that's why i really like using enums for this type of stuff um and i think you guys should do the same whenever you can it's a lot better than just having like an array of strings right um you could have had an array of strings like profile list blah blah blah you could have created a structure for it but this is just what i like to do but anyway guys now that we've talked about that a little bit more and my promise has been fulfilled let's go ahead and go over to our main tab view and start configuring that to be able to show this side menu view to the user so that they can toggle it in and out as we see here in the completed app right looks pretty smooth so let's go ahead and do that now okay guys so we're actually going to hop into our app file and we're going to change this around a bit in order to get this to work so

Second subtitle

xem mô hình cũng như nó sẽ thực sự dễ dàng cho các bạn để thay đổi những tùy chọn xung quanh hoặc thêm hoặc xóa các tùy chọn cho chúng để đó là lý do tại sao tôi thực sự thích sử dụng enums cho loại công cụ này ừm và tôi nghĩ các bạn nên làm tương tự bắt cứ khi nào bạn có thể nó tốt hơn rất nhiều hơn là chỉ thích một mảng chuỗi đúng um bạn có thể đã có một mảng các chuỗi như danh sách hồ sơ blah blah blah bạn có thể đã tạo ra một cấu trúc cho nó nhưng đây chỉ là những gì tôi thích làm nhưng dù sao đi nữa các bạn bây giờ chúng ta đã nói về điều đó Thêm một chút nữa và lời hứa của tôi đã được thực hiện hãy tiếp tục và chuyển đến chế độ xem tab chính của chúng tôi và bắt đầu định cấu hình để có thể hiển thị điều này chế độ xem menu bên cho người dùng để họ có thể bật tắt nó như chúng ta thấy ở đây trong ứng dụng đã hoàn thành trông khá đẹp tron tru vì vậy hãy tiếp tục và làm điều đó ngay bây giờ được rồi các bạn, vì vậy chúng tôi thực sự sẽ nhảy vào tệp ứng dụng của chúng tôi và chúng tôi sẽ thay đổi điều này xung quanh một chút để có được cái này để làm việc như vậy

Main subtitle

go ahead and just delete main tab view

you guys notice here that we have a

navigation bar up at the top so

i'm actually going to

make the root view

of the application be contained within
this navigation view

and then we're actually going to make
the root view the content view of the
app and that's just this blank file here
and what we're going to do in this
content view is actually create a
z-stack

where we have the main tab and the side
menu sort of stacked on top of each
other inside of another view which is
going to be this content view and then
we're going to add some presentation
logic so that like when we hit this

button up at the top on the navigation
bar

that it'll bring that side menu into
view and they'll be arranged in a
z-stack where you can see that the side
menu is sort of on top of the main tab

view so that's how we're going to do
this

so anyway

let's go ahead and actually just run the
app really quickly

and it's just gonna be like this thing
that says hello world but we should

notice that we're inside of a navigation

Second subtitle

hãy tiếp tục và chỉ cần xóa chế độ xem tab
chính

các bạn chú ý ở đây rằng chúng ta có một
thanh điều hướng lên ở trên cùng
tôi thực sự sẽ

tạo chế độ xem gốc

của ứng dụng được chứa trong
chế độ xem điều hướng này

và sau đó chúng tôi thực sự sẽ làm
chế độ xem gốc chế độ xem nội dung của
app và đó chỉ là tệp trống ở đây
và những gì chúng ta sẽ làm trong này
chế độ xem nội dung thực sự là tạo một
ngăn xếp z

nơi chúng tôi có tab chính và phụ
sắp xếp menu xếp chồng lên nhau
khác bên trong một khung nhìn khác đó là
sẽ là chế độ xem nội dung này và sau đó
chúng tôi sẽ thêm một số bài thuyết trình
logic để giống như khi chúng ta nhấn cái
này

nút lên ở trên cùng trên điều hướng
quán ba

rằng nó sẽ đưa menu phụ đó vào
xem và chúng sẽ được sắp xếp trong một
z-stack nơi bạn có thể thấy rằng bên
menu được sắp xếp ở trên cùng của tab
chính

xem vì vậy đó là cách chúng ta sẽ làm
cái này

dù sao đi nữa

hãy tiếp tục và thực sự chỉ cần chạy
ứng dụng thực sự nhanh chóng

và nó sẽ giống như thế này
điều đó nói xin chào thế giới nhưng chúng
ta nên

lưu ý rằng chúng tôi đang ở trong một điều
hướng

Main subtitle

view and we can't actually see that

right now because we have to add uh some

components to our navigation bar

so really quickly before we do that i'm just going to say z stack and the alignment is going to be dot top leading and we'll see why this

comes into play in a little bit and then we're going to add the main tab view so let's actually run it again and see if our main tab view is showing up

and you guys notice that it is and now there's some space up at the top and that's actually the navigation bar so let's go ahead and configure our navigation bar

so that it has this uh

user profile

image view up at the top and then a navigation bar title

alright guys so that's pretty easy to do we just say navigation

title and we give it a text of home

and then we say that navigation bar

title display mode is dot inline so we

don't want that large title we want it

to be in line so that's looking good then we just need to create this button up at the top here so we're actually going to use this toolbar

Second subtitle

xem và chúng ta thực sự không thể thấy điều đó

ngay bây giờ bởi vì chúng ta phải thêm uh một số

các thành phần vào thanh điều hướng của chúng tôi

rất nhanh trước khi chúng tôi làm điều đó tôi chỉ cần nói z ngăn xếp

và sự liên kết sẽ được

dẫn đầu và chúng ta sẽ thấy tại sao điều này

đi vào chơi trong một chút

và sau đó chúng ta sẽ thêm tab chính

xem vì vậy hãy thực sự chạy lại và

xem chế độ xem tab chính của chúng tôi có hiển thị không

và các bạn lưu ý rằng nó là và bây giờ có một số không gian ở trên cùng và

đó thực sự là thanh điều hướng

vì vậy hãy tiếp tục và định cấu hình thanh điều hướng

để nó có cái này uh

Thông tin người dùng

xem hình ảnh ở trên cùng và sau đó là một tiêu đề thanh điều hướng

được rồi các bạn, điều đó khá dễ thực hiện chúng tôi chỉ nói điều hướng

tiêu đề và chúng tôi cung cấp cho nó một văn bản của nhà

và sau đó chúng tôi nói rằng thanh điều hướng

chế độ hiển thị tiêu đề là dấu chấm nội tuyến vì vậy chúng tôi

không muốn tiêu đề lớn mà chúng tôi muốn nó

để được xếp hàng vì vậy mà nhìn tốt

sau đó chúng ta chỉ cần tạo cái này

nút lén ở trên cùng ở đây vì vậy chúng tôi thực sự sẽ sử dụng thanh công cụ này

Main subtitle

option

apple said that the navigation bar items

is getting deprecated they want us to
use this
so here

you just place a toolbar item
and you're going to do this one that has
placement and then you can say dot
navigation bar leading and the content
is just going to be a circle
with a frame of 32 by 32 and that's
going to be our profile image eventually

but this is essentially just going to
act as our button to show our side menu

so let's actually go ahead and wrap this
in a button or create a button out of it
so you're going to say action and label
label is going to be that and then we're
going to toggle some property here
that's actually going to show and hide
our side menu so let's go ahead and

create it now we're going to say state

private var

show

showing menu or show menu

equals false and then here we're just
going to say show menu dot toggle
okay so that's looking pretty good so
far we're also going to use the show
menu

to help us with our presentation logic

for how to show and hide the side menu

Second subtitle

lựa chọn

apple nói rằng các mục trên thanh điều
hướng

đang bị phản đối, họ muốn chúng tôi
dùng cái này
nên ở đây

bạn chỉ cần đặt một mục trên thanh công cụ
và bạn sẽ làm cái này có
vị trí và sau đó bạn có thể nói dấu chấm
thanh điều hướng hàng đầu và nội dung
sẽ chỉ là một vòng tròn

với khung 32 x 32 và đó là
cuối cùng sẽ là hình ảnh hồ sơ của chúng
tôi

nhưng điều này về cơ bản chỉ là đi đến
hoạt động như nút của chúng tôi để hiển thị
menu bên của chúng tôi

Vì vậy, hãy thực sự tiếp tục và hoàn thành
phần này

trong một nút hoặc tạo một nút từ nó
vì vậy bạn sẽ nói hành động và nhãn
nhãn sẽ là cái đó và sau đó chúng ta
sẽ chuyển đổi một số tài sản ở đây
điều đó thực sự sẽ hiển thị và ẩn
menu phụ của chúng tôi vì vậy hãy tiếp tục
và

tạo nó ngay bây giờ chúng ta sẽ nói trạng
tháí

var riêng

trình diễn

hiển thị menu hoặc hiển thị menu

bằng sai và sau đó ở đây chúng tôi chỉ
sẽ nói hiển thị menu dấu chấm chuyển đổi
được rồi vì vậy nó trông khá tốt vì vậy
xa chúng ta cũng sẽ sử dụng chương trình
thực đơn

để giúp chúng tôi với logic trình bày của
chúng tôi

để biết cách hiển thị và ẩn menu bên

Main subtitle

so one more time guys it's actually a z stack
 so we're going to put the side menu on top of
 the main tab view
 and
 we're just going to give the side menu a default offset so that it starts off the screen and then when we want to show the menu we just bring it onto the screen or into view on the screen so right now the side menu exists it's just off the screen that we hit the button and it comes onto screen and gets placed over top of the main tab view
 so let's go ahead and figure out how to do that
 so
 in a z-stack
 if you place something first and then something after the thing you place after goes on top so we're going to put the side menu view here and this will mean that the side menu view is on top of the main tab view
 so just to see that let's go ahead and give it a frame of width 300 and you guys will notice that it's going

Second subtitle

vì vậy một lần nữa các bạn nó thực sự là cây rơm
 vì vậy chúng tôi sẽ đặt menu phụ trên trên cùng của chế độ xem tab chính
 Và
 chúng tôi sẽ chỉ cung cấp menu phụ một phần bù mặc định để nó bắt đầu tắt màn hình và sau đó khi chúng tôi muốn hiển thị menu chúng tôi chỉ mang nó lên màn hình hoặc vào chế độ xem trên màn hình để ngay bây giờ menu bên tồn tại nó chỉ là tắt màn hình mà chúng tôi nhấn nút và nó xuất hiện trên màn hình và được đặt phía trên cùng của chế độ xem tab chính vì vậy hãy tiếp tục và tìm hiểu làm thế nào để làm việc đó đi
 Vì thế
 trong ngăn xếp z nếu bạn đặt một cái gì đó đầu tiên và sau đó một cái gì đó sau những điều bạn đặt sau khi đi lên trên cùng, vì vậy chúng tôi sẽ đặt chế độ xem menu bên ở đây và điều này sẽ có nghĩa là chế độ xem menu bên ở trên cùng của chế độ xem tab chính vì vậy chỉ để thấy rằng chúng ta hãy tiếp tục và cung cấp cho nó một khung của chiều rộng 300 và các bạn sẽ nhận thấy rằng nó đang diễn ra

Main subtitle

to be

on the top leading edge or that's where it's going to start because that's how we have our alignment configured for our z-stack

so let's actually just go ahead and run this and see if the side menu view is showing up

so

it's really messy but it is showing up right

so we actually want to give it a

background

of color.white

and let's go ahead and see if we can see

it a little bit better or i should probably made it like red or something

but it's okay

so we can see it now um

and we can see that this is like the

tweet or the feed view right we can

still see like the bookmark button on the cells and we can still see the tab bar down here

so this is um

you know

the side menu view showing up so that's good we just have to add a couple more things to get this all working properly

so

first off guys let's give it um an offset

Second subtitle

được

trên cạnh hàng đầu hoặc đó là nơi nó sẽ bắt đầu bởi vì đó là cách chúng tôi đã định cấu hình căn chỉnh cho ngăn xếp z

vì vậy hãy thực sự tiếp tục và chạy cái này và xem chế độ xem menu bên có hiển thị

Vì thế

nó thực sự lộn xộn nhưng nó đang hiển thị
Phải

vì vậy chúng tôi thực sự muốn cung cấp cho nó một

lý lịch

màu sắc. trắng

và hãy tiếp tục và xem liệu chúng ta có thể thấy

nó tốt hơn một chút hoặc tôi nên

có lẽ làm cho nó giống như màu đỏ hoặc một cái gì đó

Nhưng nó ổn mà

vì vậy chúng ta có thể thấy nó bây giờ um và chúng ta có thể thấy rằng điều này giống như

tweet hoặc chế độ xem nguồn cấp dữ liệu bên phải, chúng tôi có thể

vẫn thấy như nút đánh dấu trên

các ô và chúng ta vẫn có thể thấy tab thanh xuống đây

vậy đây là ừm

Bạn biết

chế độ xem menu bên hiển thị vì vậy đó là tốt, chúng ta chỉ cần thêm một vài chi tiết mọi thứ để làm cho tất cả điều này hoạt động bình thường

Vì thế

trước hết các bạn hãy cho nó um một bù lại

Main subtitle

and the x-axis is going to be if we're showing the menu yes it's 0 no it's minus 300. so this has a width of 300 and if we're not showing it we're going to just move it over to the left by 300 pixels so let's run this again and we should notice that oops y is just going to be zero that we can no longer see the side menu

oh um hold on that's weird

we can

let's just give it a color a background color of clear

so show menu yes it's zero

[Music]

no it's 300

right okay so yeah let's go here guys

we're going to say show menu yes

color:white no color.clear

okay so now we actually can't see it and

if i were to press this button

you guys notice that it comes into view

right it's just not animating so we're

gonna fix all that stuff we're gonna add

like a shadow view here so let's just

keep going with this

um

on the main tab you were in between

these guys we're gonna say if show menu

and we're gonna add like a black view

Second subtitle

và trục x sẽ là nếu chúng ta hiển thị menu vắng, đó là 0 không, nó là âm 300. vậy cái này có chiều rộng

của 300 và nếu chúng tôi không hiển thị nó, chúng tôi sẽ chỉ cần di chuyển nó qua bên trái bằng 300 pixel

vì vậy hãy chạy lại cái này và chúng ta nên lưu ý rằng rất tiếc y sẽ chỉ là số không

rằng chúng ta không còn có thể nhìn thấy menu bên

oh um giữ nó thật kỳ lạ

chúng ta có thể

chúng ta hãy tạo cho nó một màu nền màu rõ ràng

vì vậy hãy hiển thị menu vắng, không [Âm nhạc]

không, nó là 300

đúng rồi, vắng, chúng ta hãy đi đây các bạn chúng tôi sẽ nói hiển thị menu có color:white không color.clear

được rồi, bây giờ chúng tôi thực sự không thể nhìn thấy nó và

nếu tôi nhấn nút này

các bạn chú ý là nó vào xem

đúng là nó không hoạt hình nên chúng tôi sẽ sửa chữa tất cả những thứ mà chúng tôi sẽ thêm vào

giống như một cái nhìn bóng tối ở đây vì vậy hãy chỉ

tiếp tục với điều này

ô

trên tab chính bạn đang ở giữa

những người này chúng ta sẽ nói nếu hiển thị menu

và chúng ta sẽ thêm như một chế độ xem màu đen

Main subtitle

here to cover up this uh portion of the view
 um so that it looks like a shadow is being placed on top of it so we're gonna say z stack
 we're gonna say color dot black
 dot opacity
 0.25
 and we're actually going to now give this an on tap gesture
 and say with animation dot ease in out showing men or show menu equals false so what this is going to do guys
 is when we tap on this like shadow view it's going to dismiss the side menu and then on the main tab view um we notice that we're going to want to hide the navigation bar here so we're actually going to go on the main tab view and say dot navigation are hidden
 and that is going to be showing menu or show menu
 so let's see what we have now with this
 okay so that's looking pretty good and then we hit this okay
 so our side menu is looking

Second subtitle

ở đây để che đậy phần uh này của xem
 ừm để nó trông giống như một cái bóng được đặt trên đầu trang của nó vì vậy chúng tôi sẽ nói z ngăn xếp
 chúng ta sẽ nói màu sắc chấm đen độ mờ chấm 0,25 và chúng tôi thực sự sẽ bây giờ đưa cái này một cử chỉ trên máy và nói với hoạt hình chấm dễ dàng trong ra ngoài hiển thị đàn ông hoặc hiển thị menu bằng sai vậy điều này sẽ làm gì các bạn là khi chúng ta nhấn vào cái này giống như chế độ xem bóng tối nó sẽ loại bỏ menu bên và sau đó trên chế độ xem tab chính ừm, chúng tôi nhận thấy rằng chúng tôi sẽ muốn ẩn thanh điều hướng ở đây để chúng tôi thực sự sẽ đi trên tab chính xem và nói điều hướng dấu chấm Đang ẩn và điều đó sẽ được hiển thị menu hoặc hiển thị menu vì vậy hãy xem những gì chúng ta có bây giờ Với cái này được rồi, điều đó trông khá tốt và sau đó chúng tôi đánh cái này được rồi vì vậy menu phụ của chúng tôi đang tìm kiếm

Main subtitle

uh better now right and then we see that
 we have this like sort of shadow view on
 top of our main tab view and if i were
 to hit that
 it sort of slides everything back over
 so
 all right guys now we just want to clean
 up the animation a little bit as well so
 when we actually toggle the menu let's
 go down uh to this button action handler
 cut that with command x and say with
 animation dot ease in out
 and then say show menu.toggle the same
 way we toggle it in we need to toggle it
 back out
 and then we're also
 going to add um
 something called ignore safe area on
 this z-stack so
 go here guys and say just say ignore
 safe area
 um
 and that's just going to make sure that
 the shadow view stretches all the way to
 the top and bottom of the screen not
 respecting those safe areas and
 for the opacity
 or we're going to say
 show menu actually let's let me show you
 guys what that's going to do before i do
 it so you can understand it better

Second subtitle

uh tốt hơn ngay bây giờ và sau đó chúng ta
 thấy điều đó
 chúng tôi có cái này giống như loại chế độ
 xem bóng tối trên
 đầu chế độ xem tab chính của chúng tôi và
 nếu tôi là
 đánh cái đó
 nó sắp trượt mọi thứ trở lại
 Vì thế
 được rồi các bạn bây giờ chúng tôi chỉ
 muốn làm sạch
 lên hoạt hình một chút cũng vậy
 khi chúng ta thực sự chuyển menu, hãy
 đi xuống uh để xử lý hành động nút này
 cắt nó bằng lệnh x và nói với
 hoạt hình chấm dễ dàng trong ra ngoài
 và sau đó nói hiển thị menu. chuyển đổi
 tương tự
 cách chúng ta chuyển đổi nó trong chúng ta
 cần chuyển đổi nó
 rút lui
 và sau đó chúng tôi cũng
 sẽ thêm um
 một cái gì đó gọi là bỏ qua khu vực an toàn
 trên
 ngăn xếp z này vì vậy
 đi đây các bạn và nói chỉ cần nói bỏ qua
 khu vực an toàn
 ô
 và điều đó sẽ đảm bảo rằng
 chế độ xem bóng trải dài đến tận
 trên cùng và dưới cùng của màn hình không
 tôn trọng những khu vực an toàn đó và
 cho độ mờ
 hoặc chúng ta sẽ nói
 thực sự hiển thị menu, hãy để tôi chỉ cho
 bạn
 các bạn sẽ làm gì trước khi tôi làm
 nó để bạn có thể hiểu nó tốt hơn

Main subtitle

but anyway

let's see how this is looking now it
should look pretty smooth
awesome right that's pretty smooth
we click the black view or the shadow

view and it just goes right back over
and we notice that we can't do anything

on the feed this acts as sort of like a

like um

like an overlay view and we just hit it

and it dismisses our side menu and we

notice that it stretches all the way to
the top and bottom of the view as well
so really quickly guys

um i'm going to add

some logic to this opacity as well i'm
going to say show menu yes

0.25

no 0.0

and what this is going to do is just
make it so that the opacity is based off
of this

show menu guy as well and it's going to
animate in and out for us

a little smoother um than if we don't
have that there so that's looking pretty
good

um

and that's pretty much gonna wrap it up
for that side menu presentation logic

guys

um

the next thing i want us to do is

Second subtitle

nhưng dù sao

hãy xem nó trông như thế nào bây giờ
nên trông khá mượt mà
tuyệt vời đúng là khá trơn tru
chúng tôi nhấp vào chế độ xem màu đen
hoặc bóng tối

xem và nó chỉ quay lại ngay
và chúng tôi nhận thấy rằng chúng tôi không
thể làm bất cứ điều gì

trên nguồn cấp dữ liệu này hoạt động giống
như một

như ừm

giống như một chế độ xem lớp phủ và
chúng tôi chỉ nhấn nó

và nó loại bỏ menu phụ của chúng tôi và
chúng tôi

lưu ý rằng nó kéo dài tất cả các cách để
trên cùng và dưới cùng của chế độ xem
nhanh thật đấy các bác

ừm tôi sẽ thêm

một số logic cho độ mờ này cũng như tôi
sẽ nói hiển thị menu có

0,25

không 0,0

và những gì nó sẽ làm chỉ là
làm cho nó dễ độ mờ được dựa trên
điều này

hiển thị anh chàng menu và nó sẽ
sinh động trong và ngoài cho chúng tôi
một chút mượt mà hơn nếu chúng ta không
có cái đó ở đó nên nó trông rất đẹp

Tốt

ô

và đó là khá nhiều sẽ kết thúc nó
cho logic trình bày menu bên đó
các bạn

ô

điều tiếp theo tôi muốn chúng ta làm là

Main subtitle

actually implement a action to this profile button so that it actually takes us over to the user profile so let's do that and that is going to pretty much wrap it up for this chapter so what we're going to do is handle that here so we're going to go inside this guy and we're going to say if option equals dot profile so that's another awesome part about the enum we can check and see if we're looking at a certain option and we get this awesome like dot notation for it um then i'm going to create a navigation link destination label cool so the destination is going to be the profile view and then the label is going to be this guy we see here so we're going to notice that we do actually want to extract this as a like as a reusable view so let's go down at the bottom and create an extension here on side menu view we're going to say var side menu option row

Second subtitle

thực sự thực hiện một hành động với nút hồ sơ này để nó thực sự đưa chúng tôi đến với người dùng hồ sơ vì vậy hãy làm điều đó và điều đó sẽ khá nhiều kết thúc cho chương này vì vậy những gì chúng ta sẽ làm là xử lý nó đây Vì thế chúng ta sẽ đi vào bên trong anh chàng này và chúng ta sẽ nói nếu tùy chọn bằng hồ sơ dấu chấm thì đó là một phần tuyệt vời khác về enum chúng tôi có thể kiểm tra và xem nếu chúng ta đang xem xét một lựa chọn nhất định và chúng tôi nhận được ký hiệu giống như dấu chấm tuyệt vời này cho nó ừm thì tôi sẽ tạo một điều hướng liên kết nhãn đích mát mẻ vì vậy điểm đến sẽ là xem hồ sơ và sau đó nhãn sẽ đi được anh chàng này chúng ta thấy ở đây vì vậy chúng tôi sẽ nhận thấy rằng chúng tôi làm thực sự muốn trích xuất cái này như một lượt thích dưới dạng chế độ xem có thể tái sử dụng, vì vậy hãy đi xuống tại phía dưới và tạo tiện ích mở rộng tại đây ở chế độ xem menu bên chúng ta sẽ nói var tùy chọn menu bên hàng ngang

Main subtitle

or option row view
 some view
 and we're just going to
 take this guy and put it in here and i
 actually think we need to make this its
 own view guys that's probably why i had
 it like that before because we need
 access to this option right
 so let's go ahead and actually just
 create that view
 swift ui view we're going to say psi or
 the outside menu
 option
 row view
 it's a little bit long but that's okay
 and we're actually going to initialize
 this with a option or we could call it a
 view model
 side menu view model
 and then go back to this guy
 and just cut this h stack out we can
 remove this extension
 and then just go ahead and paste that in
 there and everything should act exactly
 the same
 um we just need to call this view model
 and we need to do that in the preview as
 well just pass in like dot profile
 so this is what your option row view
 should look like side menu view now we
 can go back here and in the label

Second subtitle

hoặc chế độ xem hàng tùy chọn
 một số xem
 và chúng tôi sẽ chỉ
 đưa anh chàng này và đặt nó vào đây và tôi
 thực sự nghĩ rằng chúng ta cần phải làm
 điều này
 những người có quan điểm riêng, đó có lẽ là
 lý do tại sao tôi có
 nó như thế trước bởi vì chúng ta cần
 truy cập vào tùy chọn này ngay
 vì vậy hãy tiếp tục và thực sự chỉ
 tạo chế độ xem đó
 Chế độ xem ui nhanh chóng, chúng tôi sẽ
 nói psi hoặc
 thực đơn bên ngoài
 lựa chọn
 xem hàng
 nó hơi dài nhưng không sao
 và chúng tôi thực sự sẽ khởi tạo
 cái này với một tùy chọn hoặc chúng ta có
 thể gọi nó là một
 xem mô hình
 mô hình xem menu bên
 và sau đó quay lại với anh chàng này
 và chỉ cần cắt ngăn xếp h này ra là chúng ta
 có thể
 xóa phần mở rộng này
 và sau đó chỉ cần tiếp tục và dán nó vào
 ở đó và mọi thứ nên hành động chính xác
 giống nhau
 ừm chúng ta chỉ cần gọi mô hình xem này
 và chúng ta cần làm điều đó trong bản xem
 trước như
 cũng chỉ cần vượt qua như chấm hồ sơ
 vì vậy đây là chế độ xem hàng tùy chọn của
 bạn
 bây giờ trông giống như chế độ xem menu
 bên, chúng tôi
 có thể quay lại đây và trong nhãn

Main subtitle

we can say side menu

option row view and just pass in that

guy and we can actually rename this to

view model

just so it makes a little bit more sense
and then here guys

um we're actually going to add an else
if option equals dot log out

we're going to say print

uh or you know we're actually

going to add like um

an action handler for logging out there

and then we can just go here and say
else and then pass in the side menu
option

row view

view model just like that

oh and that needs to be view model
and that needs to be modeled cool

okay so now our builds should succeed

sorry that was a little bit messy

um

so now guys let's go and run this code

and see if this actually serves as a

navigation link or works as a navigation

link so i'm going to go here

and we notice that the text changed to

blue and that's an indicator that it
turned into a navigation link we just

need to go back in there and change that

Second subtitle

chúng ta có thể nói menu phụ

tùy chọn chế độ xem hàng và chỉ cần
chuyển vào đó

anh chàng và chúng ta thực sự có thể đổi
tên cái này thành

xem mô hình

chỉ để nó có ý nghĩa hơn một chút
và sau đó ở đây các bạn

ừm, chúng tôi thực sự sẽ thêm một cái khác
nếu tùy chọn bằng dấu chấm đăng xuất
chúng ta sẽ nói in

uh hoặc bạn biết chúng tôi thực sự
sẽ thêm như um

một trình xử lý hành động để đăng xuất ở
đó

và sau đó chúng ta có thể đến đây và nói
khác và sau đó chuyển vào menu bên
lựa chọn

xem hàng

xem mô hình như thế

oh và đó cần phải được xem mô hình
và điều đó cần phải được mô hình hóa tuyệt
vời

được rồi, bây giờ các bản dựng của chúng
ta sẽ thành công

xin lỗi đó là một chút lộn xộn

ô

Vì vậy, bây giờ các bạn hãy đi và chạy mã
này

và xem nếu điều này thực sự phục vụ như
là một

liên kết điều hướng hoặc hoạt động như
một điều hướng

liên kết vì vậy tôi sẽ đi đây

và chúng tôi nhận thấy rằng văn bản đã thay
đổi thành

màu xanh và đó là một dấu hiệu cho thấy nó
biến thành một liên kết điều hướng chúng
tôi chỉ

cần quay lại đó và thay đổi điều đó

Main subtitle

so that it doesn't change the blue but if i hit profile you guys will notice it takes me right over to my profile which is pretty awesome right and then if i go back it takes me back here but the side menu view is still showing so we're going to want to make it so that if we go back it just takes us back to the feed view so let's go ahead and fix those two things and then wrap up this chapter so in the side menu row view for this guy we're going to say foreground color is dot black so that'll solve that problem where it turns the navigation link text blue and then back in our content view on this z stack guys we're just gonna say dot on appear show menu equals false so that way every time this view appears like so when we hit back on appear will get called and then it's going to hide the menu or it's going to set show menu to false and once that sets to false it goes back through our logic in the view and we'll hide the side menu so we go here we

Second subtitle

để nó không thay đổi màu xanh nhưng nếu tôi nhấn hò sơ các bạn sẽ nhận thấy nó đưa tôi đến ngay hò sơ của tôi là khá tuyệt vời phải không và sau đó nếu tôi quay trở lại nó đưa tôi trở lại đây nhưng menu bên chế độ xem vẫn đang hiển thị nên chúng tôi sẽ muốn làm cho nó để nếu chúng ta quay trở lại nó chỉ đưa chúng tôi trở lại chế độ xem nguồn cấp dữ liệu hãy tiếp tục và sửa chữa hai điều đó và sau đó kết thúc chương này để trong chế độ xem hàng menu bên cho anh chàng này, chúng tôi sẽ nói màu nền trước là dấu chấm đen vì vậy sẽ giải quyết vấn đề đó ở đâu biến văn bản liên kết điều hướng thành màu xanh lam và sau đó quay lại chế độ xem nội dung của chúng tôi trên ngăn xếp z này, chúng ta sẽ nói dấu chấm xuất hiện hiển thị menu bằng sai để theo cách đó mọi thời gian chế độ xem này xuất hiện như vậy khi chúng tôi quay lại sẽ xuất hiện được gọi và sau đó nó sẽ ẩn thực đơn hoặc nó sẽ đặt menu hiển thị thành false và một khi nó được đặt thành false, nó sẽ quay trở lại thông qua logic của chúng tôi trong chế độ xem và chúng tôi sẽ ẩn menu bên để chúng tôi đến đây chúng tôi

Main subtitle

notice that that part's fixed go here we hit the back button and we notice that like it just brings us back to our main view so that is looking super super smooth let me just hit profile one more time we go back and when we go back that like i said that on appear guy gets called it sets show menu to false and then um it just sort of animates everything back to its original state of not showing the menu and showing our main tab view here one thing that we notice however is that our logout button disappeared so really quickly let's fix that and then call this chapter or section a wrap so go back to side menu view guys we have no code with for this log out option and that's because we want to create a button here and our action we're going to just say like print handle log out here and our label is just going to be this guy so here this is a little bit messy we do have to go through some logic here um to

Second subtitle

lưu ý rằng phần đó đã được sửa ở đây chúng tôi nhấn nút quay lại và chúng tôi nhận thấy rằng nó chỉ mang lại chúng tôi trở lại chế độ xem chính của chúng tôi trông siêu đẹp siêu mượt để tôi chỉ cần nhấn hồ sơ một thêm thời gian chúng ta quay lại và khi chúng ta quay lại như tôi đã nói rằng khi xuất hiện, anh chàng được gọi là nó đặt menu hiển thị thành false và sau đó ừm, nó chỉ làm sinh động mọi thứ trở lại trạng thái ban đầu của nó không hiển thị menu và hiển thị chính của chúng tôi tab xem ở đây Tuy nhiên, một điều mà chúng tôi nhận thấy là nút đăng xuất của chúng tôi đã thực sự biến mất hãy nhanh chóng khắc phục điều đó và sau đó gọi chương hoặc phần này kết thúc vì vậy hãy quay lại chế độ xem menu bên, chúng tôi không có mã nào cho việc đăng xuất này tùy chọn và đó là bởi vì chúng tôi muốn tạo một nút ở đây và hành động của chúng tôi, chúng tôi sẽ chỉ nói như xử lý in đăng xuất tại đây và nhãn của chúng tôi sẽ là thế này chàng trai vì vậy đây là một chút lộn xộn chúng tôi làm phải trải qua một số logic ở đây um để

Main subtitle

handle what we want each one of the buttons to do
 but it's not that bad like the logic really isn't terrible um and now we can run this again and
 we can see that when we go back to our side menu the log out button is going to be there and we can click it and it's it works as a button now and we can see our print statement showing up in our console there so that's going to wrap it up for the side menu feature guys uh we can get to our profile from here which is awesome and eventually we'll be able to log out from there and show that login and sign up flow and we're going to be working on building that soon but before we do that the next ui component we're going to build the second to last one is going to be that upload a tweet view and that's going to be super quick we just have to build this floating button present a sheet and then show our upload a tweet view and that's really going to be super

Second subtitle

xử lý những gì chúng tôi muốn mỗi một trong những nút để làm nhưng nó không tệ như logic thực sự không tệ lắm ừm và bây giờ chúng ta có thể chạy cái này một lần nữa Và chúng ta có thể thấy điều đó khi chúng ta quay trở lại menu bên, nút đăng xuất sẽ xuất hiện ở đó và chúng ta có thể nhấp vào nó và nó hoạt động như một nút bây giờ và chúng ta có thể thấy bản in của mình tuyên bố hiển thị trong bảng điều khiển của chúng tôi ở đó vì vậy sẽ kết thúc nó cho tính năng menu bên các bạn uh chúng ta có thể nhận được đến hồ sơ của chúng tôi từ đây tuyệt vời và cuối cùng chúng ta sẽ có thể đăng xuất khỏi đó và hiển thị quy trình đăng nhập và đăng ký đó và chúng tôi sẽ làm việc trên xây dựng nó sớm nhưng trước khi chúng tôi làm điều đó thành phần giao diện người dùng tiếp theo chúng ta sẽ làm xây dựng cái thứ hai đến cái cuối cùng sẽ thành là tải lên một lượt xem tweet và đó là sẽ rất nhanh, chúng ta chỉ cần xây dựng nút nỗi này trình bày một trang tính và sau đó hiển thị một tweet tải lên của chúng tôi xem và đó thực sự sẽ là siêu

Main subtitle

quick and easy then we build the login and sign up page and then we are ready to start hooking up our api which is going to be super exciting the app is looking super slick i'm hoping you guys have this working right now and it's looking good for you as well let's go ahead and power through finishing up this ui so we can get to the database stuff which is going to be really exciting all right guys so we're going to start by creating this floating button here so what we're going to do is hop into the feed view and we're going to command click onto the scroll view and click embed in z stack so this is just going to uh sort of float on top of the scroll view you notice we'll be able to scroll but the the button stays stationary right which is what we want and we're going to set an alignment on this of dot bottom trailing and that's how we're going to get it to show up in the bottom right corner of the screen so just go ahead here and create a button we can say print

Second subtitle

nhanh chóng và dễ dàng sau đó chúng tôi xây dựng thông tin đăng nhập và đăng ký trang và sau đó chúng tôi đã sẵn sàng để bắt đầu kết nối api của chúng tôi, đó là ứng dụng sẽ cực kỳ thú vị trông siêu bóng bẩy, tôi hy vọng các bạn có cái này hoạt động ngay bây giờ và nó tìm kiếm tốt cho bạn cũng như chúng ta hãy đi phía trước và sức mạnh thông qua kết thúc lên giao diện người dùng này để chúng tôi có thể truy cập cơ sở dữ liệu những thứ sẽ thực sự thú vị được rồi mọi người, vậy chúng ta sẽ bắt đầu bằng cách tạo nút nổi này tại đây vì vậy những gì chúng ta sẽ làm là nhảy vào chế độ xem nguồn cấp dữ liệu và chúng tôi sẽ ra lệnh nhấp vào chế độ xem cuộn và nhấp nhúng vào ngăn xếp z vì vậy điều này sẽ diễn ra

ĐÉN

uh loại nổi trên đầu cuộn xem bạn nhận thấy chúng tôi sẽ có thể cuộn nhưng nút vẫn đứng yên bên phải là những gì chúng tôi muốn và chúng ta sẽ thiết lập một sự liên kết trên đây là dấu chấm phía dưới và đó là làm thế nào chúng ta sẽ làm cho nó xuất hiện trong góc dưới cùng bên phải của màn hình vì vậy chỉ cần tiếp tục ở đây và tạo một cái nút chúng ta có thể nói in

Main subtitle

show new tweet view

and then the label is just going to be
this button here

this is an asset that you guys are going
to need to download so the link to that
is in the description

the source code will be available on my
personal website guys and the link to
that is also in the description so check

that out if you want the completed
source code

but anyway we're gonna go here and we're

just gonna say image

tweet then we're gonna say dot resizable

make the rendering mode dot template

give it a frame of 28 by 28

and then you're gonna give it some
padding

and then uh so that looks pretty good we
just need to give it that clip shape
so we're going to say background is
color

oops parenthesis dot system blue

so let's see what that does so that
looks okay we're going to say foreground
colors.white dot clip shape is
a circle

and then we're going to give it some

padding

so that looks really good and

now what we need to do guys is make it

so that when we click this it presents

Second subtitle

hiển thị chế độ xem tweet mới

và sau đó nhãn sẽ chỉ là
nút này ở đây

đây là một tài sản mà các bạn đang đi
cần tải xuống nên liên kết đến đó
là trong mô tả

mã nguồn sẽ có sẵn trên của tôi
trang web cá nhân và liên kết đến
cái đó cũng có trong phần mô tả nên hãy
kiểm tra

đó ra nếu bạn muốn hoàn thành
mã nguồn

nhưng dù sao thì chúng ta sẽ đến đây và
chúng ta

chỉ cần nói hình ảnh

tweet thì chúng ta sẽ nói dấu chấm có thể
thay đổi kích thước

tạo mẫu dấu chấm chế độ kết xuất
cung cấp cho nó một khung hình 28 x 28
và sau đó bạn sẽ cho nó một số
đệm

và sau đó uh để trông khá tốt, chúng tôi
chỉ cần cung cấp cho nó hình dạng clip đó
vì vậy chúng ta sẽ nói nền là
màu sắc

rất tiếc dấu ngoặc đơn hệ thống dấu chấm
màu xanh lam

vì vậy hãy xem những gì nó làm để
có vẻ ổn, chúng ta sẽ nói tiền cảnh
colors.white dot clip hình dạng là
một vòng tròn

và sau đó chúng tôi sẽ cung cấp cho nó một
số

đệm

vì vậy trông thực sự tốt và
bây giờ những gì chúng ta cần làm là làm
cho nó

để khi chúng tôi nhấp vào đây, nó sẽ hiển thị

Main subtitle

this full screen

and then we're going to work on building
this view shortly after

so

uh let's go ahead and figure out how

we're going to do that so we're going to
create a state property up at the top
called state private bar show new tweet

view

and we're going to set it equal to false
and in the action for this guy we're

just going to say show new
tweetview.toggle

and then down at the bottom here right
under padding you're going to say dot
full screen cover and you're going to
select the option that says is presented
on dismiss and content

so is presented you're going to say
dollar sign shown you tweet view toggle

oops

not that toggle you just say show new

tweet view so now this is bound to the

state property so when this changes to

true it's going to present our full

screen cover

um on dismiss

we don't need

and for the content let's just go ahead

and say text

new tweet

view

Second subtitle

tòan màn hình này

và sau đó chúng ta sẽ làm việc để xây dựng
quan điểm này ngay sau đó

Vì thế

uh chúng ta hãy tiếp tục và tìm hiểu làm thế
nào

chúng tôi sẽ làm điều đó vì vậy chúng tôi sẽ
tạo một tài sản nhà nước ở trên cùng
được gọi là thanh trạng thái riêng hiển thị
tweet mới

xem

và chúng ta sẽ đặt nó bằng false

và trong hành động cho anh chàng này,
chúng tôi

chỉ cần nói hiển thị mới

tweetview.toggle

và sau đó xuống dưới cùng ở đây bên phải
dưới phần đệm bạn sẽ nói dấu chấm

che toàn màn hình và bạn sẽ

chọn tùy chọn cho biết được trình bày
về việc loại bỏ và nội dung

vì vậy được trình bày bạn sẽ nói

ký hiệu đô la hiển thị cho bạn chế độ xem
tweet chuyển đổi

Ói

không phải chuyển đổi mà bạn chỉ cần nói
hiển thị mới

chế độ xem tweet vì vậy bây giờ điều này bị
ràng buộc với

tài sản nhà nước vì vậy khi điều này thay
đổi thành

đúng là nó sẽ trình bày đầy đủ của chúng tôi
bìa màn hình

ừm nghỉ việc

chúng tôi không cần

và đối với nội dung, hãy tiếp tục

và nói văn bản

tweet mới

xem

Main subtitle

so now we can go ahead and run the

application

and we should see our button showing up there

and we should be able to press it okay so it's showing up it's acting as a floating button which is nice we press it and it shows this full

screen cover which says new tweet view so now all we got to do is build out

that new tweet view and we can move on

so there's a couple things we have to do to get this done

uh

it looks pretty straightforward but we're gonna have to build like a custom

text view here

and that's because you know you want the text to wrap like this

and if you have a text field it would just keep going off screen so we have to

create like a text view and give it a

placeholder which is kind of annoying but uh it is what it is it was the same thing that we have to do in ui kit so anyway let's go up to the core folder create two new groups or one new group sorry we're gonna let's call it upload tweet

and then we're going to create two subfolders

one for view models

Second subtitle

vì vậy bây giờ chúng ta có thể tiếp tục và chạy

ứng dụng

và chúng ta sẽ thấy nút của mình hiển thị ở đó

và chúng ta sẽ có thể nhấn nó được rồi vì vậy nó đang hiển thị hoạt động như một nút nổi tốt đẹp, chúng tôi nhấn nó và nó hiển thị đầy đủ

bà màn hình cho biết chế độ xem tweet mới vì vậy bây giờ tất cả những gì chúng ta phải làm là xây dựng

chế độ xem tweet mới đó và chúng ta có thể tiếp tục

vì vậy có một vài điều chúng ta phải làm để hoàn thành việc này uh

nó trông khá đơn giản nhưng

chúng ta sẽ phải xây dựng như một phong tục

văn bản xem tại đây

và đó là bởi vì bạn biết bạn muốn

văn bản để bọc như thế này

và nếu bạn có một trường văn bản thì nó sẽ chỉ cần tiếp tục tắt màn hình vì vậy chúng tôi phải

tạo giống như một chế độ xem văn bản và cung cấp cho nó một

trình giữ chỗ gây phiền nhiễu

nhưng uh nó là như vậy nó là như nhau

điều mà chúng ta phải làm trong ui kit vì vậy

dù sao đi nữa, hãy đi đến thư mục cốt lõi

tạo hai nhóm mới hoặc một nhóm mới

xin lỗi, chúng tôi sẽ gọi nó là tải lên

tiếng riu ríu

và sau đó chúng ta sẽ tạo ra hai thư mục con

một cho các mô hình xem

Main subtitle

and one for views

so

in the views folder go ahead and create
a swift ui view we're going to call it

new tweet

view

create

okay

so let's see um what this is going to be
we're going to this is going to be
it's not a navigation bar so we're just

going to create our own custom like

action buttons up here

and then we need to create another age

stack down here

with our picture and this custom text

area

so

our parent view is going to be a v stack

because these things are vertically
stacked and we're going to have two h
stacks

so this first one is going to be a

button

action label

print dismiss this is going to be our

cancel button right here

the label is going to be super simple

it's just going to say text cancel

dot foreground color

color parenthesis

dot system blue

cool

Second subtitle

và một cho lượt xem

Vì thế

trong thư mục lượt xem, hãy tiếp tục và tạo
một chế độ xem ui nhanh chóng, chúng tôi
sẽ gọi nó

tweet mới

xem

tạo nên

được rồi

Vì vậy, hãy xem um những gì sẽ được
chúng ta sẽ đến đây sẽ là

nó không phải là một thanh điều hướng vì
vậy chúng tôi chỉ

sẽ tạo tùy chỉnh của riêng chúng tôi như
các nút hành động ở đây

và sau đó chúng ta cần tạo ra một thời đại
khác

xếp chồng xuống đây

với hình ảnh của chúng tôi và văn bản tùy
chỉnh này

khu vực

Vì thế

chế độ xem gốc của chúng tôi sẽ là một
ngăn xếp v

bởi vì những thứ này là theo chiều dọc
xếp chồng lên nhau và chúng ta sẽ có hai h
ngăn xếp

vì vậy cái đầu tiên này sẽ là một
cái nút

nhãn hành động

in bỏ qua đây sẽ là của chúng tôi
nút hủy ngay tại đây

nhãn sẽ trở nên siêu đơn giản

nó sẽ nói hủy văn bản

chấm màu nền trước

dấu ngoặc đơn màu

hệ thống chấm màu xanh

mát mẻ

Main subtitle

uh let's resume our preview
 awesome so that looks pretty good let's
 go ahead and copy this button code guys
 after that add a spacer because we're
 just gonna add a spacer in between these
 two things
 go ahead and paste it
 so we have two cancel buttons now this
 one is now oops gonna our our action we
 could change to say tweet
 also change your text to say tweet
 and then we just need to give this like
 this sort of background here
 so let's go ahead and just delete this
 we're going to give it
 some padding
 of dot horizontal
 and also
 some padding of dot vertical
 prep comma eight
 so this sort of gives it that like wider
 or like rectangular frame with the
 padding parameters here or modifiers
 then we're gonna say dot background
 color
 parenthesis dot system blue
 so alex okay and then give it a
 foreground color of white
 so now tweet shows back up right that
 like just like that
 and then we're going to give it a clip

Second subtitle

uh hãy tiếp tục xem trước của chúng tôi
 tuyệt vời để có vẻ khá tốt chúng ta hãy
 hãy tiếp tục và sao chép mã nút này các
 bạn
 sau đó thêm một miếng đệm bởi vì chúng
 tôi
 chỉ cần thêm một miếng đệm ở giữa những
 thứ này
 hai điều
 tiếp tục và dán nó
 vì vậy bây giờ chúng tôi có hai nút hủy
 một là bây giờ rất tiếc sẽ hành động của
 chúng tôi chúng tôi
 có thể thay đổi để nói tweet
 cũng thay đổi văn bản của bạn để nói tweet
 và sau đó chúng ta chỉ cần đưa ra như thế
 này
 loại nền này ở đây
 vì vậy hãy tiếp tục và chỉ cần xóa cái này
 chúng tôi sẽ cung cấp cho nó
 một số phần đệm
 chấm ngang
 và cũng
 một số phần đệm của dấu chấm dọc
 chuẩn bị dấu phẩy tám
 vì vậy loại này mang lại cho nó rộng hơn
 hoặc như khung hình chữ nhật với
 tham số đệm ở đây hoặc công cụ sửa đổi
 sau đó chúng ta sẽ nói dấu chấm nền
 màu sắc
 dấu ngoặc đơn hệ thống dấu chấm màu
 xanh
 vì vậy alex được rồi hãy cho nó một
 màu nền trước màu trắng
 vì vậy bây giờ tweet hiển thị sao lưu đúng
 như vậy
 như thế thôi
 và sau đó chúng tôi sẽ cung cấp cho nó một
 clip

Main subtitle

shape of a capsule
so it's going to like
make it look like a capsule there and

we're also going to bold the tweet text
just like that so that looks pretty
smooth and then on this h stack we're

going to go ahead here
and just say dot padding
so that's looking pretty good
then

let's uh work on this other h stack so

we're just going to have a circle
for now
and give it a frame
of 64 by 64.
and that looks good
and then we are going to say
we have now we have sorry we have to

create this custom uh text area
so

let's just uh go ahead and do that now

so this is going to go
in our components folder
which i actually want to get up to the
top here
and let's just go ahead and create a new
file swift ui view
and we're going to call it text area
okay guys so this is gonna have uh two

properties
we're gonna say binding var text which
is a string because we have to pass like

Second subtitle

hình dạng của một viên nang
vì vậy nó sẽ thích
làm cho nó trông giống như một viên nang ở
đó và
chúng tôi cũng sẽ in đậm văn bản tweet
chỉ cần như vậy để trông đẹp
trơn tru và sau đó trên ngăn xếp h này,
chúng tôi
sẽ tiếp tục ở đây
và chỉ cần nói phần đậm dấu chấm
vì vậy điều đó trông khá tốt
sau đó
chúng ta hãy làm việc trên ngăn xếp h khác
này
chúng ta sẽ có một vòng tròn
bây giờ
và cung cấp cho nó một khung
của 64 nhân 64.
và điều đó có vẻ tốt
và sau đó chúng ta sẽ nói
chúng tôi có bây giờ chúng tôi xin lỗi chúng
tôi phải
tạo vùng văn bản uh tùy chỉnh này
Vì thế
chúng ta hãy tiếp tục và làm điều đó ngay
bây giờ
vì vậy điều này sẽ đi
trong thư mục thành phần của chúng tôi
mà tôi thực sự muốn đạt được
đầu ở đây
và chúng ta hãy tiếp tục và tạo một cái mới
tập tin nhanh chóng ui xem
và chúng ta sẽ gọi nó là vùng văn bản
được rồi các bạn vì vậy điều này sẽ có uh
hai
của cải
chúng ta sẽ nói văn bản var ràng buộc mà
là một chuỗi bởi vì chúng ta phải vượt qua
như

Main subtitle

a text in that will represent the tweet caption and we also need a placeholder so we're gonna say let placeholder which is also a string then we're going to say init underscore placeholder string text which is a binding with the string like that and we're going to say self.placeholder equals placeholder and self dot underscore text equals text so that's how you initialize a binding property just like that and also here we're going to say ui text view dot appearance oops well why is that not working appearance dot background color equals dot clear so that's going to give our text view a nice clear background color okay so now the rest of this is pretty easy we actually don't need the preview here guys we're going to go here and create a z stack and our alignment is going to be top leading and oops open up your brackets we're going to say text and we're going to put the placeholder there

Second subtitle

một văn bản trong đó sẽ đại diện cho tweet chú thích và chúng tôi cũng cần một trình giữ chỗ vì vậy chúng ta sẽ nói let placeholder which cũng là một chuỗi sau đó chúng ta sẽ nói init chuỗi giữ chỗ gạch dưới văn bản là ràng buộc với chuỗi như thế và chúng ta sẽ nói self.placeholder bằng giữ chỗ và tự chấm văn bản gạch dưới bằng văn bản vì vậy đó là cách bạn khởi tạo thuộc tính ràng buộc giống như điều đó và cũng ở đây chúng ta sẽ nói ui xem văn bản xuất hiện dấu chấm rất tiếc tại sao nó không hoạt động vẻ bề ngoài chấm màu nền bằng chấm rõ ràng vì vậy điều đó sẽ cung cấp cho chế độ xem văn bản của chúng ta màu nền rõ đẹp được rồi vì vậy bây giờ phần còn lại của điều này là khá dễ dàng chúng tôi thực sự không cần bản xem trước ở đây các bạn chúng ta sẽ đến đây và tạo một z ngăn xếp và sự liên kết của chúng tôi sẽ dẫn đầu và rất tiếc, hãy mở ngoặc của bạn chúng ta sẽ nói văn bản và chúng tôi sẽ đặt trình giữ chỗ ở đó

Main subtitle

dot foreground color and we're going to say color open parenthesis dot uh placeholder text just like that dot padding dot horizontal which is going to be 8 dot padding dot vertical which is going to be 12 okay and we're actually going to wrap this in an if statement and we're going to say if text dot is empty and then we're going to put that there and then else we're going to say text editor which is a actual swift ui component and this is where we have to pass in that binding property of our text i'm going to give it a padding of 4 pixels and the whole thing is going to have a font of that body okay so we're getting an error here let's see oh we can't put these parentheses there so now go ahead and build that and our custom component should be done let's go ahead and see now if we can drop it into our new tweet view

Second subtitle

chấm màu nền trước và chúng ta sẽ nói màu mở dấu ngoặc đơn chấm uh Văn bản giữ chỗ cú thé đậm chấm dấu chấm ngang sẽ là 8 dấu chấm đậm dấu chấm dọc đang diễn ra là 12 được rồi Và chúng tôi thực sự sẽ bọc nó trong một câu lệnh if và chúng ta sẽ nói nếu dấu chấm văn bản trống và sau đó chúng ta sẽ đặt nó ở đó và sau đó chúng ta sẽ nói văn bản trình chỉnh sửa là một giao diện người dùng nhanh thực sự thành phần và đây là nơi chúng ta phải vượt qua tài sản ràng buộc đó của văn bản của chúng tôi tôi sẽ cho nó một phần đậm của 4 điểm ảnh Và toàn bộ sẽ có một phông chữ của cơ thể đó ỗn vì vậy chúng tôi đang gặp lỗi ở đây, hãy xem oh chúng ta không thể đặt những dấu ngoặc đơn ở đó vì vậy bây giờ hãy tiếp tục và xây dựng nó và của chúng tôi thành phần tùy chỉnh nên được thực hiện chúng ta hãy tiếp tục và xem bây giờ nếu chúng ta có thể thả nó vào chế độ xem tweet mới của chúng tôi

Main subtitle

so on this h stack just go ahead and
uh add your text area
and we need to pass in a placeholder and
a text so here we're going to say what's
happening
for the placeholder and the text we
actually need to create that so we're
going to say at state private var
caption
and it's just going to be blank
and then here we're going to say dollar
sign caption
sweet so that's looking pretty good
um let's resume the preview
and okay so you guys notice that this is
like kind of huge right now
um
so i think we actually need to give it a
frame
or maybe
no sorry we don't actually
um because this this could actually just
be
this size on the h stack we can actually
change the alignment to dot top
and it puts that guy right on top of
there which looks great
and then on this h stack guys just go
ahead and add some padding
so that looks good
and this is now looking
as it should and let's go ahead and hit

Second subtitle

vì vậy trên ngăn xếp h này, hãy tiếp tục và
uh thêm vùng văn bản của bạn
và chúng ta cần chuyển vào một trình giữ
chỗ và
một văn bản vì vậy ở đây chúng ta sẽ nói
những gì
đang xảy ra
cho trình giữ chỗ và văn bản chúng tôi
thực sự cần phải tạo ra điều đó vì vậy
chúng tôi
sẽ nói ở bang private var
đầu đề
và nó sẽ trống rỗng
và sau đó ở đây chúng ta sẽ nói đó là
ký chú thích
ngọt ngào nên trông khá ổn
ừm, hãy tiếp tục xem trước
và được rồi để các bạn chú ý rằng đây là
giống như loại rất lớn ngay bây giờ
ô
vì vậy tôi nghĩ rằng chúng ta thực sự cần
phải cung cấp cho nó một
khung
hoặc có thể
không xin lỗi, chúng tôi không thực sự
ừm bởi vì điều này thực sự có thể chỉ
là
kích thước này trên ngăn xếp h chúng ta
thực sự có thể
thay đổi căn chỉnh thành dot top
và nó đặt anh chàng đó ngay trên đầu
ở đó trông thật tuyệt
và sau đó trên ngăn xếp h này, mọi người
chỉ cần đi
phía trước và thêm một số phần đệm
vì vậy có vẻ tốt
và điều này hiện đang tìm kiếm
như nó nên và hãy tiếp tục và đánh

Main subtitle

play

and start typing into this text for some reason the typing in my preview doesn't work

so really quickly before we run this on the simulator go back to your feed view

and replace this text here that says new

tweet view

with the actual new tweet view

and let's go ahead and run this and that should do it for

this section we'll be able to move on to our authentication stuff next

so that looks really good

um we do have to make it so that this that cancel button actually dismisses that view but let's make sure our text

area is working

so that looks really good and you guys notice that if i delete everything the placeholder shows back up so really quickly we can talk about how

that works

um but first let's make it so that we

this cancel button actually dismisses the view

so let me go back to new tweet view

we're gonna add an environment variable here

at environment bar oops at environment parentheses backslash dot presentation mode

Second subtitle

chơi

và bắt đầu gõ vào văn bản này cho một số lý do gõ trong bản xem trước của tôi không công việc

rất nhanh trước khi chúng tôi chạy cái này trình giả lập quay lại chế độ xem nguồn cấp dữ liệu của bạn

và thay thế văn bản này ở đây có nội dung mới

xem tweet

với chế độ xem tweet mới thực tế

và hãy tiếp tục và chạy cái này và cái kia nên làm điều đó cho

phần này chúng ta sẽ có thể chuyển sang công cụ xác thực của chúng tôi tiếp theo vì vậy trông thực sự tốt

ừm, chúng ta phải làm cho nó như thế này nút hủy đó thực sự loại bỏ

chế độ xem đó nhưng hãy đảm bảo rằng văn bản của chúng tôi

khu vực đang làm việc

vì vậy trông thực sự tốt và các bạn

lưu ý rằng nếu tôi xóa mọi thứ

trình giữ chỗ hiển thị sao lưu để

thực sự nhanh chóng chúng ta có thể nói về cách

nó hoạt động

ừm nhưng trước tiên hãy làm cho nó sao cho chúng ta

nút hủy này thực sự loại bỏ

quan điểm

vì vậy hãy để tôi quay lại chế độ xem tweet mới

chúng ta sẽ thêm một biến môi trường đây

tại thanh môi trường rất tiếc tại môi trường dấu gạch chéo ngược dấu chấm trình bày cách thức

Main subtitle

bar

presentation mode

and basically guys what this does is it's an environment variable that recognizes the presentation state of the application so it'll be able to recognize

that we have a view presented right now

so when we hit the cancel button it will dismiss that presented view

so here we just have to say

`Presentationmode.wrappedvalue.dismiss` and we can run this again and we should

be able to

toggle that new tweet view

up and down so this will present it this will cancel it present

cancel so that looks really good so really quickly i'll just explain this

text area and why we had to make it a

custom view

one more time before we move on to the

next

chapter

so let's open that text area up

if you guys want to know how to do that

you just hold down command shift o and you can just uh like sort of quick open uh any any one of your files and xcode it's really helpful

but

essentially guys once again we needed to

Second subtitle

quán ba

chế độ trình bày

và về cơ bản, những gì nó làm là đó là một biến môi trường nhận ra trạng thái trình bày của ứng dụng để nó có thể nhận ra

rằng chúng tôi có một cái nhìn được trình bày ngay bây giờ

vì vậy khi chúng tôi nhấn nút hủy, nó sẽ

loại bỏ quan điểm được trình bày

vì vậy ở đây chúng ta chỉ cần nói

`Presentationmode.wrappedvalue.dismiss`

và chúng ta có thể chạy lại cái này và chúng ta nên

có thể

chuyển chế độ xem tweet mới đó

lên và xuống vì vậy điều này sẽ trình bày nó sẽ hủy bỏ nó

hủy để trông thực sự tốt vì vậy

rất nhanh tôi sẽ chỉ giải thích điều này

vùng văn bản và tại sao chúng tôi phải biến nó thành một

theo cách nhìn thông thường

một lần nữa trước khi chúng ta chuyển sang phần

Kế tiếp

chương

vì vậy hãy mở vùng văn bản đó lên

nếu các bạn muốn biết làm thế nào để làm điều đó

bạn chỉ cần giữ phím shift o

và bạn có thể chỉ uh như loại nhanh chóng mở uh bất kỳ tệp nào của bạn và

xcode nó thực sự hữu ích

Nhưng

về cơ bản các bạn một lần nữa chúng ta cần phải

Main subtitle

create this because we want this to be a text view so the text will like wrap um i don't think you can do that with the text field but the problem is text views or in this case what swift ui calls them text editors they don't have placeholders so you basically have to make a custom view and build the placeholder functionality in yourself so basically it's a z stack right and you say if the text is empty which in this case right now it is it shows the placeholder and if the text is not empty which is what is happening here it'll show the actual text editor and we pass in a placeholder because now we can reuse this in other parts of the app if we wanted to and we could just pass in a different placeholder and a different text parameter um and it will it will like bind to the or like the uh the view that it's contained within so right now we could drag and drop this text area into any part of the app and it will act as like a standalone view

Second subtitle

tạo cái này bởi vì chúng tôi muốn cái này là một chế độ xem văn bản để văn bản sẽ giống như gói ừm tôi không nghĩ bạn có thể làm điều đó với trường văn bản nhưng vấn đề là văn bản lượt xem hoặc trong trường hợp này, Swift ui là gì gọi họ là người soạn thảo văn bản mà họ không có giữ chỗ, do đó về cơ bản bạn phải tạo chế độ xem tùy chỉnh và xây dựng chức năng giữ chỗ trong chính bạn nên về cơ bản nó là một ngăn xếp z đúng và bạn nói nếu văn bản trống mà trong trường hợp này ngay bây giờ nó là nó hiển thị trình giữ chỗ và nếu văn bản là không trống rỗng, đó là những gì đang xảy ra ở đây nó sẽ hiển thị trình soạn thảo văn bản thực tế Và chúng tôi vượt qua trong một trình giữ chỗ bởi vì bây giờ chúng ta có thể sử dụng lại cái này trong cái khác các phần của ứng dụng nếu chúng tôi muốn và chúng tôi chỉ có thể vượt qua trong một khác nhau trình giữ chỗ và một văn bản khác tham số um và nó sẽ như thế nào liên kết với hoặc thích uh quan điểm đó nó được chứa bên trong vì vậy ngay bây giờ chúng tôi có thể kéo và thả vùng văn bản này vào bất kỳ phần nào của ứng dụng và nó sẽ hoạt động giống như một chế độ xem độc lập

Main subtitle

which is really nice

so

this is why now it functions this way

which is exactly sort of what we want there so that's looking really good guys that's going to wrap it up for this chapter

in the next one we are going to now

finally start building out

this uh awesome

log out and sign up page so get excited for that guys it looks super slick and

that's going to wrap it up for the ui

once we're done with that then we'll be

able to start hooking up the back end and work with some actual data which is going to be really fun

last but not least guys we are going to

be building this login and sign up page

so let's just jump right into it go to your core folder and we're going to create another group here and this is going to be called authentication

uh and then within that we're going to

create two other subgroups

one for view model

and one for views

and in the views folder

we're going to create two swift ui views

we're going to create

our login view

Second subtitle

cái nào thực sự tốt

Vì thế

đây là lý do tại sao bây giờ nó hoạt động theo cách này

đó chính xác là những gì chúng ta muốn ở đó trông rất đẹp đấy các bạn điều đó sẽ kết thúc nó cho việc này chương

trong phần tiếp theo chúng ta sẽ đến bây giờ

cuối cùng bắt đầu xây dựng

điều này thật tuyệt vời

đăng xuất và đăng ký trang để có hứng thú đối với những người đó, nó trông siêu bong bầy và

điều đó sẽ kết thúc nó cho giao diện người dùng

một khi chúng ta hoàn thành việc đó thì chúng ta sẽ

có thể bắt đầu kết nối mặt sau

và làm việc với một số dữ liệu thực tế

sẽ rất vui

cuối cùng nhưng không kém phần quan trọng, chúng ta sẽ

đang xây dựng trang đăng nhập và đăng ký này

vì vậy chúng ta hãy nhảy ngay vào nó thư mục cốt lõi của bạn và chúng tôi sẽ tạo một nhóm khác tại đây và điều này sẽ được gọi là xác thực

uh và sau đó chúng ta sẽ

tạo hai nhóm con khác

một cho mô hình xem

và một cho lượt xem

và trong thư mục lượt xem

chúng ta sẽ tạo hai chế độ xem Swift ui chúng ta sẽ tạo ra

chế độ xem đăng nhập của chúng tôi

Main subtitle

and we are also going to create our registration view just like that okay cool so let's start with this login view and resume on your preview just to make sure that it's connected and we can start building out our ui once that loads awesome so uh first we're going to create this like sort of header view right here so really quickly guys let's go ahead and just make a v stack and it's going to within that have um another v stack and that's because uh this is going to be its own v stack and then all of this other stuff is a v stack as well so this is sort of like the parent we call it parent container and then this guy will be the header view okay just adding those comments to clarify everything on this header view we're going to create an alignment of dot leading and i'm just going to go ahead and make a text that says hello

Second subtitle

và chúng tôi cũng sẽ tạo ra xem đăng ký cứ như vậy được rồi, tuyệt, vậy hãy bắt đầu với chế độ xem đăng nhập này và tiếp tục trên bản xem trước của bạn chỉ để thực hiện chắc chắn rằng nó được kết nối và chúng ta có thể bắt đầu xây dựng giao diện người dùng của mình một khi tải tuyệt vời Vì thế uh trước tiên chúng ta sẽ tạo cái này như thế nào loại xem tiêu đề ngay tại đây Vì thế thật nhanh các bạn hãy tiếp tục và chỉ cần tạo một ngăn xếp v và nó sẽ đi vào bên trong đó có ừm ngăn xếp v khác và đó là bởi vì uh đây sẽ là ngăn xếp v của riêng nó và sau đó tất cả những thứ khác này là một v ngăn xếp cũng như vậy, đây là loại giống như cha mẹ chúng tôi gọi nó là vùng chứa chính và sau đó anh chàng này sẽ là chế độ xem tiêu đề được rồi, chỉ cần thêm những nhận xét đó vào làm rõ mọi thứ trên chế độ xem tiêu đề này chúng ta sẽ tạo ra một sự liên kết của dấu chấm hàng đầu Và tôi sẽ tiếp tục và thực hiện một văn bản nói xin chào

Main subtitle

dot or period

and then

we're we're gonna give it a font of dot

large oops dot fonts

dot large title

dot font weight dot semi bold

okay so that looks good

then we're gonna give it a text that

says welcome back

dot font dot large title

dot font weight dot semi bold

awesome

okay and then guys on this v stack here

we're going to give it a

dot frame

of height

which is 260.

okay so that looks good

and then we are going to give it a

background

of color parenthesis dot system blue

looking good we're going to give it a

foreground color of white

looking better

and then we need to give it this like

custom clip shape and we need to make it

fill the width of the screen

okay so what we can actually do to make

it fill the width of the screen there's

two things you can do

i'm sure there's other ways of doing
this as well you guys can just go here
and make an h stack

Second subtitle

dấu chấm hoặc dấu chấm

và sau đó

chúng tôi sẽ cung cấp cho nó một phông
chữ dấu chấm

rất tiếc phông chữ chấm lớn

chấm tiêu đề lớn

dấu chấm chữ đậm dấu chấm bán đậm

được rồi để có vẻ tốt

sau đó chúng tôi sẽ cung cấp cho nó một
văn bản

nói chào mừng trở lại

phông chữ chấm chấm tiêu đề lớn

dấu chấm chữ đậm dấu chấm bán đậm

tuyệt vời

được rồi các bạn trên ngăn xếp v này ở đây

chúng tôi sẽ cung cấp cho nó một

khung chấm

chiều cao

đó là 260.

được rồi để có vẻ tốt

và sau đó chúng tôi sẽ cung cấp cho nó một
lý lịch

dấu ngoặc đơn màu xanh lam

nhin tốt, chúng tôi sẽ cung cấp cho nó một
màu nền trước màu trắng

trông đẹp hơn

và sau đó chúng ta cần phải cung cấp cho
nó như thế này

hình dạng clip tùy chỉnh và chúng tôi cần
làm cho nó

lấp đầy chiều rộng của màn hình

được rồi, vậy những gì chúng ta thực sự có
thể làm để tạo ra

nó lấp đầy chiều rộng của màn hình có

hai điều bạn có thể làm

tôi chắc rằng có những cách khác để làm
cái này cũng vậy các bạn có thể vào đây
và tạo một ngăn xếp h

Main subtitle

and create a spacer just like that and you guys will notice that it will just stretch it out to fit the entire screen an alternative would be to make this a

z-stack or just embed this in a z-stack

but this is fine as well it's just a little trick that i like to use um i believe that if you tried to give it a width of like ui screen

main.bound

and like where to comment this out i don't think that works

oh it does but uh the problem is is that

um

i don't know how to get the text to

align to the leading edge when you do it

like that so it's a little bit weird

so i like to just make it this h stack here

and everything aligns perfectly

and then we're just going to give it some padding guys we're going to say dot padding

dot leading just like that

and that'll stretch it out there nicely

and then

um within our parent container for now we're just going to add a spacer and then on this parent container

as well we're going to say dot ignores safe area

Second subtitle

và tạo một miếng đệm giống như vậy và các bạn sẽ nhận thấy rằng nó sẽ chỉ kéo dài nó ra để vừa với toàn bộ màn hình một giải pháp thay thế sẽ là biến điều này thành một

z-stack hoặc chỉ cần nhúng cái này vào z-stack

nhưng điều này cũng tốt thôi, nó chỉ là một mẹo nhỏ mà tôi thích sử dụng um i tin rằng nếu bạn cố gắng

để cung cấp cho nó chiều rộng như màn hình ui chính. giới hạn

và thích nơi để bình luận điều này tôi không nghĩ rằng nó hoạt động oh nó có nhưng uh vấn đề là đó là ô

tôi không biết làm thế nào để có được văn bản

căn chỉnh với cạnh hàng đầu khi bạn làm điều đó

như thế thì hơi lạ

vì vậy tôi muốn làm cho nó ngăn xếp h này đây

và mọi thứ sắp xếp hoàn hảo

và sau đó chúng tôi sẽ cung cấp cho nó một số kẽ đệm chúng ta sẽ nói dấu chấm đệm

dấu chấm hàng đầu như thế

và điều đó sẽ kéo dài nó ra khỏi đó một cách độc đáo

và sau đó

um trong vùng chứa chính của chúng tôi bây giờ chúng ta sẽ thêm một

miếng đệm và sau đó trên vùng chứa chính này

cũng như chúng ta sẽ nói dấu chấm bỏ qua khu vực an toàn

Main subtitle

and that will get it to fill out into the top of the screen so that looks pretty good right i do however want to show you guys how to get this like sort

of custom shape here

it's actually pretty simple we're just going to create a custom view

or a view modifier rather

that's going to help us do that and it's uh really awesome because we're going to create it so that we can round whatever

edges of a shape that we want so you can

pick one two three or four edges to round in this case we're just rounding

out one of the edges it's the bottom right edge so

i'm going to go into my components folder guys uh let's move that up there um create a new file

it's going to be a

uh we can just make it a swift file

actually

and we're gonna call it rounded shape and we're gonna import swift ui up at the top

and we're gonna say uh struct

rounded shape and it's gonna inherit from the shape class and we're going to create some corners which is a ui rect corner

Second subtitle

và điều đó sẽ khiến nó điền vào phía trên cùng của màn hình để trông khá tốt đúng không tôi làm tuy nhiên muốn chỉ cho các bạn làm thế nào để có được loại như thế này

của hình dạng tùy chỉnh ở đây

nó thực sự khá đơn giản, chúng tôi chỉ sẽ tạo một tùy chỉnh xem

hoặc một công cụ sửa đổi chế độ xem thay vì

điều đó sẽ giúp chúng tôi làm điều đó và nó uh thực sự tuyệt vời bởi vì chúng ta sẽ tạo ra nó để chúng ta có thể làm tròn bất cứ điều gì

các cạnh của hình dạng mà chúng tôi muốn để bạn có thể

chọn một hai ba hoặc bốn cạnh để làm tròn trong trường hợp này chúng ta chỉ làm tròn

ra khỏi một trong các cạnh đó là đáy cạnh phải như vậy

tôi sẽ đi vào các thành phần của tôi thư mục các bạn uh hãy di chuyển nó lên đó ừm tạo một tập tin mới nó sẽ là một

uh chúng ta có thể biến nó thành một tập tin nhanh chóng

Thực ra

và chúng ta sẽ gọi nó là hình tròn

và chúng tôi sẽ nhập swift ui lên tại đĩnh

và chúng ta sẽ nói uh cấu trúc

hình tròn và nó sẽ kế thừa từ lớp hình dạng

và chúng ta sẽ tạo một số góc đó là góc ui rect

Main subtitle

see this is the corners of a rectangle
and this is going to be
how we're able to tell our shape which
corners to round out
so then we need to have this path
function
and we're going to say let path equal ui
bezier path
uh let's see what do we want we want
this rounded rect
wrecked
by rounding corners we pass in our
corners variable
and corner radii is going to be cg size
with
80 by 80.

and we're going to return a path
path.cgpath
so this is just a custom
modifier that i like to have in all of
my projects
and like i said it will allow you to
just round out whatever edges of a shape
you want so we're going to actually go
ahead and see how to use this now it's
really really cool
so let's go back to our login view
and on this v stack here
we're just going to apply
a clip shape
of our rounded shape
and it's going to ask us for the corners

Second subtitle

xem đây là các góc của một hình chữ nhật
và đây sẽ là
làm thế nào chúng ta có thể nói hình dạng
của chúng tôi mà
các góc để làm tròn
vì vậy sau đó chúng ta cần phải có con
đường này
chức năng
và chúng ta sẽ nói let path equal ui
con đường bezier
uh hãy xem những gì chúng ta muốn chúng
ta muốn
trực tràng tròn này
bị đắm
bằng cách làm tròn các góc, chúng tôi vượt
qua
biến góc
và bán kính góc sẽ có kích thước cg
với
80 nhân 80.
và chúng ta sẽ trả về một con đường
đường dẫn.cgpath
vì vậy đây chỉ là một phong tục
công cụ sửa đổi mà tôi muốn có trong tất cả
những dự án của tôi
và như tôi đã nói nó sẽ cho phép bạn
chỉ cần làm tròn bất kỳ cạnh nào của hình
dạng
bạn muốn vì vậy chúng tôi sẽ thực sự đi
trước và xem cách sử dụng cái này bây giờ
thực sự thực sự mát mẻ
Vì vậy, hãy quay lại chế độ xem đăng nhập
của chúng tôi
và trên ngăn xếp v này ở đây
chúng tôi sẽ áp dụng
một hình dạng clip
hình dạng tròn của chúng tôi
và nó sẽ yêu cầu chúng tôi cho các góc

Main subtitle

and you guys can just open up your parentheses and say dot bottom right

and let's go ahead and resume our preview and you guys will notice that that

rounds out the bottom right edge just like that and that looks really really good guys

um and you'll notice that i could also

do like dot bottom

left

and it would round out this whole thing

obviously that looks kind of silly

i don't really like that i think it's

just better to have it

with one

edge rounded out i do think that looks really good

um if you guys wanted to mess with how rounded it gets

you could just go back to

that component

and you could adjust this stuff if you

wanted to wanted it to be fully

customizable you could add another input parameter for what you want the corner radius or radii to be for that

because the way it's set up now

every time you use this clip shape it

would have this sort of static

corner radii if you wanted to be able to customize that you could just add that up here as another input parameter and

Second subtitle

và các bạn chỉ có thể mở cửa bạn

dấu ngoặc đơn và nói dấu chấm phía dưới bên phải

và hãy tiếp tục và tiếp tục

xem trước và các bạn sẽ nhận thấy rằng cái đó

làm tròn cạnh dưới bên phải

chỉ như vậy và điều đó trông thực sự những người thực sự tốt

ùm và bạn sẽ nhận thấy rằng tôi cũng có thể

làm như chấm đáy

bên trái

và nó sẽ hoàn thiện toàn bộ điều này

rõ ràng là có vẻ ngắn

tôi không thực sự thích điều đó tôi nghĩ nó chỉ tốt hơn để có nó

với một

cạnh được làm tròn tôi nghĩ rằng nó trông thật sự tốt

ùm nếu các bạn muốn gây rối với cách

làm tròn nó được

bạn chỉ có thể quay trở lại

thành phần đó

và bạn có thể điều chỉnh công cụ này nếu bạn

muốn nó được đầy đủ

tùy chỉnh bạn có thể thêm một đầu vào khác tham số cho những gì bạn muốn góc

bán kính hoặc bán kính cho điều đó

bởi vì cách nó được thiết lập bây giờ

mỗi khi bạn sử dụng clip này, hãy định hình nó

sẽ có loại tĩnh này

bán kính góc nếu bạn muốn có thể

tùy chỉnh mà bạn chỉ có thể thêm nó

lên đây như một tham số đầu vào khác và

Main subtitle

pass it in as whatever you want um

whenever you're creating a shape in

different places in your application

um okay so now that that's done

and it looks pretty darn good

we're going to keep going

with uh

these custom text fields down here so

this login page does have a lot of

custom

views guys um and it does take a little

bit of time to build that stuff out but

i think it's really worth it to have

these custom views that you can reuse

across any application that you're

building

so

let's go ahead and make another v stack

and this is going to be for these two

text fields here

um we're going to give it a spacing

parameter of 40.

and we are going to want to create some

custom text fields here

um

but for now let's just go ahead and make

them text fields

um we're just going to want a title and

a text so it's going to be email

and text we need that actual text

property so we're going to go up here

and say at state private bar email

Second subtitle

chuyển nó vào như bắt cứ điều gì bạn muốn

um

bắt cứ khi nào bạn đang tạo một hình dạng

trong

những nơi khác nhau trong ứng dụng của

bạn

ùm được rồi, bây giờ điều đó đã xong

và nó có vẻ khá tốt

chúng ta sẽ tiếp tục đi

với uh

các trường văn bản tùy chỉnh này ở đây để

trang đăng nhập này có rất nhiều

phong tục

lượt xem các bạn um và nó mất một chút

một chút thời gian để xây dựng những thứ

đó nhưng

tôi nghĩ nó thực sự đáng để có

các chế độ xem tùy chỉnh này mà bạn có

thể sử dụng lại

trên bất kỳ ứng dụng nào mà bạn đang

xây dựng

Vì thế

hãy tiếp tục và tạo một ngăn xếp v khác

và cái này sẽ dành cho hai người này

trường văn bản ở đây

ùm, chúng ta sẽ tạo khoảng cách cho nó

thông số 40.

và chúng tôi sẽ muốn tạo ra một số

trường văn bản tùy chỉnh ở đây

ô

nhưng bây giờ chúng ta hãy tiếp tục và thực

hiện

chúng trường văn bản

ùm, chúng tôi sẽ chỉ muốn một tiêu đề và

một văn bản vì vậy nó sẽ là email

và văn bản chúng tôi cần văn bản thực tế đó

tài sản vì vậy chúng tôi sẽ đi lên đây

và nói tại email thanh trạng thái riêng tư

Main subtitle

equals a blank string and let's just go

ahead and make the password variable as well

and they're both just blank strings

okay now we can pass in our email text

property

and we can just go ahead and copy this

paste it down below

this will say password

and this will be dollar sign password

and let's go ahead and resume our

preview and see how that looks

um we will come back and customize these

in a little bit i kind of want to just

get the rest of this view built out

so let's see try

again

okay so looks not great but um we're

gonna adjust

or add some modifiers to clean it up a

little bit so we're gonna say dot

padding

dot horizontal

32

and then padding

uh dot top

44.

all right so that is looking a little

bit better and like i said guys we're

going to customize this view to have

like an image and like this underline

view and we're going to make a custom

Second subtitle

bằng một chuỗi trống và chúng ta hãy bắt đầu

phía trước và biến mật khẩu thành Tốt

và cả hai đều chỉ là những chuỗi trống được rồi bây giờ chúng ta có thể chuyển văn bản email của mình

tài sản

và chúng ta có thể tiếp tục và sao chép cái này

dán nó xuống bên dưới

điều này sẽ nói mật khẩu

và đây sẽ là mật khẩu ký hiệu đô la

và hãy tiếp tục và tiếp tục

xem trước và xem nó trông như thế nào

ừm, chúng tôi sẽ quay lại và tùy chỉnh những thứ này

trong một chút tôi muốn chỉ

xây dựng phần còn lại của chế độ xem này

vì vậy hãy xem thử

lại

được rồi, trông không đẹp lắm nhưng ừm chúng tôi

sẽ điều chỉnh

hoặc thêm một số sửa đổi để làm sạch nó

một chút vì vậy chúng ta sẽ nói dấu chấm

đệm

chấm ngang

32

và sau đó đệm

uh chấm đầu

44.

được rồi vì vậy điều đó đang tìm kiếm một chút

tốt hơn một chút và như tôi đã nói các bạn sẽ tùy chỉnh chế độ xem này để có

thích một hình ảnh và thích gạch chân này xem và chúng tôi sẽ thực hiện một tùy chỉnh

Main subtitle

sub view or sorry sub class for that

view so that we can reuse it in

different places in the app

we're going to be reusing this in the registration view as well but for now let's just keep going we'll create this

forgot password guy

so this is going to actually be an h stack

because we're going to use a spacer to

push this all the way to the right so we're going to create a spacer here and then we're just going to create a navigation link

destination label

this is just going to say text like

reset password view

and the label

is just going to be a text

property that says forgot password

uh we're going to say dot font

dot

caption

font weight dot bold

and then we're going to give it a

foreground color

of color dot system blue

see how that looks

uh pretty good uh actually add a

question mark at the end of forgot

password now we just need to give it

some padding guys

so we're gonna say dot padding dot top

Second subtitle

sub view hay xin lỗi sub class vì điều đó xem để chúng tôi có thể sử dụng lại nó trong

những nơi khác nhau trong ứng dụng chúng ta sẽ sử dụng lại cái này trong xem đăng ký là tốt nhưng bây giờ chúng ta hãy tiếp tục, chúng ta sẽ tạo ra cái này

anh quên mật khẩu

vì vậy đây thực sự sẽ là một h

cây rơm

bởi vì chúng tôi sẽ sử dụng một miếng đệm để

đẩy cái này sang bên phải

chúng ta sẽ tạo một miếng đệm ở đây

và sau đó chúng ta sẽ tạo ra một

liên kết điều hướng

nhãn đích

điều này sẽ chỉ nói văn bản như

đặt lại chế độ xem mật khẩu

và nhãn

sẽ chỉ là một văn bản

tài sản cho biết quên mật khẩu

uh chúng ta sẽ nói phông chữ dấu chấm

dấu chấm

đầu đề

trọng lượng phông chữ chấm đậm

và sau đó chúng tôi sẽ cung cấp cho nó một màu nền

hệ thống chấm màu xanh lam

xem nó trông như thế nào

uh khá tốt uh thực sự thêm một

dấu chấm hỏi ở cuối quên

mật khẩu bây giờ chúng ta chỉ cần cung cấp cho nó

một số kẻ đệm

vì vậy chúng ta sẽ nói dấu chấm đậm dấu chấm trên cùng

Main subtitle

dot padding dot trailing

24.

okay so that's looking pretty good let's actually make the font weight semi-bold

right i think that looks a little bit better

sweet so that's looking pretty good now we just have to do this sign up button so now what we're going to do is go

below this h stack and we're going to create a button

action label

we're going to just print sign in here

and then the label is going to be a text property that says sign in

dot font

dot headline

that foreground color is dot white

dot frame

width height

so we're going to do 340 for the width and 50 for the height

then we're going to give it a background

of color parentheses dot system blue

dot clip shape

is a capsule

and then we're just going to give it some padding

and let's see how that looks so that looks great

um and we are going to give this a very

slight shadow guys it just makes it pop a little bit more and it's very very

Second subtitle

dấu chấm đậm dấu chấm

24.

được rồi, vậy là trông khá ổn, chúng ta hãy thực sự làm cho trọng lượng phông chữ bán đậm

đúng tôi nghĩ rằng nó có vẻ một chút tốt hơn

ngọt ngào vì vậy bây giờ trông khá tốt chúng ta chỉ cần thực hiện nút đăng ký này vì vậy bây giờ những gì chúng ta sẽ làm là đi

bên dưới ngăn xếp h này và chúng ta sẽ tạo một nút

nhãn hành động

chúng ta sẽ chỉ in

đăng nhập tại đây

và sau đó nhãn sẽ là một văn bản tài sản cho biết đăng nhập

phông chữ chấm

chấm tiêu đề

màu nền trước đó là chấm trắng

khung chấm

chiều rộng chiều cao

vì vậy chúng ta sẽ làm 340 cho chiều rộng và 50 cho chiều cao

sau đó chúng ta sẽ cung cấp cho nó một nền tảng

dấu ngoặc đơn màu xanh lam

hình dạng kẹp chấm

là một viên nang

và sau đó chúng tôi sẽ cung cấp cho nó một số phần đậm

và hãy xem nó trông như thế nào

Trông rất tuyệt

ừm và chúng tôi sẽ cung cấp cho điều này rất

các bạn bóng nhẹ nó chỉ làm cho nó bật lên một chút nữa và nó rất rất

Main subtitle

subtle

and you might not notice it in terms of it being visible to like the naked eye right away but it is there and it actually does add a little bit of very subtle dimension to your ui and make it

look like it pops out a little bit more so we're going to say dot shadow color is going to be dot gray
dot opacity oops
dot opacity

0.5 radius is 10 x 0 y 0.

and you guys could like change the just to see that it's actually showing up
you could mess with some of these properties here like if you took off this opacity

it's a little bit more visible you can kind of see that gray shadow like if i make it red

you can see that red shadow right there but then if you give it that opacity back

it becomes much more faint and much more

subtle but it is still there
so just change it back to gray
and um i always like to do that with my buttons to make it look like they pop

out a little bit more anyway now we just need to uh create this uh spit like little button down here that's gonna take us over to our registration view

Second subtitle

tế nhị

và bạn có thể không nhận thấy nó về mặt nó có thể nhìn thấy bằng mắt thường ngay lập tức nhưng nó ở đó và nó thực sự có thêm một chút rất

kích thước tinh tế cho giao diện người dùng của bạn và làm cho nó có vẻ như nó bật ra thêm một chút nữa vì vậy chúng ta sẽ nói chấm bóng màu sẽ là chấm xám độ mờ của dấu chấm rất tiếc độ mờ chấm

0,5 bán kính là 10 x 0 y 0.

và các bạn có thể thích thay đổi chỉ để thấy rằng nó thực sự xuất hiện
bạn có thể gây rối với một số trong số này tài sản ở đây như thể bạn cắt cánh độ mờ này

nó rõ ràng hơn một chút bạn có thể kiểu như nhìn thấy bóng xám đó như thể tôi làm cho nó đỏ

bạn có thể thấy bóng đỏ đó ngay tại đó nhưng sau đó nếu bạn cho nó độ mờ mặt sau

nó trở nên mờ nhạt hơn nhiều và nhiều hơn nữa
tinh tế nhưng nó vẫn ở đó
vì vậy chỉ cần thay đổi nó trở lại màu xám và ừm tôi luôn thích làm điều đó với tôi các nút để làm cho nó trông giống như chúng bắt

ra thêm một chút dù sao bây giờ chúng ta chỉ
uh cần phải tạo cái này uh
nhỏ như nút nhỏ xuống đây đó là
sẽ đưa chúng tôi đến đăng ký của chúng tôi xem

Main subtitle

so below the spacer guys that's within the parent v stack container we're going to add this navigation link the destination will be our registration view excuse me and on the registration view guys you're going to say dot navigation bar hidden is true because we do not want a navigation bar on this guy um we're going to have this custom header and our label is just going to be an h stack it's going to say text don't have an account dot font dot system size or we can just give it dot caption actually one sec guys okay and then we're gonna say text sign up dot font dot caption dot font weight dot semi bold now we give this some uh padding on the bottom right so that looks pretty good and let's actually give this a font of body caption's a little small uh and that looks a little too big now let's try footnote foot nope

Second subtitle

vì vậy bên dưới những người đệm trong đó bộ chứa ngăn xếp cha v chúng tôi sẽ thêm liên kết điều hướng này điểm đến sẽ là đăng ký của chúng tôi xem thứ lỗi tôi và các bạn vào xem đăng ký bạn sẽ nói thanh điều hướng dấu chấm ẩn là sự thật bởi vì chúng tôi không muốn có một thanh điều hướng về anh chàng này ừm chúng ta sẽ có cái này tiêu đề tùy chỉnh và nhãn của chúng tôi sẽ chỉ là một h cây rơm nó sẽ nói văn bản không có một tài khoản kích thước hệ thống chấm phông chữ chấm hoặc chúng ta chỉ có thể cung cấp cho nó chú thích dấu chấm thực sự một giây nhé các bạn được rồi chúng ta sẽ nói dấu hiệu văn bản hướng lên chấm phông chữ chấm tiêu đề dấu chấm chữ đậm dấu chấm bán đậm bây giờ chúng tôi cung cấp cho một số uh đệm ở phía dưới đúng như vậy trông khá tốt và hãy thực sự cung cấp cho phông chữ này chú thích cơ thể hơi nhỏ uh và bây giờ trông hơi quá lớn hãy thử chú thích chân không

Main subtitle

okay footnote looks uh good to me

sweet okay so now we just need to give

it a foreground color

with color

dot system blue

and that's looking pretty clean guys

and also on the navigation bar here

we're gonna say dot navigation bar

hidden

is true so that'll hide it on our um

login view as well so let's actually go

ahead and make this the root view of our

application and then we'll be able to

see it coming up on the simulator and

see how this looks and see if this

navigation link is working

so let's see let's go to our youtube app

file

and replace content view

with login view

and run it

so it should look pretty similar to that

we just won't have those custom text

fields

okay so yeah that looks pretty darn good

and we noticed if we hit the sign up

button it takes us over to this or what

our registration view is currently set

up as uh to go back guys you can just

hold the edge of the screen and i think

Second subtitle

được rồi, chú thích cuối trang có vẻ tốt với tôi

ngọt ngào được rồi vì vậy bây giờ chúng ta chỉ cần cho

nó là một màu nền trước

với màu sắc

hệ thống chấm màu xanh

và đó là trông khá sạch sẽ guys

và cả trên thanh điều hướng ở đây

chúng ta sẽ nói thanh điều hướng dấu chấm ẩn giấu

là đúng vì vậy sẽ ẩn nó trên um của chúng tôi

chế độ xem đăng nhập cũng vậy, hãy thực sự bắt đầu

phía trước và biến đây thành chế độ xem gốc của chúng tôi

ứng dụng và sau đó chúng tôi sẽ có thể

thấy nó xuất hiện trên trình giả lập và

xem cái này trông như thế nào và xem cái này có

liên kết điều hướng đang hoạt động

vì vậy hãy xem chúng ta hãy truy cập ứng dụng youtube của chúng tôi

tài liệu

và thay thế chế độ xem nội dung

với chế độ xem đăng nhập

và chạy nó

vì vậy nó sẽ trông khá giống với điều đó

chúng tôi sẽ không có những văn bản tùy chỉnh đó

lĩnh vực

được rồi, vâng, nó trông khá đẹp

và chúng tôi nhận thấy nếu chúng tôi nhấn đăng ký

nút nó đưa chúng ta đến cái này hay cái gì

chế độ xem đăng ký của chúng tôi hiện

được đặt

up as uh to go back guys bạn chỉ có thể

giữ cạnh của màn hình và tôi nghĩ

Main subtitle

you can just drag
it's actually not working
okay we'll fix that later
obviously um right now what i want us to
do is actually
go ahead and build out these custom text
fields so once again back in your
components we're going to create another
file
and it's going to be a swift ui view
and we're going to just call this custom
text to be able to input fields
okay so let's go ahead and see how we're
going to create this guy let me get this
open on my code
okay so we need a couple things guys uh
we need to be able to pass in an image
name so that this can have an im a
unique image for each field uh we need
to pass into placeholder texts because
that's also going to be unique whenever
we want to create one of these things
and then um we also need a binding
text property so let's go ahead and
create that stuff we're going to say let
image name is a string
let placeholder text
to string
and let or sorry at binding
var text

Second subtitle

bạn chỉ có thể kéo
nó thực sự không hoạt động
được rồi, chúng tôi sẽ sửa nó sau
rõ ràng là um ngay bây giờ những gì tôi
muốn chúng tôi
làm là thực sự
hãy tiếp tục và xây dựng những văn bản tùy
chỉnh này
các trường để một lần nữa trở lại trong bạn
các thành phần chúng ta sẽ tạo ra một
thành phần khác
tài liệu
và nó sẽ là một giao diện người dùng nhanh
chóng
và chúng ta sẽ chỉ gọi đây là tùy chỉnh
văn bản để có thể nhập các trường
được rồi, hãy tiếp tục và xem chúng ta thế
nào
sẽ tạo ra anh chàng này để tôi lấy cái này
mở mã của tôi
được rồi vì vậy chúng tôi cần một vài điều
các bạn uh
chúng ta cần để có thể vượt qua trong một
hình ảnh
tên để cái này có thể có im a
hình ảnh duy nhất cho mỗi lĩnh vực uh
chúng tôi cần
để chuyển vào văn bản giữ chỗ vì
đó cũng sẽ là duy nhất bắt cứ khi nào
chúng tôi muốn tạo ra một trong những thứ
này
và sau đó um chúng tôi cũng cần một ràng
buộc
thuộc tính văn bản, vì vậy hãy tiếp tục và
tạo ra những thứ mà chúng ta sẽ nói hãy để
tên hình ảnh là một chuỗi
để văn bản giữ chỗ
thành chuỗi
và để hoặc xin lỗi khi ràng buộc
văn bản var

Main subtitle

is also a string

and this is going to give us some

problems

with our preview so let's actually just

go ahead and add these properties
we can go here and say envelope for the
image name

placeholder text we could just make it
email

and the binding text you can just say
dot constant

and give it a blank text
just like that

so now if we resume our preview

um

it should just say hello world so that's
fine um now let's go ahead and make this

guy so it's going to be within a v stack
and then within that we have an h stack

and we're going to have our image which
is system name

image name oops

dot resizable

dot scaled to fit

dot frame

width and height

of 20 by 20

dot foreground color and we're going to
give it a color of open parenthesis dot
dark gray

okay so that should be oh i spelled
envelope wrong there we go so that's
showing up nicely

Second subtitle

cũng là một chuỗi

và điều này sẽ cung cấp cho chúng tôi một
số

các vấn đề

với bản xem trước của chúng tôi, vì vậy hãy
thực sự chỉ

hãy tiếp tục và thêm các thuộc tính này
chúng ta có thể đến đây và nói phong bì cho
Tên Hình ảnh

văn bản giữ chỗ chúng ta có thể làm cho nó
e-mail

và văn bản ràng buộc bạn chỉ có thể nói
hàng số chấm

và cung cấp cho nó một văn bản trống
cứ thế

vì vậy bây giờ nếu chúng tôi tiếp tục xem
trước của mình

ô

nó chỉ nên nói xin chào thế giới vậy thôi
tốt thôi, bây giờ chúng ta hãy tiếp tục và làm
điều này

vì vậy nó sẽ nằm trong ngăn xếp v
và sau đó trong đó chúng ta có một ngăn
xếp h

và chúng ta sẽ có hình ảnh của chúng ta
là tên hệ thống

tên hình ảnh rất tiếc

dấu chấm có thể thay đổi kích thước
chấm thu nhỏ để phù hợp
khung chấm

chiều rộng và chiều cao
của 20 nhân 20

chấm màu nền trước và chúng ta sẽ
cho nó một màu của dấu ngoặc đơn mở
màu xám đen

được rồi, đó nên là oh tôi đánh vần
phong bì sai rồi, chúng ta đi vậy đó
xuất hiện độc đáo

Main subtitle

all right now

um

we're going to go here and create a text field

and the title key is going to be our placeholder text

and the binding property will just be our text

right so that's our text field that's

looking pretty good

now what we're going to have guys is

a divider below that h stack we're going

to say just divider

dot background

color open parenthesis dot dark gray

okay so that looks pretty good

now we can go ahead and create or add

these custom input fields on our login

page so we're gonna go to our login view

and we're gonna replace uh these text

fields here

with our custom input field so we're

gonna say custom input fields

image name is going to be envelope

placeholder text is email text is dollar

sign email

and i'm just going to

divide that up just like that and let's

go ahead and hit resume

and see if it's looking good

Second subtitle

tất cả ngay bây giờ

ô

chúng ta sẽ đến đây và tạo một văn bản
cánh đồng

và khóa tiêu đề sẽ là của chúng tôi

Văn bản giữ chỗ

và thuộc tính ràng buộc sẽ chỉ là
văn bản của chúng tôi

đúng vậy đó là trường văn bản của chúng
tôi đó là

nhìn khá tốt

bây giờ những gì chúng ta sẽ có các chàng
trai là

một dải phân cách bên dưới ngăn xếp h đó
chúng ta sẽ đi

để nói chỉ chia

chấm nền

màu mờ dấu ngoặc đơn chấm màu xám
đậm

được rồi vì vậy trông khá tốt

bây giờ chúng ta có thể tiếp tục và tạo hoặc
thêm

các trường nhập tùy chỉnh này khi đăng
nhập của chúng tôi

trang để chúng tôi sẽ chuyển đến chế độ
xem đăng nhập của chúng tôi

và chúng ta sẽ thay thế uh những văn bản
này

lĩnh vực ở đây

với trường đầu vào tùy chỉnh của chúng tôi,
vì vậy chúng tôi

sẽ nói các trường nhập tùy chỉnh

tên hình ảnh sẽ là phong bì

văn bản giữ chỗ là văn bản email là ô la
ký email

và tôi sẽ

chia nó ra như vậy và chúng ta hãy
tiếp tục và nhấn tiếp tục

và xem nó có đẹp không

Main subtitle

so we should notice that our email guy looks different yep that looks pretty good to me and then let's uh do the same thing for our password so just go ahead and copy and paste that we're going to create this and say it's lock i believe and this guy is going to say password and this is going to be dollar sign password um all right yep so that looks pretty pretty darn good right um i do think that we should run this one more time just to see how it looks and now that we're done with that we'll be able to move on to creating our registration page which is going to go really really quickly because we just did a lot of the heavy lifting here alright guys so this looks phenomenal right that's one of the cleanest looking login pages i've ever seen built by yours truly you're welcome now you guys can literally drag and drop this login page into any one of your apps that you have you can replace this

Second subtitle

vì vậy chúng ta nên chú ý rằng anh chàng email của chúng ta trông khác yep trông đẹp đât tốt với tôi và sau đó chúng ta hãy làm điều tương tự cho mật khẩu của chúng tôi vì vậy chỉ cần tiếp tục và sao chép và dán nó chúng ta sẽ tạo cái này và nói nó là khóa tôi tin và anh chàng này sẽ nói mật khẩu và đây sẽ là ký hiệu đô la mật khẩu ừm được rồi, yep trông rất đẹp khá darn tốt phải không ừm tôi nghĩ rằng chúng ta nên chạy cái này một lần nữa chỉ để xem nó trông như thế nào Và Hiện nay rằng chúng tôi đã hoàn thành với điều đó chúng tôi sẽ có thể để chuyển sang tạo đăng ký của chúng tôi trang sẽ thực sự đi một cách nhanh chóng bởi vì chúng tôi vừa làm rất nhiều nâng vật nặng ở đây được rồi các bạn nên cái này trông thật phi thường, đó là một trong những các trang đăng nhập sạch nhất mà tôi từng có nhìn thấy được xây dựng bởi bạn thực sự bạn được chào đón bây giờ các bạn có thể kéo và thả theo đúng nghĩa đen trang đăng nhập này vào bất kỳ trang nào của bạn ứng dụng mà bạn có, bạn có thể thay thế

Main subtitle

text with images or whatever the hell

you want

but this is a pretty standard login page

it has everything you need to just drag and drop it into any one of the applications that you're building that requires an authentication feature alrighty

so next up we're going to be building out this registration page so let's go

ahead and get started with that guys so this is what our registration page is going to look like guys as you can see

it's extremely similar to the login page it just has a couple extra input fields and this is going to be really really easy for us to create because it reuses so many components from the login view

so

the first thing we're going to want is we're going to want to take this header and copy and paste it over into our registration view

so first of all let's put everything in a v stack and then just copy and paste

that header and you could hit resume and

that would technically be okay

but since we are reusing this component

and the only thing that differs is the text or the title we are going to want

Second subtitle

ứng dụng này

văn bản với hình ảnh hoặc bất cứ điều gì địa ngục

bạn muốn

nhưng đây là một trang đăng nhập khá chuẩn

nó có mọi thứ bạn cần chỉ cần kéo và thả nó vào bất kỳ một trong những các ứng dụng mà bạn đang xây dựng yêu cầu một tính năng xác thực được rồi

vì vậy tiếp theo chúng ta sẽ xây dựng ra khỏi trang đăng ký này vì vậy chúng ta hãy đi

phía trước và bắt đầu với những kẻ đó đây là trang đăng ký của chúng tôi sẽ trông giống như những chàng trai như bạn có thể thấy

nó cực kỳ giống với trang đăng nhập nó chỉ có thêm một vài trường đầu vào và điều này sẽ thực sự thực sự chúng tôi dễ dàng tạo vì nó tái sử dụng rất nhiều thành phần từ chế độ xem đăng nhập

Vì thế

điều đầu tiên chúng ta sẽ muốn là chúng tôi sẽ muốn lấy tiêu đề này và sao chép và dán nó vào cửa chúng tôi xem đăng ký

vì vậy trước hết hãy đặt mọi thứ vào một ngăn xếp v và sau đó chỉ cần sao chép và dán

tiêu đề đó và bạn có thể nhấn tiếp tục

Và

về mặt kỹ thuật sẽ ổn thôi

nhưng vì chúng tôi đang sử dụng lại thành phần này

và điều duy nhất khác biệt là

văn bản hoặc tiêu đề chúng ta sẽ muốn

Main subtitle

to

be good programmers here and create a

reusable component for this so in our

components folder guys let's just go ahead and make a new group for authentication

and let's make a new file here

swift ui view and we're going to call it authentication header

so uh let's actually call it

authentication header view

header view

and we could probably shorten it to auth header view

auth header view

authentication header view is a bit long

underscore previews it's okay so

um

then replace that with auth header view

okay so that looks

better to me

um now let's just go ahead and

copy this guy here

and paste that inside of our auth header

view

right and the only thing we need to do

guys is replace the title okay

so the title does have two parts

so

we're going to uh say let title uh 1

string let title 2

string

you guys might

want to give those more descriptive

Second subtitle

ĐẾN

trở thành lập trình viên giỏi ở đây và tạo ra một

thành phần tái sử dụng cho điều này vì vậy trong chúng tôi

thư mục thành phần đi thôi mọi người phía trước và tạo một nhóm mới cho xác thực

và hãy tạo một tệp mới ở đây

swift ui view và chúng ta sẽ gọi nó

tiêu đề xác thực

vì vậy uh hãy thực sự gọi nó là

xem tiêu đề xác thực

xem tiêu đề

và chúng ta có thể rút ngắn nó thành auth xem tiêu đề

chế độ xem tiêu đề xác thực

chế độ xem tiêu đề xác thực hơi dài

xem trước gạch dưới không sao đâu ô

sau đó thay thế bằng chế độ xem tiêu đề auth

được rồi, có vẻ như vậy

tốt hơn với tôi

ừm bây giờ chúng ta hãy tiếp tục và sao chép anh chàng này ở đây

và dán nó vào bên trong tiêu đề xác thực của chúng tôi

xem

đúng và điều duy nhất chúng ta cần làm các bạn thay tiêu đề đi nhé

vì vậy tiêu đề có hai phần

Vì thế

chúng ta sẽ uh nói hãy để tiêu đề uh 1

chuỗi để tiêu đề 2

sợi dây

các bạn có thể

muốn cung cấp cho những mô tả nhiều hơn

Main subtitle

names

i think that's fine

we can just go ahead here and now say

title 1

and title 2.

and for the preview let's just go ahead and hit fix

and you can say hello and title two is welcome back

and if you guys were to hit resume on that

you should notice that it looks pretty good right hello welcome back and for this guy we're just going to say get started create your account so really quickly let's go back to the login view and let's just replace this with auth header view

and title 1 will be hello

and this will be welcome back

uh and then

go ahead and

copy that

and instead of having this

uh

same exact code repeated twice

we just put it into its own sub view or

subclass and now we can reuse it a lot

more efficiently and cleanly in different places in our code

so this will just be get started

and then this will say

title 2 will be create

your account

Second subtitle

tên

tôi nghĩ điều đó ổn

chúng ta chỉ có thể tiếp tục ở đây và bây giờ nói

tiêu đề 1

và tiêu đề 2.

và để xem trước, hãy tiếp tục

và nhấn sửa

và bạn có thể nói xin chào và tiêu đề hai là chào mừng trở lại

và nếu các bạn nhấn tiếp tục cái đó

bạn nên chú ý rằng nó trông đẹp

tốt đúng xin chào chào mừng trở lại và cho anh chàng này chúng ta sẽ nói có được bắt đầu tạo tài khoản của bạn rất thực sự nhanh chóng quay lại giao diện đăng nhập và hãy thay thế cái này bằng auth

xem tiêu đề

và tiêu đề 1 sẽ là xin chào

và điều này sẽ được chào đón trở lại

uh và sau đó

đi trước và

sao chép cái đó

và thay vì có cái này

uh

cùng một mã chính xác được lặp lại hai lần chúng tôi chỉ đặt nó vào chế độ xem phụ của chính nó hoặc

lớp con và bây giờ chúng ta có thể sử dụng lại nó rất nhiều

hiệu quả và sạch sẽ hơn trong

những nơi khác nhau trong mã của chúng tôi

vì vậy điều này sẽ chỉ được bắt đầu

và sau đó điều này sẽ nói

tiêu đề 2 sẽ được tạo

tài khoản của bạn

Main subtitle

so let's go ahead and hit resume
 awesome right and then guys within this
 we're just gonna add a spacer
 to move that to the top
 and then on this guy we are going to say
 dot ignores safe area
 and that is now our custom header
 extracted out to a sub view so that it's
 reusable now we are going to create
 another v stack here
 and this is going to be for all these
 input fields
 so before we create the input fields
 let's go ahead and create our state
 properties
 so we're going to have email
 blank string at state private var user
 name
 blank string at state private var
 full name
 blank string at state private var
 password
 blank string
 now we'll be able to go ahead and create
 all of these custom input fields so
 let's go back to our login view and we
 could just grab these two because we're
 going to be reusing them
 and paste them in there again
 and so this will be
 let's go ahead and resume
 just to see how those look
 so they're really kind of like cluttered

Second subtitle

vì vậy hãy tiếp tục và nhấn tiếp tục
 tuyệt vời đúng rồi các bạn trong này
 chúng ta sẽ thêm một miếng đệm
 để di chuyển nó lên trên cùng
 và sau đó về anh chàng này, chúng ta sẽ
 nói
 dấu chấm bỏ qua khu vực an toàn
 và đó bây giờ là tiêu đề tùy chỉnh của chúng
 tôi
 trích xuất ra một chế độ xem phụ để nó
 có thể tái sử dụng bây giờ chúng ta sẽ tạo
 ngăn xếp v khác ở đây
 và điều này sẽ dành cho tất cả những điều
 này
 trường đầu vào
 vì vậy trước khi chúng tôi tạo các trường
 đầu vào
 hãy tiếp tục và tạo trạng thái của chúng ta
 của cải
 vì vậy chúng ta sẽ có email
 chuỗi trống ở trạng thái private var user
 tên
 chuỗi trống ở trạng thái private var
 Họ và tên
 chuỗi trống ở trạng thái private var
 mật khẩu
 chuỗi trống
 bây giờ chúng ta sẽ có thể tiếp tục và tạo
 tất cả các trường đầu vào tùy chỉnh này
 hãy quay lại chế độ xem đăng nhập của
 chúng tôi và chúng tôi
 chỉ có thể lấy hai cái này vì chúng tôi
 sẽ tái sử dụng chúng
 và dán chúng vào đó một lần nữa
 và vì vậy đây sẽ là
 hãy tiếp tục và tiếp tục
 chỉ để xem chúng trống như thế nào
 vì vậy chúng thực sự giống như lỗ xộn

Main subtitle

so let's actually just go ahead and fix
um
our padding really quickly we're going
to say dot padding
dot vertical
32
dot padding
uh actually
just go ahead and give it a padding of
32 guys my bad
um and then on your spacing up at the
top here
on the v stack i believe it's 40 pixels
right so that looks a lot more spaced
out and a lot nicer
um
also
we now just need to drop this button in
here but before we do that let's go
ahead and create
make sure that all these input fields
have the correct data right so we have
email password email password we don't
want that
um so this is email this next guy should
be person for the image
the placeholder text should be user
username
and the text should be dollar sign
username
and then
the image name for the next one will

Second subtitle

vì vậy hãy thực sự tiếp tục và sửa chữa
ô
phần đậm của chúng tôi thực sự nhanh
chóng, chúng tôi sẽ đi
để nói dấu chấm đậm
chấm dọc
32
đèm chấm
thực ra
chỉ cần tiếp tục và cung cấp cho nó một
phần đậm của
32 chàng trai tồi tệ của tôi
ừm và sau đó vào khoảng cách của bạn tại
đầu ở đây
trên ngăn xếp v tôi tin rằng nó là 40 pixel
đúng để trông có vẻ cách đều hơn rất nhiều
ra và đẹp hơn rất nhiều
ô
Mà còn
bây giờ chúng ta chỉ cần thả nút này vào
ở đây nhưng trước khi chúng ta làm điều đó
chúng ta hãy đi
phía trước và tạo ra
đảm bảo rằng tất cả các trường đầu vào
này
có dữ liệu chính xác, vì vậy chúng tôi có
email mật khẩu email mật khẩu chúng tôi
không
muốn điều đó
ừm vậy đây là email mà anh chàng tiếp theo
nên
là người cho hình ảnh
văn bản giữ chỗ phải là người dùng
tên tài khoản
và văn bản phải là ký hiệu đô la
tên tài khoản
và sau đó
tên hình ảnh cho cái tiếp theo sẽ

Main subtitle

also be person

placeholder text will be full name
and then this will be dollar sign full
name and then you guys could leave this
password guy here
right so that looks a lot better
once again we can pretty much just grab
this button from our login view

but instead of copy and pasting once
again guys we are going to create a
reusable component here

um that's actually i take that back that
gets a little tricky because the action
that's linked to that will be different
and i could show you guys how to do that

but it is a little bit trickier and
since this is just a button we can
actually go ahead and just drop this in
to our code

uh okay so that looks really good the
text here is going to say sign up
and this will say sign up here
awesome

and then we're going to do the same
thing guys for this uh
this guy right here

um actually just go ahead and grab the
hdac this is not going to be a
navigation link it's just going to be a
button

so let's hit resume and see where we're

at so that looks really good
now guys below the spacer right here
let's go ahead and create a button

Second subtitle

cũng là người

văn bản giữ chỗ sẽ có tên đầy đủ
và sau đó đây sẽ là ký hiệu đô la đầy đủ
tên và sau đó các bạn có thể để lại này
anh chàng mật khẩu ở đây
đúng như vậy trông đẹp hơn rất nhiều
một lần nữa chúng ta có thể lấy khá nhiều
nút này từ chế độ xem đăng nhập của
chúng tôi

nhưng thay vì sao chép và dán một lần
một lần nữa, chúng ta sẽ tạo ra một
thành phần tái sử dụng ở đây
ừm đó thực sự là tôi rút lại điều đó
có một chút khó khăn vì hành động
cái đó được liên kết với cái đó sẽ khác
và tôi có thể chỉ cho các bạn cách làm điều
đó

nhưng nó phức tạp hơn một chút và
vì đây chỉ là một nút nên chúng ta có thể
thực sự tiếp tục và chỉ cần thả cái này vào
mã của chúng tôi

uh được rồi vì vậy trông thực sự tốt
văn bản ở đây sẽ nói đăng ký
và điều này sẽ nói đăng ký tại đây
tuyệt vời

và sau đó chúng ta sẽ làm như vậy
điều này mọi người cho uh
anh chàng này ngay tại đây
um thực sự chỉ cần đi trước và lấy
hdac đây sẽ không phải là một
liên kết điều hướng nó sẽ chỉ là một
cái nút

vì vậy hãy nhấn tiếp tục và xem chúng ta
đang ở đâu
trông thật tuyệt
bây giờ các bạn bên dưới miếng đệm ngay
tại đây
hãy tiếp tục và tạo một nút

Main subtitle

action label

paste in that guy

replace this with already have an account and then we're going to replace sign up with sign in

and go back to your login view and let's

just grab

these two modifiers the padding

and the color

so we can actually remove that color

because

that's the default color of a button

already

and here we just want to actually dismiss

the uh the current view that we're looking at so we do that through the use

of that environment variable so we go ahead and say environment up at the top we're going to create a property that's going to look at the presentation mode

var presentation mode

and then let's just copy that

presentation mode guy and in the action

go ahead here and say presentation mode dot wrapped value dot dismiss and guess what boom shakalaka that wraps

it up for the registration page so you

guys can see that it was a little bit

more work to build the login page but

that saved us a ton of time in building

Second subtitle

nhãn hành động

dán vào anh chàng đó

thay thế cái này bằng đã có một tài khoản và sau đó chúng tôi sẽ thay thế đăng ký với đăng nhập

và quay lại chế độ xem đăng nhập của bạn và hãy

chỉ cần lấy

hai công cụ sửa đổi này phần đệm

và màu sắc

vì vậy chúng tôi thực sự có thể loại bỏ màu đó

bởi vì

đó là màu mặc định của một nút

đã

và ở đây chúng tôi chỉ muốn thực sự miễn nhiệm

uh quan điểm hiện tại mà chúng tôi đang

nhìn vào vì vậy chúng tôi làm điều đó thông qua việc sử dụng

của biến môi trường đó vì vậy chúng tôi đi phía trước và nói môi trường ở trên cùng chúng ta sẽ tạo một tài sản

đó là sẽ nhìn vào bài thuyết trình cách thức

chế độ trình bày var

và sau đó chúng ta hãy sao chép nó

anh chàng chế độ trình bày và trong hành động

hãy tiếp tục ở đây và nói chế độ trình bày giá trị bọc dấu chấm loại bỏ dấu chấm

và đoán xem kết thúc của boom shakalaka là gì

nó lên cho trang đăng ký để bạn

mọi người có thể thấy rằng đó là một chút nhiều công việc hơn để xây dựng trang đăng nhập nhưng

đã tiết kiệm cho chúng tôi rất nhiều thời gian trong việc xây dựng

Main subtitle

the registration page because of how reusable we have made our code so that is one of the true like hallmarks of a good coder guys you create reusable components that will make your code easier to work with down the line right so it's a little bit more work up front but that work pays off for you a lot in the end so um guys give yourselves a huge pad on the back we are done building the user interface for this application believe it took us about two and a half hours and honestly that's not bad this is a pretty complex ui right um especially things in like the profile view and we have scrollable views and floating buttons and we got a bunch of awesome in here and this is starting to look really really good now we're actually ready to hook up our back end so we're going to be creating a firebase project for this so that we can actually authenticate our users there and start uploading data to the database

Second subtitle

trang đăng ký vì làm thế nào có thể tái sử dụng, chúng tôi đã tạo mã của mình để là một trong những dấu hiệu giống như thật của một những người lập trình tốt, bạn tạo ra có thể tái sử dụng thành phần sẽ làm cho mã của bạn dễ dàng hơn để làm việc với xuống dòng bên phải vì vậy cần thêm một chút công việc phía trước nhưng công việc đó trả hết cho bạn rất nhiều trong kết thúc như vậy ừm các bạn cho mình một tấm đệm lớn ở phía sau chúng tôi đã hoàn thành việc xây dựng giao diện người dùng cho ứng dụng này tin rằng nó đã đưa chúng tôi khoảng hai tiếng rưỡi và thành thật mà nói điều đó không tệ, đây là một vấn đề khá phức tạp ừm, đặc biệt là những thứ như hồ sơ xem và chúng tôi có chế độ xem có thể cuộn và các nút nỗi và chúng tôi có một loạt các nút shit tuyệt vời ở đây và đây là bắt đầu trông thực sự thực sự tốt bây giờ chúng tôi thực sự đã sẵn sàng để kết nối trở lại của chúng tôi kết thúc vì vậy chúng tôi sẽ tạo ra một dự án firebase cho điều này để chúng tôi có thể thực sự xác thực người dùng của chúng tôi ở đó và bắt đầu tải dữ liệu lên cơ sở dữ liệu

Main subtitle

that we'll be able to start viewing in the app like tweets profile images user accounts all that awesome stuff so get really excited for that guys everything's about to come together and we're about to have a fully working social media application that's pretty much exactly like twitter so this is going to be awesome get excited we're almost done okay guys so let's go ahead and set up our back end with firebase now so you guys are going to want to go to an internet browser and you're going to want to head to console.firebaseio.google.com okay and all you need is a google account to sign in and be able to create firebase projects here so this is where our backend is going to be set up so we're just going to go here and click on add projects and i'm going to call this twitter youtube tutorial obviously you guys just name this whatever you want let's select continue and we don't need analytics for this project let's just hit create project and wait for it to do its thing and actually set up our back end and once more guys this is going to be where we

Second subtitle

mà chúng tôi sẽ có thể bắt đầu xem trong ứng dụng thích tweet hình ảnh hồ sơ người dùng tài khoản tất cả những thứ tuyệt vời đó để có được thực sự phần khích cho những người đó mọi thứ sắp đến với nhau và chúng tôi sắp có một công việc đầy đủ ứng dụng truyền thông xã hội khá đẹp giống hệt như twitter vì vậy đây là sẽ trở nên tuyệt vời, hãy phần khích, chúng tôi sắp xong được rồi các bạn, hãy tiếp tục và thiết lập phần cuối của chúng tôi với firebase bây giờ để bạn mọi người sẽ muốn đi đến một trình duyệt internet và bạn sẽ muốn hướng đến console.firebaseio.google.com được rồi và tất cả những gì bạn cần là google tài khoản để đăng nhập và có thể tạo firebase dự án ở đây vì vậy đây là nơi của chúng tôi phụ trợ sẽ được thiết lập vì vậy chúng tôi chỉ cần vào đây và nhấp vào thêm dự án và tôi sẽ gọi đây là Twitter hướng dẫn youtube rõ ràng các bạn chỉ cần đặt tên này bất cứ điều gì bạn muốn, hãy chọn tiếp tục và chúng tôi không cần phân tích cho việc này dự án chúng ta hãy nhấn tạo dự án và đợi nó làm việc của nó và thực sự thiết lập back-end của chúng tôi và một lần nhiều người hơn, đây sẽ là nơi chúng ta

Main subtitle

authenticate all our users and store all of our data including profile images and app data like user information tweets and all that good stuff so it's sort of a one stop shop for uh your backend services so now that we're here let's go ahead and go to this authentication tab okay and you're just gonna go to the top and you're gonna click get started and we're just gonna set up an email and password uh authentication method um there's a ton of other authentication methods that we can use with firebase but for this app we're just going to do email password so click that hit enable and then click save so that is now set up now let's go to our uh firestore database and you're go actually sorry guys go back to project overview that's my bad um and you're gonna say here get started by adding firebase to your app you're gonna click on ios and we need the bundle id for our application here so let's go back to the xcode project and click on your project name all the way up at the top

Second subtitle

xác thực tất cả người dùng của chúng tôi và lưu trữ tất cả dữ liệu của chúng tôi bao gồm hình ảnh hồ sơ và dữ liệu ứng dụng như tweet thông tin người dùng và tất cả những thứ tốt đẹp như vậy một cửa hàng duy nhất cho phần phụ trợ của bạn dịch vụ vì vậy bây giờ chúng tôi ở đây, hãy đi trước và chuyển đến tab xác thực này được rồi và bạn sẽ chỉ đi đến đỉnh và bạn sẽ nhấp vào bắt đầu và chúng tôi sẽ thiết lập một email và mật khẩu uh phương thức xác thực ừm có rất nhiều xác thực khác các phương thức mà chúng ta có thể sử dụng với firebase nhưng đối với ứng dụng này, chúng tôi sẽ làm mật khẩu email để nhấp vào đó nhấn bật và sau đó nhấp vào lưu vì vậy bây giờ đã được thiết lập bây giờ chúng ta hãy đi đến uh cơ sở dữ liệu firestore của chúng tôi và bạn đang đi thực sự xin lỗi mọi người đi quay lại tổng quan dự án đó là điểm xấu của tôi ô và bạn sẽ nói ở đây hãy bắt đầu bằng thêm firebase vào ứng dụng của bạn, bạn sẽ nhấp vào ios và chúng tôi cần id gói cho ứng dụng của chúng tôi ở đây vì vậy hãy đi quay lại dự án xcode và nhấp vào tên dự án của bạn tất cả các cách lên tại đứng đầu

Main subtitle

and then you're just going to see in this general tab right here with your target selected so

if you guys aren't seeing this go right here where you see your target

name and make sure that you have target selected if you don't this is going to look different so make sure you just

select that target and we're going to copy and paste this bundle identifier that you see here go back to firebase and paste it in there and then we're just going to say register app and now we have to download this google

service info plist file so let's just go

ahead and click download there this is going to show up in your finder

under or sorry in your downloads folder

what i like to do guys is uh if you've done this before you're gonna see a dash and a number there you cannot

have that dash number so here's what i

like to do um i actually like to drag and drop this into my project folder so mine is located in my desktop folder

under this programming folder that i have and you can see my

project name right here so you just drag and drop it into there

Second subtitle

và sau đó bạn sẽ thấy trong tab chung này ngay tại đây với mục tiêu của bạn được chọn để

nếu các bạn không nhìn thấy điều này đi ngay đây nơi bạn nhìn thấy mục tiêu của bạn đặt tên và đảm bảo rằng bạn có mục tiêu đã chọn nếu bạn không điền này sẽ trông khác nhau vì vậy hãy chắc chắn rằng bạn chỉ

chọn mục tiêu đó và chúng tôi sẽ sao chép và dán số nhận dạng gói này mà bạn thấy ở đây quay lại firebase và dán nó vào đó và sau đó chúng tôi chỉ cần nói đăng ký ứng dụng và bây giờ chúng ta phải tải xuống google này

tập tin plist thông tin dịch vụ vì vậy chúng ta hãy bắt đầu

phía trước và nhập vào tải xuống ở đó cái này sẽ hiển thị trong công cụ tìm của bạn

bên dưới hoặc xin lỗi trong thư mục tải xuống của bạn

những gì tôi thích làm mọi người là uh nếu bạn đã làm điều này trước khi bạn sẽ thấy một dấu gạch ngang và một số ở đó bạn không thể

có số gạch ngang đó vì vậy đây là những gì tôi

thích làm ừm tôi thực sự thích kéo và thả cái này vào thư mục dự án của tôi để của tôi nằm trong thư mục máy tính để bàn của tôi

dưới chương trình này thư mục mà tôi có và bạn có thể thấy thư mục của tôi tên dự án ngay tại đây vì vậy bạn chỉ cần kéo và thả nó vào đó

Main subtitle

and what is going on there
 huh that's weird i should have
 oh there it is okay
 so my bad that was in the wrong
 uh
 let's move it
 so i'm gonna go back to desktop
 and i believe it's twitter swift ui
 youtube that i want to put it in yeah
 okay
 so i'm gonna put it in there
 and then i'm gonna rename it
 to delete that dash and the number okay
 and then let's just hit next
 um and this is how we add our firebase
 to our xcode project we're going to do
 that in just a second really quickly
 guys let's drag and drop this into our
 project
 so i'm just going to drag this guy
 over right below my assets folder and
 hit finish
 so now we can see this google service
 info file
 and that's looking good now we just have
 to actually add the firebase sdk to our
 project so
 we can do this now through the swift
 package manager previously it was all
 done through cocoapods but we have the
 ability to add third-party code into our

Second subtitle

và những gì đang xảy ra ở đó
 huh đó là kỳ lạ tôi nên có
 ồ không sao đâu
 vì vậy xấu của tôi đó là trong sai
 uh
 Chúng ta hãy di chuyển nó
 vì vậy tôi sẽ quay trở lại máy tính để bàn
 và tôi tin đó là twitter swift ui
 youtube mà tôi muốn đưa nó vào yeah
 được rồi
 vì vậy tôi sẽ đặt nó ở đó
 và sau đó tôi sẽ đổi tên nó
 để xóa dấu gạch ngang và số đó đi nhé
 và sau đó chúng ta hãy nhấn tiếp theo
 ừm và đây là cách chúng tôi thêm căn cứ
 hỏa lực của mình
 với dự án xcode của chúng tôi, chúng tôi sẽ
 làm
 điều đó chỉ trong một giây thực sự nhanh
 chóng
 các bạn hãy kéo và thả cái này vào
 dự án
 vì vậy tôi sẽ kéo anh chàng này
 ở ngay bên dưới thư mục tài sản của tôi và
 nhấn kết thúc
 vì vậy bây giờ chúng ta có thể thấy dịch vụ
 google này
 tập tin thông tin
 và điều đó có vẻ tốt bây giờ chúng ta chỉ có
 để thực sự thêm sdk firebase vào của
 chúng tôi
 dự án như vậy
 chúng ta có thể làm điều này ngay bây giờ
 thông qua nhanh chóng
 trình quản lý gói trước đây là tất cả
 được thực hiện thông qua cacaopods
 nhưng chúng tôi có
 khả năng thêm mã của bên thứ ba vào của
 chúng tôi

Main subtitle

app through the swift package manager now so really quickly guys let's just go ahead and copy this link we're going to

go back to xcode you're going to go to file and you're going to say add packages or click on add packages in this enter package url paste the url that we got click add package and let's just wait for it to do its thing it's basically fetching all of the code it needs to from firebase um from github so that we can integrate it into our project and this is going to allow us to be able to use code that firebase has created to interact with our api it's going to help us authenticate users upload data

download data upload images download images and all that good stuff

okay guys so now that all that has been fetched let's check what we want to add into our project so definitely going to

want to add firebase auth uh we are going to add uh let's expand this a little bit um we are going to add firebase database

swift
or actually
no sorry not firebase database switch we're going to add fire by firebase firestore swift that's my bad

Second subtitle

app thông qua trình quản lý gói nhanh bây giờ thật nhanh các bạn hãy đi thôi phía trước và sao chép liên kết này, chúng tôi sẽ

quay lại xcode bạn sẽ truy cập tập tin và bạn sẽ nói thêm gói hoặc nhập vào thêm gói trong này nhập url gói dán url mà chúng tôi đã nhập vào thêm gói và chúng ta hãy đợi nó làm điều về cơ bản là tìm nạp tất cả mã nó cần từ firebase um từ github để chúng tôi có thể tích hợp nó vào dự án của chúng tôi và điều này sẽ cho phép chúng tôi có thể sử dụng mã mà firebase đã tạo để tương tác với api của chúng tôi, nó sẽ giúp ích chúng tôi xác thực người dùng tải lên dữ liệu tải xuống dữ liệu tải lên hình ảnh tải xuống hình ảnh và tất cả những thứ tốt được rồi các bạn vì vậy bây giờ tất cả những gì đã được đã tìm nạp, hãy kiểm tra những gì chúng tôi muốn thêm

vào dự án của chúng tôi vì vậy chắc chắn sẽ

muốn thêm xác thực firebase uh chúng ta sẽ thêm uh hãy mở rộng cái này một chút

ừm, chúng ta sẽ thêm cơ sở dữ liệu firebase

nhanh

hoặc thực sự

không xin lỗi không chuyển đổi cơ sở dữ liệu firebase

chúng ta sẽ thêm lửa bằng firebase firestore swift đó là lỗi của tôi

Main subtitle

and then we are going to add firebase storage
 um so make sure you have firebase storage
 firebase firestore swift
 firebase firestore
 and firebase so you should have four checked one more time firebase off
 firebase firestore firebase firestore swift and firebase storage okay
 so
 let's say add package
 and that is going to add that package dependency to our project
 and
 uh to close out this chapter guys i just want to make sure that we have successfully integrated firebase so let's go to our twitter swift ui app
 file right here just your main app file
 and we are going to import firebase up at the top
 and within this guy we're going to add an init method and we're going to say firebase app dot configure okay
 so now let's go ahead and run this project
 and we should notice that everything works just fine
 and we should not get any crashes
 so if that's the case that means we

Second subtitle

và sau đó chúng ta sẽ thêm firebase kho
 ừm, vậy hãy chắc chắn rằng bạn có firebase kho
 firebase firestore swift
 căn cứ hỏa lực
 và firebase, vì vậy bạn nên có bốn đã kiểm tra thêm một lần nữa khi tắt firebase
 firebase firestore firebase firestore lưu trữ nhanh và firebase ổn
 Vì thế
 giả sử thêm gói
 và điều đó sẽ thêm gói đó
 phụ thuộc vào dự án của chúng tôi
 Và
 uh để kết thúc chương này guys tôi chỉ muốn đảm bảo rằng chúng tôi có tích hợp thành công firebase như vậy
 hãy truy cập twitter của chúng tôi nhanh chóng ui
 ứng dụng
 tệp ngay tại đây chỉ là tệp ứng dụng chính của bạn
 và chúng tôi sẽ nhập firebase lên ở trên cùng
 và trong anh chàng này chúng ta sẽ thêm một phương thức init và chúng ta sẽ nói ứng dụng firebase chấm cấu hình ổn
 vì vậy bây giờ chúng ta hãy tiếp tục và chạy cái này
 dự án
 và chúng ta nên chú ý rằng mọi thứ hoạt động tốt
 và chúng ta sẽ không gặp phải bất kỳ sự cố nào
 vì vậy nếu đó là trường hợp có nghĩa là

Main subtitle

successfully integrated firebase into our app and set up our back end correctly let's go ahead and actually finish the setup process we hit next so this is uh the step we just did but we obviously don't have an app delegate file so that's just how you sort of work around that when you have this uh just app file you don't have an app delegate you basically just have to initialize firebase when the app launches and now we can just continue to console and let's go ahead and continue with our setup here so we're going to go to firestore database we're going to say create database we can just start in test mode and let's hit enable so cloud firestore is where we're going to upload actual data like information about our users in tweet data and all that good stuff likes comments whatever it may be it's all going to get stored in firestore and that's where we're going to be fetching that data from as well

Second subtitle

chúng tôi tích hợp thành công firebase vào ứng dụng của chúng tôi và thiết lập mặt sau của chúng tôi một cách chính xác, hãy tiếp tục và thực sự hoàn thành thiết lập quá trình chúng tôi nhấn tiếp theo Vì thế đây là uh bước chúng tôi vừa làm nhưng rõ ràng là chúng tôi không có đại biểu ứng dụng tập tin để đó chỉ là cách bạn sắp xếp công việc xung quanh đó khi bạn có cái này uh chỉ tệp ứng dụng mà bạn không có đại biểu ứng dụng về cơ bản bạn chỉ cần khởi tạo căn cứ hỏa lực khi ứng dụng khởi chạy và bây giờ chúng ta có thể chỉ cần tiếp tục điều khiển và hãy tiếp tục và tiếp tục với thiết lập ở đây vì vậy chúng tôi sẽ đi đến cơ sở dữ liệu lò sưởi chúng ta sẽ nói tạo cơ sở dữ liệu chúng ta chỉ có thể bắt đầu trong chế độ thử nghiệm và hãy nhấn cho phép vì vậy cloud firestore là nơi chúng ta sẽ đến để tải lên dữ liệu thực tế như thông tin về người dùng của chúng tôi trong tweet dữ liệu và tất cả những thứ hay ho đó bình luận bắt kể nó có thể là gì tất cả sẽ được lưu trữ trong lò sưởi và đó là nơi chúng ta sẽ đến cũng lấy dữ liệu đó từ

Main subtitle

so this is what you should see on your screen now and the last thing we have to do while our app is still running is go to storage and this is where we're going to store our images so you can see that it's already set up so it looks good to go and the reason we have to store images separately guys is because uh typical back-end standard practice is to store large files in a separate storage bucket you don't want to store your files where all of your sort of metadata is stored you store larger files like images and videos in a separate storage bucket and you get some sort of pointer to the file that you're going to store in the actual database something that's known as like an image url or video video url or media url stuff like that so we'll actually see how that works in just a little bit but just know that that's standard practice you never want to store your metadata and your large files in one place you want to keep them separate so that should just about do it i just want

Second subtitle

vì vậy đây là những gì bạn sẽ thấy trên của bạn màn hình ngay bây giờ và điều cuối cùng chúng ta phải làm trong khi ứng dụng của chúng tôi vẫn đang chạy là đi đến kho lưu trữ và đây là nơi chúng ta sẽ đến để lưu trữ của chúng tôi hình ảnh như vậy bạn có thể thấy rằng nó đã được thiết lập như vậy nó có vẻ tốt để đi và lý do chúng ta phải lưu trữ hình ảnh các bạn riêng biệt là bởi vì uh điển hình thực hành tiêu chuẩn back-end là lưu trữ các tệp lớn trong một thùng lưu trữ riêng bạn không muốn lưu trữ các tệp của mình ở đâu tất cả các loại siêu dữ liệu của bạn được lưu trữ bạn lưu trữ các tệp lớn hơn như hình ảnh và video trong một thùng lưu trữ riêng biệt và bạn nhận được một số loại con trỏ tới tệp mà bạn sẽ lưu trữ trong thực tế cơ sở dữ liệu một cái gì đó được gọi là như url hình ảnh hoặc url video video hoặc phương tiện những thứ url như thế vì vậy chúng tôi sẽ thực sự xem nó hoạt động như thế nào chỉ trong một chút nhưng chỉ cần biết rằng đó là tiêu chuẩn thực hành bạn không bao giờ muốn lưu trữ siêu dữ liệu của bạn và các tệp lớn của bạn trong một nơi bạn muốn giữ chúng riêng biệt Vì thế điều đó chỉ nên làm điều đó tôi chỉ muốn

Main subtitle

to make sure that our app runs and it looks like it's almost done so once we can see that our app is running we can close out this chapter and actually start creating some view models in our app that are going to help us do things like authenticate our users and upload user information to the database which is the first thing we're going to do we're going to be working with this sign up page so that we can authenticate some users and store their information in the database and then we'll be able to you know search for users and stuff like that and see their profiles and then ultimately be able to upload tweets and really quickly you guys might notice that there isn't a profile image selector here so we're we are going to integrate that once the user signs up we're going to take them over to a different screen that asks them to uh add a profile image and that's where we're going to upload it and then actually save that in firebase storage so that's going to be really cool and we can even make it so that if the

Second subtitle

để đảm bảo rằng ứng dụng của chúng tôi chạy và nó có vẻ như nó gần như đã hoàn thành khi chúng tôi có thể thấy rằng ứng dụng của chúng tôi đang chạy, chúng tôi có thể kết thúc chương này và thực sự bắt đầu tạo một số mô hình xem trong của chúng tôi ứng dụng sẽ giúp chúng tôi làm mọi việc như xác thực người dùng của chúng tôi và tải lên thông tin người dùng vào cơ sở dữ liệu là điều đầu tiên chúng ta sẽ làm chúng ta sẽ làm việc với dấu hiệu này lên trang để chúng tôi có thể xác thực một số người dùng và lưu trữ thông tin của họ trong cơ sở dữ liệu và sau đó chúng tôi sẽ có thể cho bạn biết tìm kiếm người dùng và những thứ tương tự và xem hồ sơ của họ và sau đó cuối cùng có thể tải lên các tweet và thực sự nhanh chóng các bạn có thể nhận thấy rằng không có một hình ảnh hồ sơ bộ chọn ở đây vì vậy chúng tôi sẽ tích hợp điều đó một khi người dùng đăng ký lên, chúng tôi sẽ đưa họ đến một màn hình khác hỏi họ để uh thêm một hình ảnh hồ sơ và đó là nơi chúng tôi sẽ tải nó lên và sau đó thực sự lưu nó trong firebase lưu trữ vì vậy sẽ thực sự mát mẻ và chúng tôi thậm chí có thể làm cho nó để nếu

Main subtitle

use the user could choose not to upload

an image but for the purposes of this app we are going to sort of force them to upload a profile image so that we don't have to worry about adding a default profile image and having edit profile functionality and stuff like that

okay guys so

we noticed that our app has successfully

built we're not getting any crashes and

everything looks really good so the next phase of this is actually going to be signing up some users and creating some view models for our app and really

getting into the fun stuff so that's what we're going to be doing in the next chapter guys

so we are going to get started in this chapter guys by creating our

authentication view model and this auth view model as we're going to call it or authentication view model is going to be responsible for communicating with our api with anything that involves authenticating a user meaning logging them in

signing them up if they have never created an account before

and doing other things like checking to see if they're logged in when we launch

the app so we know whether or not to

Second subtitle

sử dụng người dùng có thể chọn không tải lên

một hình ảnh nhưng cho mục đích này ứng dụng, chúng tôi sẽ sắp xếp buộc họ để tải lên một hình ảnh hồ sơ để chúng tôi không phải lo lắng về việc thêm một hình ảnh hồ sơ mặc định và có chỉnh sửa chức năng hồ sơ và những thứ như cái đó

được rồi mọi người

chúng tôi nhận thấy rằng ứng dụng của chúng tôi đã thành công

được xây dựng, chúng tôi không gặp phải bất kỳ sự cố nào và

mọi thứ trông thực sự tốt vì vậy tiếp theo giai đoạn này thực sự sẽ là

đăng ký một số người dùng và tạo một số xem các mô hình cho ứng dụng của chúng tôi và thực sự

tham gia vào những thứ thú vị vì vậy đó là chúng ta sẽ làm gì tiếp theo chương kẻ

vì vậy chúng ta sẽ bắt đầu trong việc này chương guys bằng cách tạo ra của chúng tôi

mô hình xem xác thực và xác thực này xem mô hình như chúng ta sẽ gọi nó hoặc mô hình xem xác thực sẽ là

chiến trách nhiệm liên lạc với chúng tôi api với bất cứ thứ gì liên quan

xác thực người dùng có nghĩa là ghi nhật ký họ trong

đăng ký cho họ nếu họ chưa bao giờ đã tạo một tài khoản trước đó

và làm những việc khác như kiểm tra xem họ có đăng nhập khi chúng tôi khởi chạy không

ứng dụng để chúng tôi biết có nên hay không

Main subtitle

show the user with the main interface or the authentication flow so let's go ahead and get started with this so let's collapse that components folder we're going to go into this view model folder and create a new file and it's going to be a swift file and i'm just going to call it auth view model and we are going to import swift ui and we're also going to import firebase and we're going to create a class called off view model and it's going to conform to the observable object

[Music]

right here

type alias

okay and we're just going to create an init

here and we're going to create a couple parameters up at the top so we're going to create a published variable

uh var

and it's going to be called user session and it's going to be a firebase auth dot user

and it's going to be optional so this is going to be where we store the actual user session so if the user

is logged in this property will have a value

if they're not this property will be nil

so here's how we're going to initialize it in our initializer what we're going

Second subtitle

hiển thị cho người dùng giao diện chính hoặc

quy trình xác thực, vì vậy hãy bắt đầu phía trước và bắt đầu với điều này vì vậy hãy thu gọn thư mục thành phần đó chúng ta sẽ đi vào mô hình xem này thư mục và tạo một tệp mới và nó là sẽ là một tập tin nhanh chóng và tôi chỉ sẽ gọi nó là mô hình xem auth và chúng tôi sẽ nhập Swift ui và chúng tôi cũng sẽ nhập firebase và chúng ta sẽ tạo một lớp gọi là tắt mô hình xem và nó sẽ phù hợp đến đối tượng quan sát

[Âm nhạc]

ngay chỗ này

gõ bí danh

được rồi và chúng ta sẽ tạo một trong đó

ở đây và chúng ta sẽ tạo ra một vài thông số lên ở đầu vì vậy chúng tôi sẽ tạo ra một xuất bản Biến đổi

uh var

và nó sẽ được gọi là phiên người dùng và nó sẽ là một dấu chấm xác thực firebase người dùng

và nó sẽ là tùy chọn vì vậy đây sẽ là nơi chúng tôi lưu trữ phiên người dùng thực tế vì vậy nếu người dùng

được đăng nhập vào tài sản này sẽ có một giá trị

nếu họ không thuộc tính này sẽ là con số không

vì vậy đây là cách chúng ta sẽ khởi tạo nó trong trình khởi tạo của chúng tôi những gì chúng tôi sẽ làm

Main subtitle

to do is say self dot user session

equals auth.auth.currentuser

so

basically what this is going to do is if a current user or a user is logged in

we're going to store the user session in this variable and we're going to uh implement some presentation logic in our content view in a second and

basically if this user session has a value meaning there's a user that's logged in we're gonna show the user the

main interface of the application and load their user data if not we're going

to show the user the authentication flow and have them log in or sign up so let's

go over how to do that now

so we're going to go to our content view

and we are actually going to go here and create a group so

i want to grab all this stuff here right

and i'm going to cut it

and then i'm going to create something called a group

and

make sure you guys uh

let's just go ahead and paste that

actually we actually need to introduce a

property here

so i'm going to say at environment object

Second subtitle

phải làm là nói phiên người dùng tự chấm bằng auth.auth.currentuser

Vì thế

về cơ bản điều này sẽ làm là nếu người dùng hiện tại hoặc người dùng đã đăng nhập

chúng tôi sẽ lưu trữ phiên người dùng trong biến này và chúng ta sẽ

uh thực hiện một số logic trình bày trong xem nội dung của chúng tôi trong một giây và

về cơ bản nếu phiên người dùng này có giá trị có nghĩa là có một người dùng đã đăng nhập, chúng tôi sẽ hiển thị cho người dùng

giao diện chính của ứng dụng và tải dữ liệu người dùng của họ nếu không chúng tôi sẽ

để hiển thị cho người dùng luồng xác thực và yêu cầu họ đăng nhập hoặc đăng ký, vì vậy hãy

đi qua làm thế nào để làm điều đó bây giờ vì vậy chúng tôi sẽ đi đến chế độ xem nội dung của chúng tôi

và chúng tôi thực sự sẽ đi đến đây và tạo một nhóm như vậy

tôi muốn lấy tất cả những thứ này ngay tại đây

và tôi sẽ cắt nó

và sau đó tôi sẽ tạo ra một cái gì đó được gọi là một nhóm

Và

đảm bảo nhé các bạn uh

chúng ta hãy tiếp tục và dán nó

thực sự chúng ta thực sự cần phải giới thiệu một

tài sản ở đây

vì vậy tôi sẽ nói tại môi trường sự vật

Main subtitle

view model

and it's going to be of type off view
model and i'm going to go over why
this is an environment object man make

sure you guys add the word var there
okay

so let's actually cut this whole thing
out one more time
and we're going to say if view model dot
user session

equals nil

and else

so if the user session is nil that means

no user logged in
else

have a logged
in user so basically if we have a logged

in user then we're going to show all the

main code right but let's actually
extract this to its own view
um so we can actually go down here
and just say extension
content view

and we can just go ahead and call this
like var

main

interface view or something

some view

paste it all in there

and we can now go ahead and say main

interface view and here we're going to
say login view

so this is our presentation logic guys

Second subtitle

xem mô hình

và nó sẽ bị loại khỏi chế độ xem
mô hình và tôi sẽ đi qua lý do tại sao
đây là một đối tượng môi trường mà con
người tạo ra

chắc chắn các bạn thêm từ var vào đó
được rồi

vì vậy hãy thực sự cắt bỏ toàn bộ điều này
ra ngoài một lần nữa

và chúng ta sẽ nói nếu xem mô hình chấm
phiên người dùng

bằng không

và khác

vì vậy nếu phiên người dùng bằng không,
điều đó có nghĩa là

không có người dùng nào đăng nhập
khác

có một đăng nhập

trong người dùng về cơ bản nếu chúng ta
có một bản ghi

trong người dùng thì chúng tôi sẽ hiển thị tất
cả

mã chính đúng nhưng hãy thực sự
trích xuất cái này để xem riêng
ùm để chúng ta thực sự có thể đi xuống đây
và chỉ cần nói phần mở rộng
xem nội dung

và chúng ta có thể tiếp tục và gọi đây là
như var
chủ yếu

xem giao diện hoặc một cái gì đó
một số xem

dán tất cả vào đó

và bây giờ chúng ta có thể tiếp tục và nói
chính

xem giao diện và ở đây chúng ta sẽ
nói đăng nhập xem
đây là logic trình bày của chúng ta

Main subtitle

so you guys are going to notice that if we were to run the application right now that it's going to still show us our login view actually guys go ahead and

stop that that's my bad let's go ahead and replace this login view so delete that so it's

no longer the root view of the application and let's go ahead and make the root view this content view

again and then run the application and here's where you guys are going to be able to see this magic working you're going to see that even though the root view of the app is the content view

that we should be presented with the login view here and i actually think we're going to get

a crash here and i'll explain why in a second this ties into the environment object that we created so if you guys look at the error that we get

we say fatal error no observable object

of type auth.viewmodel found a.viewmodel or sorry view.environment

object for autoview model baby may be missing as an ancestor of this view so here guys we are using this environment object uh variable

Second subtitle

vì vậy các bạn sẽ chú ý rằng nếu chúng tôi đã chạy ứng dụng ngay bây giờ rằng nó sẽ vẫn cho chúng ta thấy đăng nhập xem thực tế các bạn hãy tiếp tục và

dừng lại đó là lỗi của tôi, hãy tiếp tục và thay thế

chế độ xem đăng nhập này vì vậy hãy xóa nó đi

không còn chế độ xem gốc của ứng dụng và hãy tiếp tục và làm cho chế độ xem gốc chế độ xem nội dung này

một lần nữa và sau đó chạy ứng dụng và đây là nơi các bạn sẽ đến có thể thấy phép thuật này hoạt động bạn sẽ thấy điều đó mặc dù

chế độ xem gốc của ứng dụng là chế độ xem nội dung

rằng chúng ta nên được trình bày với đăng nhập xem tại đây

và tôi thực sự nghĩ rằng chúng ta sẽ nhận

được một vụ tai nạn ở đây và tôi sẽ giải thích tại sao trong một

thứ hai điều này liên quan đến môi trường đối tượng mà chúng tôi đã tạo

vì vậy nếu các bạn nhìn vào lỗi mà chúng tôi lấy

chúng tôi nói lỗi nghiêm trọng không có đối tượng có thể quan sát được

loại auth.viewmodel được tìm thấy

một.viewmodel hoặc xin lỗi view.environment

đối tượng cho em bé mô hình autoview có thể là

mất tích như một tổ tiên của quan điểm này vì vậy ở đây các bạn chúng tôi đang sử dụng cái này

đối tượng môi trường uh biến

Main subtitle

and

you notice that we don't initialize this

off view model here like we don't say

like

equals

bam right

we're not doing that and that's because
we need this

variable to be used across multiple
locations in the app right so we're
going to need this in the content view
we're going to need it in the login view

we're going to need it in the
registration view so instead of
initializing that view model over and
over again or passing it around to all
the different views

we're going to create this environment
object so that it's usable within that

whole environment of views so that's why

we create this as an environment object

but we do need to initialize it
somewhere so we're actually going to do
that in the app file

so

we are going to go here
and we are going to say
at state objects

var

view model equals auth view model
and then on the content view we're going

to say dot environment object

Second subtitle

Và

bạn lưu ý rằng chúng tôi không khởi tạo cái
này

tắt xem mô hình ở đây như chúng tôi không
nói

giống

bằng

bam phải

chúng tôi không làm điều đó và đó là bởi vì
chúng tôi cần cái này

biến được sử dụng trên nhiều

vị trí trong ứng dụng ngay vì vậy chúng tôi
sẽ cần điều này trong chế độ xem nội dung
chúng tôi sẽ cần nó trong chế độ xem đăng
nhập

chúng ta sẽ cần nó trong

đăng ký xem như vậy thay vì

khởi tạo lại mô hình xem đó và

một lần nữa hoặc chuyển nó cho tất cả
các quan điểm khác nhau

chúng ta sẽ tạo ra môi trường này

đối tượng để nó có thể sử dụng được trong
đó

toàn bộ môi trường của quan điểm vì vậy đó
là lý do tại sao

chúng tôi tạo cái này như một đối tượng môi
trường

nhưng chúng ta cần phải khởi tạo nó

ở đâu đó vì vậy chúng tôi thực sự sẽ làm
cái đó trong tệp ứng dụng

Vì thế

chúng ta sẽ đi đây

và chúng ta sẽ nói

tại các đối tượng trạng thái

var

xem mô hình bằng auth xem mô hình

và sau đó trên chế độ xem nội dung, chúng
tôi sẽ

để nói đối tượng môi trường dấu chấm

Main subtitle

view model

and let's go ahead and go back to this

off view model and actually just go ahead and say print debug user session

is

oops

self.user session so we're going to get

a print statement right and that's going to happen when this guy initializes when this off view model initializes and it's initializing inside of our app file and

then it's passing it as an environment object to our content view

then

back in our content view now we'll

actually be able to use this because it will have been initialized somewhere so now we should see

that we are getting shown our login view

so that is really really cool right and

if you guys were to notice if i just replace the or like change the logic here

and this is incorrect logic i just want

to show you guys what would happen if i

changed it

so this is saying hey if the user session has a value then show the login

view well we obviously don't want to do

Second subtitle

xem mô hình

và chúng ta hãy tiếp tục và quay lại vấn đề này

tắt xem mô hình và thực sự chỉ cần đi trước và nói in gỡ lỗi

phiên người dùng

là

Ói

phiên self.user vì vậy chúng tôi sẽ nhận được

một tuyên bố in đúng và điều đó sẽ xảy ra xảy ra khi anh chàng này khởi tạo khi mô hình tắt chế độ xem này khởi tạo và nó khởi tạo bên trong tệp ứng dụng của chúng tôi và

sau đó nó chuyển nó thành một môi trường phản đối chế độ xem nội dung của chúng tôi sau đó

quay lại chế độ xem nội dung của chúng tôi bây giờ chúng tôi sẽ

thực sự có thể sử dụng cái này bởi vì nó sẽ được khởi tạo ở đâu đó

vì vậy bây giờ chúng ta nên xem

rằng chúng tôi đang được hiển thị chế độ xem đăng nhập của mình

vì vậy điều đó thực sự rất tuyệt đúng không và

nếu các bạn chú ý nếu tôi chỉ thay thế hoặc thích thay đổi logic đây

và đây là logic không chính xác, tôi chỉ muốn

để cho các bạn thấy điều gì sẽ xảy ra nếu tôi

đã thay đổi nó

vì vậy đây là nói xin chào nếu người dùng phiên có một giá trị sau đó hiển thị thông tin đăng nhập

xem tốt rõ ràng là chúng tôi không muốn làm

Main subtitle

that

it's just to show you that this presentation logic is actually working right so you notice i changed the logic and it went through this uh sort of view logic here and it said hey.viewmodel.user session is nil so i'm going to show you the main

interface view it's going to hit this

else block here

and now everything is working

as we have seen

um

that we built before which is looking

really good

so anyway let's change the logic back

and now let's go over what we need to do

to actually

um make our app work so that it can

register users log them in and upload some user data so we're going to hop

back into our auth view model

oh and let's see if we got that print

statement

so i'm going to filter this out with debug yep we see that user session is

nil meaning we do not have a user signed

in because we haven't added any users to the application

Second subtitle

cái đó

nó chỉ để cho bạn thấy rằng điều này logic trình bày đang thực sự hoạt động đúng để bạn nhận thấy tôi đã thay đổi logic và nó đã trải qua kiểu xem uh này

logic ở đây và nó nói này
phiên.viewmodel.user

là con số không vì vậy tôi sẽ chỉ cho bạn cái chính

xem giao diện nó sẽ đánh cái này

khối khác ở đây

và bây giờ mọi thứ đang hoạt động
như chúng ta đã thấy

ô

mà chúng tôi đã xây dựng trước đó đang
tìm kiếm

thật sự tốt

Vì vậy, dù sao đi nữa, hãy thay đổi logic trở lại

và bây giờ chúng ta hãy đi qua những gì
chúng ta cần làm

thực sự

um làm cho ứng dụng của chúng tôi hoạt
động để nó có thể

đăng ký người dùng đăng nhập và tải lên
một số dữ liệu người dùng vì vậy chúng tôi
sẽ nhảy

quay lại mô hình xem xác thực của chúng
tôi

ồ và hãy xem liệu chúng ta có bắn in đó
không

tuyên bố

vì vậy tôi sẽ lọc cái này ra với
gỡ lỗi vâng, chúng tôi thấy rằng phiên người
dùng là

không có nghĩa là chúng tôi không có người
dùng đã ký

trong vì chúng tôi chưa thêm bất kỳ người
dùng nào vào

ứng dụng

Main subtitle

so i'm going to add two functions here
 for now guys i'm going to say funk login
 with email email string
 password
 string
 and then i'm going to add another
 function
 that says func
 register
 with email
 email string
 password
 string full name
 string
 username
 string
 awesome
 and
 that is going to be it for now
 um
 so obviously this is gonna log our user
 in this is gonna sign our user up let's
 hop back into our login view just to see
 where we're gonna actually call these
 functions so for now i'm just gonna say
 print
 debug
 login with email
 email
 and here let's add a print statement
 that says register with email
 email
 and let's see how we're going to
 implement our view model in our other

Second subtitle

vì vậy tôi sẽ thêm hai chức năng ở đây
 bây giờ các bạn, tôi sẽ nói đăng nhập funk
 với chuỗi email email
 mật khẩu
 sợi dây
 và sau đó tôi sẽ thêm một cái khác
 chức năng
 điều đó nói rằng func
 đăng ký
 với email
 chuỗi email
 mật khẩu
 chuỗi tên đầy đủ
 sợi dây
 tên tài khoản
 sợi dây
 tuyệt vời
 Và
 đó sẽ là nó bây giờ
 ô
 vì vậy rõ ràng điều này sẽ đăng nhập người
 dùng của chúng tôi
 trong phần này sẽ đăng ký người dùng của
 chúng tôi, hãy
 nhảy trở lại chế độ xem đăng nhập của
 chúng tôi chỉ để xem
 nơi chúng ta sẽ thực sự gọi những thứ này
 chức năng vì vậy bây giờ tôi sẽ chỉ nói
 in
 gỡ lỗi
 đăng nhập bằng email
 e-mail
 và ở đây hãy thêm một câu lệnh in
 nói rằng đăng ký với email
 e-mail
 và hãy xem chúng ta sẽ làm như thế nào
 thực hiện mô hình xem của chúng tôi trong
 khác của chúng tôi

Main subtitle

views and actually call these functions and let's actually just go ahead and we can keep that guy there for now might be useful later

but let's go back to our login view

and we're going to add another instance of that view model so we're going to say dot environment object

```
var view model auth view model suite
```

and down here where we say print sign in here

we're going to say view model dot login with email

and password and the email and password guys are these text properties that we

created up here these state variables so let's do the same thing in our

registration view

instead of sign up here

we're going to add uh some some actual code

and it's going to but sorry first we need to say environment object bar.viewmodel off.viewmodel

so once again guys this concept is really important to understand it's very helpful in swift ui we notice that we

are not initializing this view model in multiple places it's only been initialized in one place which is the at file and then we launch the content view

Second subtitle

lượt xem và thực sự gọi các chức năng này và chúng ta hãy thực sự tiếp tục và chúng ta có thể giữ anh chàng đó ở đó bây giờ có thể là

hữu ích sau này

nhưng hãy quay lại giao diện đăng nhập của chúng ta

và chúng tôi sẽ thêm một trường hợp khác của mô hình xem đó, vì vậy chúng tôi sẽ nói đối tượng môi trường dấu chấm

```
var xem mô hình auth xem mô hình
```

Thượng hạng

và dưới đây nơi chúng tôi nói in đăng nhập đây

chúng ta sẽ nói xem mô hình dot đăng nhập với email

và mật khẩu và email và mật khẩu

các bạn là những thuộc tính văn bản mà chúng tôi

tạo ra ở đây các biến trạng thái này

vì vậy chúng ta hãy làm điều tương tự trong của chúng tôi

xem đăng ký

thay vì đăng ký tại đây

chúng tôi sẽ thêm uh một số thực tế mã số

và nó sẽ nhưng xin lỗi đầu tiên chúng tôi cần phải nói môi trường

chế độ xem thanh đối tượng

tắt chế độ xem

vì vậy một lần nữa các bạn khái niệm này là thực sự quan trọng để hiểu nó rất

hữu ích trong Swift ui, chúng tôi nhận thấy rằng chúng tôi

không khởi tạo mô hình xem này trong

nhiều nơi nó chỉ được

được khởi tạo ở một nơi đó là tại

tệp và sau đó chúng tôi khởi chạy chế độ

Main subtitle

let's actually add this as an environment object to the navigation view that's my bad so let's say dot environment object view model right so we add this as an environment object and because this login view is part of the same environment as the content view right because it's set up in the content view so it's part of the same environment we get access to that view model there and it's the same thing with the content view and it's going to be the same thing with the registration view so that's really really cool we don't have to worry about passing it around or doing any sort of you know initializing it multiple times it's all just taken care of through this environment object property which is really really helpful so anyway let's go back to this button and we're going to say.viewmodel.register with email password full name username and let's just do let's break this up a little bit boom now let's run our app guys and see if we

Second subtitle

xem nội dung hãy thực sự thêm cái này như một đối tượng môi trường để điều hướng xem đó là xấu của tôi vì vậy hãy nói đối tượng môi trường dấu chấm xem mô hình đúng vậy nên chúng tôi thêm cái này làm môi trường đối tượng và bởi vì chế độ xem đăng nhập này là một phần của cùng một môi trường như nội dung xem đúng bởi vì nó thiết lập trong chế độ xem nội dung để nó là một phần của cùng một môi trường chúng ta có quyền truy cập vào mà xem mô hình ở đó và đó là điều tương tự với nội dung xem và nó sẽ giống như vậy với chế độ xem đăng ký vì vậy đó là thực sự rất tuyệt, chúng ta không cần phải lo lắng về việc vượt qua nó xung quanh hoặc làm bất kỳ loại nào bạn biết khi khởi tạo nó nhiều lần tất cả chỉ được chăm sóc thông qua đối tượng môi trường này tài sản thực sự thực sự hữu ích Vì vậy, dù sao đi nữa, hãy quay lại nút này và chúng ta sẽ nói.viewmodel.register với email mật khẩu Họ và tên tên tài khoản và chúng ta hãy phá vỡ điều này một chút bùng nổ bây giờ hãy chạy ứng dụng của chúng tôi và xem liệu chúng tôi có

Main subtitle

are getting back those print statements that we added in our off view model just to let us know that we're successfully connected to it and that we're able to call functions that are a part of that view model so it's going to show us that log in with email and we don't see anything there let's actually type an email in test gmail.com hit that and you guys notice that our print statement says log in with test gmail.com so that's looking great we go here let's add another email uh something gmail.com let's hit sign up and it says register with email something gmail.com so that is looking awesome next step we know so once again what that tells us is that we have successfully connected our view model to our views it's being used as an environment object which is great we have some presentation logic set up and the main effect of that guys is once the user logs in it's always going to just take them straight to the main interface of the application they won't have to

Second subtitle

đang lấy lại những câu lệnh in đó mà chúng tôi đã thêm vào mô hình tắt chế độ xem của mình để cho chúng tôi biết rằng chúng tôi đang thành công kết nối với nó và chúng tôi có thể gọi các chức năng là một phần của điều đó xem mô hình vì vậy nó sẽ cho chúng ta thấy rằng đăng nhập bằng email và chúng tôi không thấy bắt cứ điều gì ở đó hãy thực sự gõ một gửi email vào kiểm tra gmail.com nhấn vào đó và các bạn nhận thấy rằng chúng tôi tuyên bố in cho biết đăng nhập bằng thử nghiệm gmail.com vì vậy trông thật tuyệt, chúng tôi đi ở đây hãy thêm một email khác uh thứ gì đó gmail.com nhấn đăng ký và nó nói đăng ký với email gì đó gmail.com vì vậy điều đó trông thật tuyệt bước tiếp theo chúng ta biết như vậy một lần nữa điều đó cho chúng ta biết là chúng ta đã kết nối thành công chế độ xem của chúng tôi mô hình theo quan điểm của chúng tôi, nó đang được sử dụng như một đối tượng môi trường thật tuyệt vời chúng ta có một số logic trình bày được thiết lập và tác dụng chính của những kẻ đó là một khi người dùng đăng nhập nó sẽ luôn luôn đưa họ thẳng đến giao diện chính của ứng dụng họ sẽ không phải

Main subtitle

log in every time right that's what that presentation logic and the content view is doing it says hey if they've already

logged in just take them to the main interface if not take them to the login view so

this is obviously only going to happen when the app launches and the user hasn't signed up yet or if they physically log out of the application so this is looking really really good it's super slick now we're going to get into actually

adding the authentication code to these functions so we can actually authenticate a user with our back end

so we are going to hop into this register function guys because we obviously can't log someone in if we

haven't registered them yet right so

registering this user is just going to involve grabbing these properties that we get from the view when we call this

function and uploading it to the database

so we first have to authenticate the user and then we're going to save their

information in the database it's two separate processes and it'll make more

sense when we actually write the code so let's go ahead and just delete this

Second subtitle

đăng nhập mọi lúc đúng như vậy đó logic trình bày và chế độ xem nội dung đang làm điều đó hãy nói này nếu họ đã làm rồi

đăng nhập chỉ cần đưa họ đến chính giao diện nếu không đưa họ đến đăng nhập xem như vậy

điều này rõ ràng là sẽ chỉ xảy ra khi ứng dụng khởi chạy và người dùng chưa đăng ký hoặc nếu họ

đăng xuất vật lý khỏi ứng dụng để cái này trông thực sự rất tốt nó siêu bóng bẩy bây giờ chúng ta sẽ đi vào Thực ra

thêm mã xác thực vào những các chức năng để chúng ta thực sự có thể xác thực người dùng với phần cuối của chúng tôi

vì vậy chúng ta sẽ nhảy vào đây chức năng đăng ký bởi vì chúng tôi rõ ràng là không thể đăng nhập ai đó nếu chúng tôi

chưa đăng ký đúng không Vì thế

đăng ký người dùng này chỉ là đi đến liên quan đến việc lấy các thuộc tính này chúng tôi nhận được từ chế độ xem khi chúng tôi gọi đây là

chức năng và tải nó lên cơ sở dữ liệu

vì vậy trước tiên chúng ta phải xác thực người dùng và sau đó chúng tôi sẽ lưu của họ

thông tin trong cơ sở dữ liệu đó là hai các quy trình riêng biệt và nó sẽ tạo ra nhiều hơn

cảm giác khi chúng ta thực sự viết mã vì vậy hãy tiếp tục và chỉ cần xóa cái này

Main subtitle

print statement and we're going to say auth.auth.createUser with email password and completion so guys this is obviously code that we're

getting from firebase this is the whole reason we added this swift package because all this code here this dot create user function is code that firebase has written that we have integrated into our project so that we

can use it to communicate with the api so obviously this function creates a user

and you see here that we need an email

we need a password and then we get some

sort of callback and we're going to go over what that does in a second obviously we just pass in these properties email password and then we're

going to tab over to this completion and we're going to hit enter okay

so we notice that we get back some sort

of result

and some sort of potential error okay

so obviously something might go wrong in

the authentication process so we might get an error back but we also might not

which is why this property is optional so the first thing we're going to do is check to see if an error happened maybe

Second subtitle

tuyên bố in và chúng tôi sẽ nói auth.auth.createUser với mật khẩu email và hoàn thành như vậy các bạn, đây rõ ràng là mã mà chúng tôi đang

nhận được từ firebase đây là toàn bộ lý do chúng tôi thêm gói Swift này bởi vì tất cả mã này ở đây dấu chấm này tạo chức năng người dùng là mã firebase đã viết rằng chúng tôi có tích hợp vào dự án của chúng tôi để chúng tôi

có thể sử dụng nó để giao tiếp với api vì vậy rõ ràng chức năng này tạo ra một người dùng

và bạn thấy ở đây rằng chúng tôi cần một email

chúng tôi cần một mật khẩu và sau đó chúng tôi nhận được một số loại gọi lại và chúng tôi sẽ đi hơn những gì nó làm trong một giây rõ ràng là chúng ta chỉ vượt qua những mật khẩu email thuộc tính và sau đó chúng tôi

đi đến tab để hoàn thành này và chúng ta sẽ nhấn enter nhé

vì vậy chúng tôi nhận thấy rằng chúng tôi lấy lại một số loại kết quả

và một số loại lỗi tiềm năng được vì vậy rõ ràng một cái gì đó có thể đi sai trong

quá trình xác thực để chúng tôi có thể nhận lại lỗi nhưng chúng tôi cũng có thể không

đó là lý do tại sao tài sản này là tùy chọn vì vậy điều đầu tiên chúng ta sẽ làm là kiểm tra xem có lỗi xảy ra không

Main subtitle

the user's email was invalid maybe their password didn't meet the password requirements maybe something went wrong with the back end maybe the user already exists who knows what it might be right so we're going to say if let error equals error oops error error we're going to say print debug failed to register with error and then we're just going to say error dot localized description and then we're going to return so basically what's going to happen is this is an if-let statement it's going to check to see if the error is nil or not if it's not it's going to store the error in this let constant and then we're going to print this statement out right and it's going to give us some description about what happened or what went wrong and then we're going to return out of the function because we obviously don't want to keep going with the sign up flow if something went wrong so now we're going to go here and we're going to say guard let user equal results dot user else return so this is we're accessing this result as

Second subtitle

email của người dùng không hợp lệ có thể họ mật khẩu không đáp ứng mật khẩu yêu cầu có thể đã xảy ra sự cố với mặt sau có thể người dùng đã tồn tại ai biết nó có thể đúng vì vậy chúng ta sẽ nói if let error bằng lỗi lỗi rất tiếc lỗi chúng ta sẽ nói in gõ lỗi không thể đăng ký với lỗi và sau đó chúng ta sẽ nói lỗi chấm mô tả bản địa hóa và sau đó chúng ta sẽ trở lại về cơ bản điều gì sẽ xảy ra đây là một câu lệnh if-let nó sẽ kiểm tra xem lỗi có phải là không hay không nếu đó là không phải nó sẽ lưu trữ lỗi trong điều này hãy để nó liên tục và sau đó chúng ta sẽ đi để in tuyên bố này ra bên phải và nó sẽ cung cấp cho chúng tôi một số mô tả về những gì đã xảy ra hoặc những gì đã sai và sau đó chúng ta sẽ trở về chức năng bởi vì chúng tôi rõ ràng là không muốn tiếp tục với quy trình đăng ký nếu như đã xảy ra sự cố vì vậy bây giờ chúng ta sẽ đi đến đây và chúng ta sẽ nói bảo vệ để người dùng bình đẳng người dùng chấm kết quả khác trở lại Vì thế đây là chúng tôi đang truy cập kết quả này như

Main subtitle

part of the callback

and um we get access to this user

property and if you guys were to hold down option and click on that user you guys notice that it's this user type

which is actually the same type that we have up here it's the it's part of the firebase auth library

so

if we get the user back then we're going

to say

self

dot user session equals user

just like that

and let's go ahead and just hit command b to make sure our code still works

okay looks pretty good to me so you guys notice that we're able to set this property to this user guy right here

because they're of the same data type and now

something else we have to do is we've

through this process we've successfully

authenticated a user but we haven't

actually uploaded any of their information to the database so i'm going to stop here and actually test this out in the simulator so i can

show you guys what i mean

and let's actually just go ahead and add

Second subtitle

một phần của cuộc gọi lại

và ừm, chúng tôi có quyền truy cập vào người dùng này

tài sản và nếu các bạn đã nắm giữ tùy chọn xuống và nhấp vào người dùng đó các bạn chú ý rằng đó là loại người dùng này

mà thực sự là cùng một loại mà chúng tôi có ở đây nó là một phần của thư viện xác thực firebase

Vì thế

nếu chúng tôi lấy lại được người dùng thì chúng tôi sẽ

để nói

bản thân

phiên người dùng chấm bằng người dùng cũ thế

và hãy tiếp tục và chỉ cần nhấn lệnh

b để đảm bảo mã của chúng tôi vẫn hoạt động

được rồi có vẻ khá tốt với tôi vì vậy các bạn lưu ý rằng chúng tôi có thể thiết lập điều này tài sản cho anh chàng người dùng này ngay tại đây

vì chúng có cùng kiểu dữ liệu

và bây giờ

một cái gì đó khác chúng ta phải làm là chúng ta đã

thông qua quá trình này, chúng tôi đã thành công

đã xác thực người dùng nhưng chúng tôi chưa

thực sự tải lên bất kỳ của họ

thông tin vào cơ sở dữ liệu vì vậy tôi sẽ dừng lại ở đây và thực sự

kiểm tra điều này trong trình giả lập để tôi có thể

cho các bạn thấy ý tôi là gì

và hãy thực sự tiếp tục và thêm

Main subtitle

a print statement that says print debug registered user successfully and let's go ahead and actually print out user is self that user session all right guys so let's go ahead and run this and see if it actually creates a user for us this is gonna be pretty exciting so go ahead and actually get your um what's it called authentication tab opened up in firebase and let's go ahead and sign a user up so we're just going to say test gmail.com username is test full name is test test password let's just do q q q q q q and you guys notice that we have to make this a secure field we're going to do that in just a second i just want to see if this works so let's go ahead and hit

sign up oh and check that out right you guys noticed that it took me directly to the main interface because that means we successfully registered our user so that's really cool and i'll explain why that happened as well and you guys will notice that here

Second subtitle

một câu lệnh in có nội dung gõ lỗi in đăng ký người dùng thành công và hãy tiếp tục và thực sự in ngoài người dùng là bản thân phiên người dùng đó được rồi các bạn, vì vậy hãy tiếp tục và chạy điều này và xem nếu nó thực sự tạo ra một người dùng đối với chúng tôi điều này sẽ rất đẹp thú vị vì vậy hãy tiếp tục và thực sự nhận được um của bạn nó được gọi là gì tab xác thực đã mở trong firebase và hãy tiếp tục và đăng ký người dùng chúng tôi sẽ nói thử nghiệm gmail.com tên người dùng là kiểm tra tên đầy đủ là kiểm tra kiểm tra mật khẩu hãy làm q q q q q và các bạn lưu ý rằng chúng ta phải làm đây là một lĩnh vực an toàn chúng ta sẽ làm rằng chỉ trong một giây tôi chỉ muốn xem nếu điều này hoạt động thì hãy tiếp tục và nhấn đăng ký ô và kiểm tra xem có đúng không các bạn nhận thấy rằng nó đã đưa tôi trực tiếp đến các giao diện chính vì điều đó có nghĩa là chúng tôi đăng ký thành công người dùng của chúng tôi vì vậy điều đó thực sự tuyệt vời và tôi sẽ giải thích tại sao điều đó cũng xảy ra và các bạn sẽ chú ý rằng ở đây

Main subtitle

at first the user session was nil and

then we get this registered user

successfully and then we get back an
actual user it's just some weird object

type but

um that means we've actually set this

user session property to something and

guys this is a published variable so

whenever this thing gets set it's going

to publish that notification to the view
that is utilizing this view model

and so if we go back to our content view

this logic here depends on that user

session property so when that user
session property gets published it's
going to reconfigure the view run back
through this logic and go oh hey the
user session's not nil anymore it got

published and now that's why as soon as

we sign up it shows us this main

interface view which is really really
really freaking cool right so now if we
run the app again

it's going to have a logged in user you
guys will notice that this print
statement says hey you guys have a user

and it's now showing us
the actual main interface of the app

Second subtitle

lúc đầu, phiên người dùng là không và
sau đó chúng tôi có được người dùng đã
đăng ký này

thành công và sau đó chúng tôi nhận lại một
người dùng thực tế nó chỉ là một số đối
tượng kỳ lạ

gõ nhưng

ừm, điều đó có nghĩa là chúng tôi đã thực
sự thiết lập điều này

thuộc tính phiên người dùng cho một cái gì
đó và

các bạn đây là một biến được công bố vì
vậy

bất cứ khi nào điều này được thiết lập nó sẽ
đi

để xuất bản thông báo đó để xem

đó là sử dụng mô hình xem này

và vì vậy nếu chúng ta quay lại chế độ xem
nội dung của mình

logic này ở đây phụ thuộc vào người dùng
đó

thuộc tính phiên để khi người dùng đó

thuộc tính phiên được xuất bản

sắp cấu hình lại view chạy lại

qua logic này và đi oh hey

phiên người dùng không còn là con số
không nữa

được xuất bản và bây giờ đó là lý do tại sao
ngay sau khi

chúng tôi đăng ký nó cho chúng tôi thấy
chính này

chế độ xem giao diện thực sự thực sự
thực sự tuyệt vời ngay bây giờ nếu chúng ta
chạy lại ứng dụng

nó sẽ có một người dùng đăng nhập bạn
mọi người sẽ nhận thấy rằng bản in này
tuyên bố nói này, các bạn có một người
dùng

và bây giờ nó đang cho chúng ta thấy
giao diện chính thực tế của ứng dụng

Main subtitle

which is awesome
and if we were to refresh our authentication page we notice that we have a user successfully registered in the application so this is all looking really really really good guys um but there is a but if we go to our database we haven't actually saved any of that user information anywhere so authenticating the user and actually storing their information in the database are two separate processes so for example we don't have any information about their username right now or what their full name is so if i were to go to my profile i obviously need to fetch that information from the database but all the authentication gives us is their email and their user id so i need to take this information that i have about the user their uh you know their full name and their username and also their email and i actually need to store it in the database so that's what we're going to be working on next so let's just go ahead and jump right into it so we are going to create this data

Second subtitle

đó là tuyệt vời và nếu chúng ta làm mới trang xác thực, chúng tôi nhận thấy rằng chúng tôi có một người dùng đã đăng ký thành công trong ứng dụng vì vậy đây là tất cả tìm kiếm thực sự thực sự thực sự tốt mọi người um nhưng có một nhưng nếu chúng tôi truy cập cơ sở dữ liệu của mình, chúng tôi chưa thực sự đã lưu bất kỳ người dùng nào thông tin bất cứ nơi nào vì vậy xác thực người dùng và thực sự lưu trữ thông tin của họ trong cơ sở dữ liệu là hai quá trình riêng biệt vì vậy Ví dụ chúng tôi không có bất kỳ thông tin nào về tên người dùng của họ ngay bây giờ hoặc những gì họ tên đầy đủ là vì vậy nếu tôi đi đến cửa tôi hồ sơ tôi rõ ràng cần phải tìm nạp đó thông tin từ cơ sở dữ liệu nhưng tất cả xác thực mang lại cho chúng tôi là của họ email và id người dùng của họ vì vậy tôi cần lấy thông tin này tôi có về người dùng uh của họ bạn biết tên đầy đủ của họ và họ tên người dùng và cả email của họ và tôi thực sự cần phải lưu trữ nó trong cơ sở dữ liệu vì vậy đó là những gì chúng ta sẽ làm việc tiếp theo vì vậy hãy tiếp tục và nhảy ngay vào nó như vậy chúng ta sẽ tạo dữ liệu này

Main subtitle

dictionary so we're going to say let data equal and we're going to say email and once again guys this is back in that register function and we're just going to say email just like that username we're going to say username.lowercase and this is this for this reason guys you don't want case-sensitive usernames in your app if you guys look at any app like twitter or instagram they force lowercase all the usernames because you don't want somebody to be named uh billy or billy bob and then have one uh all that be lowercase and then have someone be capital b billy bob or capital uh b capital i billy bob right you you want to force lower case everything so that's why we say dot lower case right there so no matter what the user types into that lo of that username field we're just going to automatically lower case it and then we're going to say full name full name and we don't care about capitalization there and then we're going to say uid user.uid

Second subtitle

từ điển vì vậy chúng ta sẽ nói let dữ liệu bằng nhau và chúng ta sẽ nói email và một lần nữa các bạn, điều này đã trở lại trong đó chức năng đăng ký và chúng tôi sẽ chỉ nói email như thế tên tài khoản chúng ta sẽ nói tên người dùng.chữ thường và đây là lý do này nhé các bạn bạn không muốn tên người dùng phân biệt chữ hoa chữ thường trong ứng dụng của bạn nếu các bạn xem bất kỳ ứng dụng nào như twitter hoặc instagram họ buộc chữ thường tất cả tên người dùng bởi vì bạn không muốn ai đó được đặt tên uh billy hoặc billy bob và sau đó có một uh tất cả những gì được chữ thường và sau đó có ai đó được vốn b billy bob hoặc vốn uh b vốn tôi billy bob đúng bạn bạn muốn buộc mọi thứ viết thường vì vậy đó là lý do tại sao chúng tôi nói dấu chấm chữ thường ngay tại đó vì vậy bất kể người dùng nhập gì vào lo của trường tên người dùng đó, chúng tôi chỉ cần tự động viết thường Nó và sau đó chúng ta sẽ nói Họ và tên tên đầy đủ và chúng tôi không quan tâm đến viết hoa ở đó và sau đó chúng tôi sẽ nói uid người dùng.uid

Main subtitle

and i'm going to
indent this stuff properly
so it should look just like this
so this is our data dictionary okay so

we're going to actually upload this
information to firestore
and if you guys don't know what a
dictionary is i just recommend looking
up some youtube videos or googling but
it's essentially a key value store

so every user will have the same keys
email username full name uid and they
will just have different values
associated with them so to actually
upload this data we're going to say
firebase.firebaseio.collection
users
then we're going to say dot
document
user.uid and this user guys is what we
get back from the result in our callback
so for those of you who don't know what
that means a brief explanation
is this create user function

involves reaching out to the internet
and actually creating a user and once
that process finishes
you notice that there's a separate block
here this is what's known as a
completion block

Second subtitle

và tôi sẽ
thực hiện nội dung này đúng cách
vì vậy nó sẽ trông giống như thế này
vì vậy đây là từ điển dữ liệu của chúng tôi
được rồi
chúng tôi sẽ thực sự tải cái này lên
thông tin đến firestore
và nếu các bạn không biết những gì một
từ điển là tôi chỉ khuyên bạn nên tìm kiếm
lên một số video youtube hoặc googling
nhưng
về cơ bản nó là một kho lưu trữ giá trị quan
trọng
vì vậy mọi người dùng sẽ có các khóa giống
nhau
email tên người dùng tên đầy đủ uid và họ
sẽ chỉ có các giá trị khác nhau
liên kết với họ để thực sự
tải lên dữ liệu này chúng ta sẽ nói
firebase.firebaseio.collection
người dùng
sau đó chúng ta sẽ nói dấu chấm
tài liệu
user.uid và những người dùng này là những
giá trị chúng tôi
lấy lại từ kết quả trong cuộc gọi lại của
chúng tôi
Vì vậy, đối với những người bạn không biết
những gì
điều đó có nghĩa là một lời giải thích ngắn
gọn
đây có phải là chức năng tạo người dùng
không
liên quan đến việc tiếp cận với internet
và thực sự tạo một người dùng và một lần
quá trình đó kết thúc
bạn nhận thấy rằng có một khối riêng biệt
đây là những gì được gọi là
khối hoàn thành

Main subtitle

and what we get back
from that completion or that api call
are these two properties we either get
back some result or some error as part
of that result that we get back from the
callback or within the completion block
is this result
or we could get back an error
and within this block
then we can do everything we need to do
after this create user function has been
called that's just a very brief
explanation of it so now we're going to
say dot set data
and we actually want a completion block
so let's see set data completion we're
going to pass in our data
and then on the completion we don't
really need to handle the error because
nothing should ever really go wrong
there
we're just going to say print debug
did
upload
user
data so
you guys will notice that the app is
automatically launching to this uh
interface right the main interface we
want to get back to our login flow so we
actually need to implement the

Second subtitle

và những gì chúng ta nhận lại
từ lần hoàn thành đó hoặc cuộc gọi api đó
hai thuộc tính này chúng ta có được không
quay lại một số kết quả hoặc một số lỗi như
một phần
của kết quả mà chúng tôi nhận được trở lại
từ
gọi lại hoặc trong khối hoàn thành
đây là kết quả
hoặc chúng tôi có thể lấy lại một lỗi
và trong khối này
sau đó chúng ta có thể làm mọi thứ chúng
ta cần làm
sau khi chức năng tạo người dùng này đã
được
được gọi là đó chỉ là một cái gì đó rất ngắn
gọn
giải thích về nó vì vậy bây giờ chúng ta sẽ
nói dữ liệu tập hợp dấu chấm
và chúng tôi thực sự muốn có một khối
hoàn thành
Vì vậy, hãy xem hoàn thành dữ liệu thiết lập,
chúng tôi
sẽ vượt qua trong dữ liệu của chúng tôi
và sau đó khi hoàn thành, chúng tôi không
thực sự cần phải xử lý lỗi vì
không có gì nên thực sự đi sai
ở đó
chúng ta sẽ nói gỡ lỗi in
làm
tải lên
người dùng
dữ liệu như vậy
các bạn sẽ nhận thấy rằng ứng dụng này là
tự động khởi chạy tới cái này uh
giao diện ngay giao diện chính chúng ta
muốn quay lại quy trình đăng nhập của
chúng tôi vì vậy chúng tôi
thực sự cần phải thực hiện các

Main subtitle

functionality for this log out button
before we can get this stuff done and
then we also need to make the password a
secure field
so we're gonna
proceed with those two things right now
so when we hit this log out button we
actually want it to log us out of the
application so we're going to add
another function here that's going to be
called sign out
or you know sign out
just like that
and
we're going to do a couple things we're
going to set user session to nil
we're going to set and then we're just
going to say try
auth.auth.sign out
okay so this is doing two things
we're going to set this user session to
nil and once again guys this is a
published property so it's going to go
back to our content view sorry and it's
going to say oh hey this user session
has changed that means i need to
reconfigure this view
and run this logic again so now user
session once it gets set to nil it's
going to show us with the login view
so that handles everything on the client
side or locally right

Second subtitle

chức năng cho nút đăng xuất này
trước khi chúng ta có thể hoàn thành công
việc này và
sau đó chúng ta cũng cần đặt mật khẩu
lĩnh vực an toàn
vì vậy chúng ta sẽ
tiến hành với hai điều đó ngay bây giờ
vì vậy khi chúng tôi nhấn nút đăng xuất này,
chúng tôi
thực sự muốn nó đăng xuất chúng tôi ra
khỏi
ứng dụng vì vậy chúng tôi sẽ thêm
một chức năng khác ở đây sẽ là
được gọi là đăng xuất
hoặc bạn biết đăng xuất
cú thể
Và
chúng tôi sẽ làm một vài điều chúng tôi
sẽ đặt phiên người dùng thành không
chúng ta sẽ thiết lập và sau đó chúng ta chỉ
sẽ nói thử
auth.auth.đăng xuất
được rồi vì vậy điều này đang làm hai việc
chúng tôi sẽ đặt phiên người dùng này
thành
không và một lần nữa các bạn, đây là một
tài sản đã xuất bản vì vậy nó sẽ đi
quay lại chế độ xem nội dung của chúng tôi
xin lỗi và đó là
sẽ nói oh hey phiên người dùng này
đã thay đổi điều đó có nghĩa là tôi cần phải
định cấu hình lại chế độ xem này
và chạy lại logic này để bây giờ người dùng
phiên khi nó được đặt thành không
sẽ cho chúng tôi thấy với chế độ xem đăng
nhập
để xử lý mọi thứ trên máy khách
bên hoặc bên phải cục bộ

Main subtitle

but we actually physically need to log out of our back uh log our user out in our back end as well so we need to handle things on two different fronts on the client side and on the server side so if i were to just set user session to nil so let's go ahead and actually see this working live so you guys can actually understand this so let's see off viewmodel i'm just going to set user session to nil and i'm going to delete this or comment this out this function is what logs us out on the back end so i'm going to add a comment here signs user out on backend or on server and then this user session guy let's see this uh sets user session to nil so we show login view okay so we have our comments um and before this actually works we need to hop back and actually call this function in our side menu guide

Second subtitle

nhưng chúng tôi thực sự cần phải đăng nhập ra khỏi lưng của chúng tôi uh đăng xuất người dùng của chúng tôi trong phần cuối của chúng tôi vì vậy chúng ta cần xử lý mọi thứ trên hai mặt trận khác nhau ở phía máy khách và trên máy chủ bên vì vậy nếu tôi chỉ đặt người dùng phiên thành con số không, vì vậy hãy tiếp tục và thực sự thấy điều này làm việc trực tiếp vì vậy bạn các chàng trai thực sự có thể hiểu điều này vì vậy hãy tiễn viewmodel tôi sẽ đặt phiên người dùng thành không và tôi sẽ xóa cái này hoặc nhận xét điều này ra chức năng này là thứ đăng xuất chúng tôi trên back end vì vậy tôi sẽ thêm một bình luận đây dấu hiệu người dùng ra ngoài phụ trợ hoặc trên máy chủ và sau đó anh chàng phiên người dùng này hãy xem cái này à đặt phiên người dùng đến con số không để chúng tôi hiển thị chế độ xem đăng nhập được rồi vì vậy chúng tôi có ý kiến của chúng tôi ô và trước khi điều này thực sự hoạt động, chúng ta cần để nhảy trở lại và thực sự gọi đây là chức năng trong hướng dẫn menu bên của chúng tôi

Main subtitle

so when we hit that log out button we want to call this function so we need to go to our side menu views side menu view and we notice that we have this print handle log out here so that's uh just where we're going to stick this function but we do need access to the view model so inside of the side menu view we're going to say at environment object bar i think i spelled that wrong at in vi ron meant object view model auth view model cool and i think we should get access to that because it's all part of the same environment so now we're going to go here and we're going to say view model dot sign out and build that so it's actually getting confused guys because we're using this view model term um sort of ambiguously right so we have.viewmodel here and.viewmodel here so let's just replace this as off.viewmodel and then down here say auth.viewmodel.signout okay

Second subtitle

vì vậy khi chúng tôi nhấn nút đăng xuất đó, chúng tôi muốn gọi chức năng này vì vậy chúng ta cần đi đến menu phụ của chúng tôi chế độ xem menu bên và chúng tôi nhận thấy rằng chúng tôi có tay cầm in này đăng xuất tại đây vì vậy đó chỉ là nơi chúng ta sẽ đến dính chức năng này nhưng chúng tôi cần truy cập vào mô hình xem nên bên trong chế độ xem menu bên chúng ta sẽ nói tại thanh đôi tượng môi trường tôi nghĩ rằng tôi đã đánh vần sai ở trong vi ron có nghĩa là sự vật xem mô hình auth xem mô hình tuyệt vời và tôi nghĩ rằng chúng tôi nên có quyền truy cập vào đó bởi vì nó tất cả các phần của cùng một môi trường vì vậy bây giờ chúng ta sẽ đi đây và chúng ta sẽ đi để nói xem mô hình chấm đăng xuất và xây dựng nó Vì vậy, nó thực sự khiến mọi người bối rối bởi vì chúng tôi đang sử dụng thuật ngữ mô hình chế độ xem này ừm hơi mơ hồ đúng không vì vậy chúng tôi có.viewmodel ở đây và.viewmodel ở đây vì vậy hãy thay thế cái này thành tắt.viewmodel rồi xuống đây nói auth.viewmodel.signout được rồi

Main subtitle

and let's run this and see what we get

so

just a quick recap we're going to notice

that this is going to sign us out or

appear to sign us out

basically it's going to you know

set the user session to nil on the

client side but it's not actually going

to do anything on the back end so this

that change won't actually persist once

we run the app again so i'll show you

guys what i mean

so i go here and i log out right and it

takes me to the login page so on the
surface everything looks good

it says hey i clicked the logout button
and i've been taken to a login page

makes sense

but if i run the app again it should

take me back to the login page but you

guys are going to notice that it doesn't
it takes me back to the main interface
you're probably like well what the hell
dude i logged out right well that's

because we didn't really log the user

out so we go back to auth view model and

we've set this user session to nil but

this is just a local property so when we

Second subtitle

và hãy chạy cái này và xem những gì chúng ta nhận được

Vì thế

chỉ là một bản tóm tắt nhanh chúng ta sẽ
nhận thấy

rằng điều này sẽ đăng xuất chúng tôi hoặc
xuất hiện để đăng xuất chúng tôi
về cơ bản nó sẽ cho bạn biết

đặt phiên người dùng thành không trên
phía khách hàng nhưng nó không thực sự
diễn ra

để làm bất cứ điều gì ở mặt sau vì vậy điều
này

thay đổi đó sẽ không thực sự tồn tại một lần
chúng tôi chạy lại ứng dụng vì vậy tôi sẽ chỉ
cho bạn

ý tôi là gì

vì vậy tôi vào đây và tôi đăng xuất ngay và
nó

đưa tôi đến trang đăng nhập để trên
bề mặt mọi thứ có vẻ tốt

Nó nói này, tôi đã nhấp vào nút đăng xuất
và tôi đã được đưa đến một trang đăng
nhập

có ý nghĩa

nhưng nếu tôi chạy lại ứng dụng thì nó sẽ
như vậy

đưa tôi trở lại trang đăng nhập nhưng bạn

mọi người sẽ nhận thấy rằng nó không
nó đưa tôi trở lại giao diện chính

bạn có lẽ cũng giống như cái quái gì vậy
anh bạn tôi đã đăng xuất đúng rồi đó

bởi vì chúng tôi đã không thực sự đăng
nhập người dùng

out để chúng tôi quay lại mô hình xem xác
thực và

chúng tôi đã đặt phiên người dùng này
thành không nhưng

đây chỉ là một tài sản địa phương vì vậy khi
chúng tôi

Main subtitle

relaunch the app and we re-initialize

this auth view model we set self.user

session equal to something that depends on reaching out to the firebase servers and checking to see if there's a user logged in well guess what there is because we never sign them out

so this user session property will still

have a value and it'll take us back to the main interface so

that's where if we comment out this line of code this code right here this line will actually sign us out on the back end as well

so that when we log out it does uh it logs us out on both fronts client

side and on the back end

so if we run this

and hit sign out again let's just run through that process one more time to make sure everything is fine

so

i'm going to log out and everything still looks the same on the surface but now you guys will notice if i run the app again

that now it takes me to the what's it called the login page and it's recognizing that the user session is nil so when we have this line of code here and this stuff reaches out to the firebase servers to check if someone's logged in it's like oh hey guess what

Second subtitle

khởi chạy lại ứng dụng và chúng tôi khởi tạo lại

mô hình chế độ xem xác thực này, chúng tôi đặt self.user

phiên bằng với một cái gì đó phụ thuộc

khi liên hệ với các máy chủ firebase

và kiểm tra xem có người dùng nào không

đã đăng nhập đoán xem có gì

bởi vì chúng tôi không bao giờ đăng xuất chúng

vì vậy thuộc tính phiên người dùng này sẽ vẫn

có một giá trị và nó sẽ đưa chúng ta trở lại giao diện chính vậy

đó là nơi nếu chúng ta nhận xét dòng này của mã này ngay tại đây dòng này thực sự sẽ đăng xuất chúng tôi ở mặt sau cũng kết thúc

để khi chúng ta đăng xuất thì nó làm uh nó đăng xuất chúng tôi trên cả hai máy khách

bên và ở mặt sau

vì vậy nếu chúng ta chạy cái này

và nhấn đăng xuất một lần nữa, hãy chạy thông qua quá trình đó một lần nữa để đảm bảo mọi thứ đều ổn

Vì thế

tôi sẽ đăng xuất và mọi thứ

vẫn trông giống nhau trên bề mặt nhưng bây giờ các bạn sẽ chú ý nếu tôi chạy

ứng dụng lại

rằng bây giờ nó đưa tôi đến

nó được gọi là trang đăng nhập và nó là gì nhận ra rằng phiên người dùng là không

vì vậy khi chúng tôi có dòng mã này ở đây và những thứ này tiếp cận với

máy chủ firebase để kiểm tra xem ai đó đăng nhập nó giống như oh hey đoán

Main subtitle

you signed me out so with this line of code so nobody's logged in so then it takes us back here so that's looking really good now let's register another user guys and see what happens when we actually try to upload this data to the database so i'm going to say test 2 gmail.com test test and just qqq hit sign up fail to register email address is already in use by another account okay so i think i registered that off screen yeah so um let me just delete this account delete but what that shows guys is that our error functionality is actually working right so um i had registered that account um off-camera and because i tried to register that account again firebase doesn't allow you to register accounts on the same app that have a duplicate email so it threw an error and printed out that error message for us which is looking really good so but now since i deleted the account and i hit sign up again everything's going to work just fine

Second subtitle

những gì bạn đã đăng xuất tôi vì vậy với dòng này mã để không ai đăng nhập nên nó đưa chúng tôi trở lại đây vì vậy đó là tìm kiếm thật sự tốt Hiện nay hãy đăng ký người dùng khác đi các bạn và xem điều gì xảy ra khi chúng ta thực sự cố gắng tải dữ liệu này lên cơ sở dữ liệu vì vậy tôi sẽ nói bài kiểm tra 2 gmail.com kiểm tra kiểm tra và chỉ qqq nhấn đăng ký không thể đăng ký địa chỉ email là đã được sử dụng bởi một tài khoản khác rồi nhé vì vậy tôi nghĩ rằng tôi đã đăng ký ngoài màn hình vâng vậy um để tôi xóa cái này xóa tài khoản nhưng điều đó cho mọi người thấy rằng chúng ta chức năng lỗi đang thực sự hoạt động đúng vậy ừm tôi đã đăng ký tài khoản đó ừm tắt máy ảnh và bởi vì tôi đã cố gắng đăng ký lại tài khoản đó firebase không cho phép bạn đăng ký tài khoản trên cùng một ứng dụng có một trùng lặp email để nó gây ra lỗi và in ra thông báo lỗi đó cho chúng tôi mà trông thực sự tốt vậy nhưng bây giờ kể từ khi tôi xóa tài khoản và tôi nhấn đăng ký một lần nữa mọi thứ sẽ hoạt động tốt thôi

Main subtitle

i run the or refresh the page test2
 gmail.com is there i have this new user
 id and you guys will notice now that if
 i go to firestore database i actually
 have some data in the database here
 right and it's test2 gmail.com it's
 exactly the information i typed in
 um
 what's it called
 on the signup page you guys will notice
 that test is lower case on the username
 which is great this user id matches this
 document id and we've actually saved
 some information in the database now we

get back this print statement as did
 upload user data so that's looking
 really good
 we've successfully uploaded some
 information to the database and we'll be
 able to fetch this information later to
 do things like display it user
 information in a tweet right when we
 want to like look at their full name and
 their username we're going to pull that
 information from the database and also
 we'll be able to go to any user's

profile and view their information there
 as well so this is looking really really
 good guys
 um
 now uh that we got to do a couple more
 things though
 um

Second subtitle

tôi chạy hoặc làm mới trang test2
 gmail.com ở đó tôi có người dùng mới này
 id và bây giờ các bạn sẽ nhận thấy rằng nếu
 tôi đi đến cơ sở dữ liệu firestore tôi thực sự
 có một số dữ liệu trong cơ sở dữ liệu ở đây
 phải và đó là test2 gmail.com đó là
 chính xác thông tin tôi đã nhập
 ô
 nó được gọi là gì
 trên trang đăng ký các bạn sẽ chú ý
 bài kiểm tra đó là chữ thường trên tên
 người dùng
 thật tuyệt id người dùng này phù hợp với
 điều này
 id tài liệu và chúng tôi đã thực sự lưu
 một số thông tin trong cơ sở dữ liệu bây giờ
 chúng tôi
 lấy lại câu lệnh in này như đã làm
 tải lên dữ liệu người dùng để tìm kiếm
 thật sự tốt
 chúng tôi đã tải lên thành công một số
 thông tin vào cơ sở dữ liệu và chúng tôi sẽ
 có thể lấy thông tin này sau để
 làm những việc như hiển thị nó người dùng
 thông tin trong một tweet ngay khi chúng tôi
 muốn thích nhìn vào tên đầy đủ của họ và
 tên người dùng của họ, chúng tôi sẽ kéo nó
 thông tin từ cơ sở dữ liệu và cũng
 chúng tôi sẽ có thể truy cập bất kỳ người
 dùng nào
 hồ sơ và xem thông tin của họ ở đó
 vì vậy điều này trông thực sự thực sự
 Người tốt
 ô
 bây giờ chúng ta phải làm thêm một vài điều
 nữa
 mọi thứ mặc dù
 ô

Main subtitle

we want to make it so that we can actually log back in as the user with their existing credentials so we're going to hop into this login function and take care of that now so it's going to be really really similar to the registration function it's just going to be a little bit simpler right so we just have to worry about what happens on the back end we don't actually have to upload information anywhere so we can just say off.auth.sign in with email password and completion signs in using an email address and password and that's exactly what we have so we're going to pass in our email password and then once again we have access to this callback we're going to get back either a result or an error hit enter name these properties result in error we're going to just handle the error in the exact same way but instead of register we're going to say fail to sign in with error and print out the error and then stop with that return and then we need to set this user session property to something so that it will um you know run back through that

Second subtitle

chúng tôi muốn làm cho nó để chúng tôi có thể thực sự đăng nhập lại với tư cách là người dùng với thông tin đăng nhập hiện có của họ vì vậy chúng tôi sẽ nhảy vào chức năng đăng nhập này và chăm sóc nó ngay bây giờ vì vậy nó sẽ thực sự thực sự tương tự như chức năng đăng ký nó sẽ chỉ là một chút đơn giản hơn đúng không vì vậy chúng ta chỉ phải lo lắng về những gì xảy ra ở mặt sau chúng ta không thực sự phải tải lên thông tin bất cứ nơi nào để chúng ta chỉ có thể nói off.auth.sign in bằng mật khẩu email và hoàn thành đăng nhập bằng địa chỉ email và mật khẩu và đó chính xác là những gì chúng tôi có vì vậy chúng tôi sẽ chuyển qua email của chúng tôi mật khẩu và sau đó một lần nữa chúng ta có truy cập vào cuộc gọi lại này, chúng tôi sẽ lấy lại kết quả hoặc lỗi Nhấn Enter đặt tên cho các thuộc tính này dẫn đến lỗi chúng ta sẽ chỉ xử lý lỗi trong chính xác theo cùng một cách nhưng thay vì đăng ký, chúng tôi sẽ nói không thể đăng nhập có lỗi và in ra lỗi và sau đó dừng lại với điều đó trở lại và sau đó chúng ta cần phải đặt thuộc tính phiên người dùng này

Main subtitle

presentation logic in the content view
and show us the main interface view so
we're going to do the same thing we did
right here self.user session
bam
and that looks really really good guys
um and then we can just add a print
statement that says you know like did
log user in
and let's go ahead and run this and see
if it works
we're already calling this function back
in the login view if you guys remember
we replaced that print statement with
viewmodel.log in with email and password
so right now we see our user session is
nil um actually guys before we do this
let's hop back up to this init method
and let's go ahead and say
printself.usersession.uid
so that way we actually get back like an
actual string instead of just some weird
like object
representation
and run that again that's my bad
okay so let's log our user in
let's start with the first use that we
created test gmail.com
qqq sign in
and we notice that we get that did log
user in and it takes us back to the main
interface of the application so the
login functionality is working here as
well that's awesome so now we'll run the
app
and we should get some user id back
right so we notice that we get back p z

Second subtitle

Main subtitle

uh blah blah blah blah

if we go to our authentication page we
notice that test at gmail's user id is
pzu so that's working really nicely now
let's see if we can log in as our other
user test2 at gmail.com

so i'm going to log out

log in as test2 gmail.com one two three
four five six sign in

excuse me and we actually have to run
the app again to get this print
statement guys and once again that's
because the off view model
only gets initialized once so that's why
this print statement only runs once

[Music]

so we run it again and we'll notice that
we get back a different user id and this
is the one associated with test2
gmail.com

and we noticed that test2 gmail.com is
the only user that's actually has
information saved in the database
so this is looking really really good
guys

um one more thing i want to do

um as a nice touch-up is actually make
this password field

uh you know be an actual password field
where you can't see the letters being
typed
and

that's going to affect the
capitalization
of the password like
we'll see that uh
swift ui they have a secure field

Second subtitle

Main subtitle
that**Second subtitle**

defaults everything to a lower case um
because that's what all passwords fields
do

because passwords are case sensitive so
anyway let's just go and uh fix this

right now

so we're going to go back into our
components go back to this custom input
field and we're going to add a property
called let

is secure

field and it's a boolean

and you notice the build bill is going
to fail let's go ahead and add that

missing property is secure field we can
say false

and we can actually um
give this a default value of false
and make it optional i think

and what we're going to do now guys is
go through some logic here

and we're going to say cut this out
we're going to say if is secure field
and else

paste the text field back into the else
block

and then here

you can copy this and replace text field
with secure field

and secure field is something that has
all the password field properties

and we can give this a default value of
false

and this way guys we don't have to add
the is secure field

input parameter every time we create

Main subtitle

this custom input field if it's not there it will just default to false so now we only need to add this to one spot we're gonna go or sorry two spots we're gonna go back to the login view so authentication views login view and this is going to for the password we're going to go here and say is secure field is true and let's just copy this and go to the registration view and replace the password guy with that as well oh we need to add a comma that's my bad comma let's run this again and let's see if the field is secure and just remember that you're going to have to capitalize your passwords first letter or you could just delete these accounts and create new accounts so anyway i'm going to say test gmail.com i'm going to capitalize the first queue two three four five six and you guys will notice that it's now obviously a secure field hit sign in and everything is still good to go um so that is looking really good guys for authentication next what we have to do is make it so that when the user hits sign up it actually

Second subtitle

Main subtitle

prompts them to add a profile photo so
that's a pretty lengthy process uh
buckle in for that one guys it's a lot
of fun and like i said it's something
that you can drag and drop into any one
of your applications that involves
uploading images so it's worth the time
to you know to figure out how it works
and how it's built but basically once
the user hits sign up and they get
authenticated it's going to take them to
a new screen that says hey please upload
a profile photo and then you know we'll
be able to actually fetch that
information from the database and like
display some user information on their
profile so it's going to be a lot of fun
let's go ahead and get started with that
now

so guys we need to create one more view
that allows the user to select a profile
photo that's going to be the easiest
part of this process so let's just go
ahead and knock that out now so in your
authentication folder in the views
folder

let's go ahead and create a new swift ui
view

and this is going to be called
photo
selector
view

and actually make it a profile photo
selector view

okay so we have this view here
and it's gonna be pretty simple but
let's get our preview

Second subtitle

Main subtitle

working

um we're gonna re-add this header
and it's just going to say like select a
profile photo

but really quickly guys we got to wait
for this preview to get going

so we got the preview working guys let's
go ahead and just create a v stack here
so we're going to say v stack

and we're going to add this auth header
off header view

uh title 1 title two

uh we're just gonna say create your
account as the first title

and title two will be select a profile
photo

and the body

we don't need that

and let's just enter right there

and

let's go ahead and hit try again

oh yeah sorry guys we don't want that no
take those brackets out so that looks uh
pretty good right um

we just want to add a spacer here and
then say dot ignores safe area

um

we could change this to like add profile
photo i think

add a profile photo yeah that looks good
um and now we just need to add like a
button here that allows the user to add
a profile photo

so uh let's see

below this

we're going to create a button
action

Second subtitle

Main subtitle

we're going to say print

pick

image here

and then our label guys

is going to be

an image that i have in the assets and
once again you guys can download these
assets the link is in the description

it's just this like circular button that
says like hey add a photo

it's pretty self-explanatory that you
click on that to add a photo

so let's go ahead and um

just add that guy right now so

we're going to say the label is going to
be

um an image

with the name plus underscore photo i'm
pretty sure that's what it's called

yeah plus underscore photo that's

dot resizable

dot frame

which is going to be

180 by 180

and then we're going to give it a

padding on the top of

44.

and

i think we also want to say like dot

scaled to fill

shouldn't really change much but

okay so that looks pretty good um

guys we noticed that the blues don't

really match um i think we can fix that

let's try to

so under resizable you can say dot

rendering mode dot template

Second subtitle

Main subtitle

and then you can say like dot uh
foreground color is color
dot system
blue and let's try it with like system
pink or something
see if it changes it yeah so that
actually is really freaking cool right
you can change the color of any one of
your buttons just like this so let's
just go ahead and make it system blue so
that it matches up with everything
okay
so that's our plus photo button
let's work on the presentation logic for
how we get to this screen right meaning
when we the user fills out this form and
they sign up
then we want to somehow magically take
them over to this add a profile photo
screen right
so
it's a little bit tricky it involves a
little bit of magic but i'm gonna show
you guys how to do it because i'm
awesome and i love you guys
so
we're gonna go back to the registration
view okay
and we're going to create this
navigation link
so on the v stack you guys are just
going to go here and add a navigation
link
navigation link
we're going to have a destination and a
label oh sorry guys navigation link
destination label and we want this is

Second subtitle

Main subtitle

active property in here as well okay
so the destination is obviously going to
be the profile photo selector view
is active
this is where the trick comes in
the label
also another trick it's just going to be
blank
excuse me so this is basically a blank
navigation link and this is active
property so there's no label guys but
we're going to be able to
activate this navigation link based on
when something happens in our off view
model
so we're going to go to our off view
model right now
so go to view model auth view model
we're going to add another published
property
we're going to say at
published
var
[Music]
or actually i think we could just use
this user session
right
so basically
when
we uh or actually not we're not going to
do that
so at publishedvar
did authenticate
user
equals false
so
excuse me

Second subtitle

Main subtitle

all right sorry guys my throat's like
super dry
um but after we set this data and we say
did upload user data in our register
function we're gonna go here
and we're gonna say
self.did authenticate user equals true
and then we're going to go back to
our registration view
and we are going to say view model
dot did authenticate user
and that is going to determine whether
or not this link is active
and i believe you have to put a dollar
sign in front of that
so basically here's the rundown of
what's going to happen guys this is
super super smooth
um
in the off view model
when we authenticate this user
this published property is going to
change its value
and then back in our registration view
so we're here we authenticate the user
that means we authenticated them and we
uploaded their information to the
database
then this navigation link will become
active once we authenticate that user
and it will take us over to the select
the profile photo page or profile photo
selector view
so that is about as smooth as it gets
homie let's run this and see if it
works
and then you know we still got a little

Second subtitle

Main subtitle

bit more work to do

uh because we actually have to uh
implement the code for the user to
select the profile photo right now this
is just the presentation logic for how
to get there

so as you can see this stuff is pretty
complicated guys but we are working
through it and best part about this is
it's free

so

let's go and we're going to say like
test like 345 at gmail.com

test test

one two three four five six we sign up
okay

so it doesn't do what we want it to do
that's interesting right so we got to
work through the logic in our auth view
model a little bit

so

here

we need to change how we configure this
user session

so we're actually going to delete this
line of code right because as soon as it
authenticates the user it sets this user
session property to some value and then
in our content view

it says oh hey user session has a value
now so now you want me to show the main
interface view but we're like hold up

wait a minute

we actually want to show the profile
photo selector view so we're not going
to set that user session property just
yet

Second subtitle

Main subtitle

and that's another reason we have this
we created this uh additional property
guys because if we try to just reuse
this same property we wouldn't be able
to get around the content view
showing us the main interface right
because that depends on this user
session property so we have to create a
different property that is unique to
the registration view
or unique to this like sort of
authentication flow for selecting a
profile photo
so now that we have that
let's uh let's log out of the app real
quick
run this again
and see if it works now
i'm pretty sure this is going to work
when i say pretty sure i mean like 99
you know
but i always got a little leave a little
bit of room for error because code is
unpredictable
so we gotta register a new user i'm
gonna say test like 888
gmail.com one two uh test
test one two three four five six
hit sign
up boom
shakashaka
takes us right over to that plus photo
or add a profile photo guy
we hit this
and that's where we actually want to
present the user with an image picker
right

Second subtitle

Main subtitle

um so this is looking really good guys
the next part
is not super fun
unfortunately swift ui still to this day
does not have like a native image picker
that you can just access through use of
swift ui
we have to work through ui kit to do
this so it's a valuable lesson to learn
because swift ui is still in its infant
stages even though we're on the third
iteration of it
but we have to be able to work with ui
kit within swift ui 2 and luckily they
give us the ability to integrate ui kit
and swift ui together so that they can
you know sort of work together in unison
and where swifty wash falls short we can
still work with ui kit to make things
happen so that's what we're going to be
doing next we're going to be adding this
sort of image picker so the user can
select the profile photo
and then once they select that they'll
be able to hit like a continue button
down here to actually take them back to
the main interface so
get excited for that guys that's what's
up next
really quickly guys i actually want to
change this to just say like setup
account
because
throughout the rest of the app the top
line is shorter than the bottom line so
i think it's cool
or better to stay consistent with that

Second subtitle

Main subtitle

just for like a smooth ui that looks a little bit messy so i think that's a little bit better but anyway with that little touch up let's go ahead and i think yeah see i look i think that looks better

um let's go ahead and create this image picker that we need to create in order to get this to work

so

we're going to go back to our components and we are going to create or sorry guys do we have a utils folder

i think that's what we need

yeah let's create a utilities folder so

let's go up to the top

and create something called utils right

and then within this utils folder we're going to create a swift file

and it's going to be called image picker

and guys this is something you thankfully only have to create one time

i'm sure you guys who have worked with swift ui before have created this

um

but you know any any app that you have that involves selecting a photo you can literally just drag and drop this into any app which is really nice so you only have to make it one time

so um let me get this opened up in my code imagepicker.swift so we're going to create a struct called image picker and it conforms to this ui view representable protocol so basically this allows us to create some ui kit ui view or create a like a view controller

Second subtitle

Main subtitle

essentially in ui kit and it translates it into a swift ui view for us which is pretty cool right so we are going to create this function called make coordinator and we have to create another function called make ui view controller uh sorry this needs to be ui view controller representable and then this goes to make ui view controller okay and then we need one more function called update uiviewcontroller and we are just going to leave this guy blank we do need to however put some stuff in here so we're going to make an extension for this image picker and we're going to call or make a class called coordinator it's going to conform to the ns object or subclass ns objects and it's also going to conform to these two protocols ui navigation controller delegate and ui image picker controller delegate so if you guys have worked with ui kit before these are the two protocols you have to conform to whenever you work with an image picker this is what this coordinator is going to do here so back up in this make coordinator we're just going to say return coordinator and we actually need to write an init for this coordinator guys so we're going to say init

Second subtitle

Main subtitle

underscore parent which is an image

picker

and we are going to make a parent
property here we're going to say let
parent

image picker and it will become clear

while we're creating this property a
little bit later on um within like the
next five minutes so if you guys are
confused by this right now don't worry
explanation is coming we're just going
to say self.parent equals parent

and we are going to initialize this
coordinator with self right because self
refers to image picker and we're asking
this uh

init to pass in some image picker
reference

which is in this case in this struct is
just self

okay um and then in make ui view
controller guys we're gonna say let
picker equal ui image picker controller
so we're actually creating a view
controller here

we're going to say picker.delegate
equals uh context

dot coordinator

and we're going to return our picker so
typically you guys are probably used to
saying picker.delegate equals self
but in this context

this needs to say itself

it's our coordinator that conforms to
the protocols that we need so we have to
say picker.delegate each is equal to the
context.coordinator

Second subtitle

Main subtitle

the context in this case just refers to some context property that is associated with this image picker class every ui view controller representable or ui view representable has to have some coordinator this sort of acts as the link that bridges the gap between ui kit and swift ui this is where we can conform to the ui kit protocols and write the code that we need to okay anyway um let's now conform to this ui image picker controller delegate here and say funk image picker controller did finish picking media with info so this is where we actually need to set some image property so we're going to go here and we're going to say guard let mo image equal info dot original image as ui image else return and here we need to be able to say like parent dot image equals this image right but we don't have an image on the parent yet because ultimately what we want to be able to do guys is get the image from this image picker and put it into our swift ui view right so right now we're building this image

Second subtitle

Main subtitle

picker so that we can do that but we need to be able to access the image that the user picked so here we're going to create a binding variable called image or we could call you know like selected image to make it a little bit more clear it's going to be ui image and then we're also going to create an environment variable called backslash.presentation mode var presentation mode and this is so that we can dismiss the image picker okay so now we'll be able to go here and say parent dot image or select an image equals image and parent dot presentation mode dot rap value dot dismiss so this is where this parent comes in guys we need to be able to link the coordinator back to the image picker so the parent is a reference to the image picker and when we initialize this guy we ask for a reference to the image picker or an image picker property essentially so then here this function requires us to make a coordinator this function is a requirement of this protocol and so is this make ui view controller so here we make the coordinator and we return this coordinator and it's like oh hey this coordinator conforms to these

Second subtitle

Main subtitle
protocols**Second subtitle**

right and in this coordinator we have access to the parent class which is this class so this is essentially a subclass of this class and this parent property is how we communicate between the two classes so here is where we get the image but we need to be able to pass it back to or over to the image picker class so we can say parent dot whatever right selected image and then we also have access to the presentation mode because when we pick that image we want to dismiss the image picker so that's pretty much it guys um we should now be able to go back to our um what's it called photo selector view and on this button we're going to add a sheet parameter we're going to say dot sheet is presented and then we need to add a property up here we're going to say at state private var show image picker equals false the buttons action is going to be show image picker dot toggle and then is presented is going to be show image picker on dismiss um we're not going to do anything with that for now

Main subtitle

but the content is going to be our image picker and it's going to ask us for a selected image right here so we're going to need to create that image we're going to say state private var image equals and we're just going to set it to a ui image just like that and it's just going to be a blank image okay and here we're going to just pass in that image so let's go ahead and run this and um guys let's actually stop the app really quick oh this is a binding property so one second um but uh what i was going to say just now is that we don't want to have to register a user every time we want to get to this part of the app so really quickly let's just go to this guy and we're going to go here and we're just going to once again comment this content view out and here we're going to say profile photo selector view we know that our presentation logic is working so now we just have to get you know the actual image picker functionality and stuff working here

Second subtitle

Main subtitle

so let's see what this guy is giving me
grief for
so let's actually create uh let's make
this a little different guys
we're going to say state private var
selected image
and that's going to be a ui image just
like that
and we're going to have a state private
var
of an actual pro of the actual profile
image
and that's going to be an image so the
difference here is the ui prefix right
so ui prefix means that it comes from ui
kit so this image guy is actually just
part of swift ui
so for our selected image down here
we're going to be able to pass in dollar
sign selected image just like that
so that looks pretty good
let's go ahead and run this it should
just take us straight over to our
what's it called
profile photo selector view
but you guys are gonna notice that
there's one more thing we gotta do
so i'm gonna hit that
and it's gonna oh awesome no it's good
to go
so i have some custom images in here
guys if you guys want to be able to get
that
you just close out or go to safari
and you can do any sort of like google
search for images
so just go to you know on the safari app

Second subtitle

Main subtitle

just go to images

you can search for anything literally

like let's search for like goku

and you know like you can just literally

click on any image

and like long press on it and then say

add to photos

and you can go back here to your image

picker and you guys will notice that it

shows up right away so that's pretty

cool but anyway we can click on that

image now and you guys notice that it

dismisses the image picker

and so it you know it actually selected

a photo for us but we haven't actually

done anything with that photo just yet

so what we're going to want to do is

actually replace this guy

with the image that we selected

so

um we're going to do that in this in

this label right here right because we

want it to be a button no matter what

because like let's say the user selects

an image and they want to change it we

still want them to be able to click on

it to change it again so the only thing

that's going to change is this label

so

for the image

let's see what we got to do here guys

so let's just go ahead and cut this with

command x

and we're going to say if let profile

image equals profile image

and

else we put the image there and if not

Second subtitle

Main subtitle

we put profile image

and we still do all this stuff right

and we're actually going to want a clip
shape of a circle

just like that

all right

but

um we noticed that this is not great

right to have all this code listed twice

um and we can actually create like a

custom modifier

to handle this so

um let's see if we can

do that now

so down at the bottom here guys let's

just create like a private struct

uh and we call it profile image

modifier

view modifier

and it's gonna have this funk

called body content

right so we'll be able to take some

content which is essentially gonna be

this image whatever this image is and

add this modifier to it

so we're gonna just say content and then

we can take all this code

and modify it to that content

and we'll just be able to go here and

say

dot modifier

profile

image modifier

and

just take this line of code and do the

same thing there so that's like

significantly cleaner than having all

Second subtitle

Main subtitle

that code sort of jumbled up in there we just need to create this private structure and it's like you know private to this file

ooh

um

i think we need to initialize it like that and that should be good to go all right it's yelling at me a little bit

um so let's see guys

um looks like we can't have resizable in there which should be fine

um

rendering mode template we only need that on this part right here and padding top we can actually just put that on the button itself

just like that

so we're actually going to have to add a little helper function so that when the image picker dismisses it replaces this uh button this

image we have here with the selected image from the user

so let me clean this up a little bit outside of this body guys or uh we can actually just go ahead and add a function called load image and we're just going to say guard let selected even image equal selected image else return

um

i like my guard statements like this then we're just going to say profile image equals image

Second subtitle

Main subtitle
and then type in ui

and pass in that selected image
so this is really really cool
this allows us to construct a swift ui
image from a ui kit image so this
selected image is the image we get back
from the user selection right that's why
we named it selected image
but in order for us to get this to work
we can't use a ui image we have to use a
swift ui image which is just the regular
image guide that we see here
so we need to be able to construct that
from
a ui image which is what we're allowed
to do here
so now we just have to figure out where
to call this function
so we can go back to this sheet guys and
we can say comma on dismiss
load image
so your function should now look like
this we had this option when we created
this sheet but basically
this is really really cool
and what it's going to do is when that
sheet dismisses it's going to call that
load image function and load our profile
image into that spot
so let's run the application guys and
see what happens when we select a
profile photo
so i'm going to go here
i'm going to select goku
and so that's working which is awesome
but
the uh

Second subtitle

Main subtitle

image is way too zoomed in right like
that's not what we want so we're just
gonna go up to our profile image guy and
we're going to go here and say dot
resizable and that will allow the image
to be resizable and actually fit the
frame we don't need to do that here
because we have the template rendering
mode and this does not need to be
resizable

so if we go ahead and run that again
guys

we should notice that this is working
really nicely and then the last piece of
this puzzle

is to add a little bit more to the view
okay that looks freaking awesome right
guys this is really really coming
together it's exciting right we're
building it piece by piece and we can
see it just like slowly morphing into an
awesome application i'm loving this man
and i've already built this before
you guys must be having a great time but
anyway um the last piece of the puzzle
guys

is
or pieces rather we got two more things
to do with this image stuff

we have to add a button that says like
continue

once the user selects a profile photo
right so once they select it we want
them to be able to see what they've
selected make sure it looks good and
make sure they don't want to change it
and we notice that hey we can still

Second subtitle

Main subtitle

change it if we want to

and then this button is going to show up
they'll be able to hit continue and then
continue on to the main interface of the
application

so let's go ahead and implement this
button really quickly so we're going to
go back here and add to the v stack
so i'm just going to say if let profile
image equals profile image

then i'm going to want to grab a button
and it's going to look really similar to
our authentication button

so we can literally just go ahead
grab this code

go back here and put that in there
and it'll say continue

but we don't need this let's just go
ahead and say print
debug finish registering
user

and we actually don't need access to
that profile image property guys so we
could just go ahead here and say if
profile image is not equal to nil

let me fix that not equal to nil and
remove the let

so let's go here
select a guy

and we notice that that continue button
shows up and it looks really good and
it's giving us that print statement back
of finish registering user god damn this
is looking good

right

so

i mean guys this is just like super

Second subtitle

Main subtitle

super smooth man
so anyway now all we got to do
is upload this profile image that's the
final piece of the puzzle right so we
actually have to write some code
that is going to upload this profile
image to firebase storage
and then we are going to have to modify
our uh user
uh data in the database so let me bring
this over really quickly
uh sorry guys that was a you know just
got to bring this uh
stuff over from my other screen
so we need to add like a profile image
url
property right so it would look
something like this it would say like
profile image url and it's gonna be like
uh
dot image is
here dot firebase storage
dot com right add so like
this is like basically what it's going
to look like but the www dot blah blah
blah is going to be the actual location
of the image and when we go to that url
it's going to download the image to our
device and then we'll be able to access
the profile image so that's what we're
going to do next get excited for that
guys
and that will be the final piece of our
authentication
functionality then we can actually get
into
loading some of the user data that we've

Second subtitle

Main subtitle

uploaded into like user profiles and then we can get into uploading tweets and searching for users and all the fun stuff you guys have been waiting for that are going to really bring this app together so let's keep going so we are going to hop into this service class or folder sorry guys we're gonna create a new file it's gonna be a swift file we're gonna call it image uploader so what this is going to serve as is essentially some sort of helper class that its sole purpose is going to be uploading images so if you could imagine if you want to be able to upload images in multiple places in your application you'll be able to just reuse this image uploader helper instead of having the same code in multiple places hey i want to be able to upload a profile image or hey i also want to be able to upload a tweet image right now having it in one centralized location allows us to reuse it which is what we want so we are going to go up here and import firebase and we're also going to import ui kit and we're going to create a struct called image uploader and it's going to have one static function called upload image

Second subtitle

Main subtitle

it's going to be an image which is a ui
image

and it's going to have a completion of
at escaping parentheses string
void just like that

so

what's the reasoning behind this
completion that we see here so
essentially guys

what we want from this function is to
for it to give us give us back an image
url so we're going to upload an image to
firebase storage

but we need to get back the location of
that image so that we can put it in our
data dictionary here right we ultimately
need the location of that image
so that when we want to like download
the user data we get the location of
their profile image

because if you remember what i said we
can't actually store the image here
we can only store the image in the
storage bucket

and then get a pointer to that image
right it'll make more sense once we
actually go through it if that doesn't
make sense already

so let's go ahead and get to it we're
going to say guard let image data equal
image.jpg data compression quality
0.5 else return

so this grabs the image gives us back
the image data and it compresses it to
half the original size

okay so

it's going to slightly lower the quality

Second subtitle

Main subtitle

of the image but it's going to reduce the image size so that when we download these images it's not going to be as large of a download which will ultimately make our app faster and limit our data usage so you could even limit this to like something like 0.3 but let's just keep it at 0.5 so we don't lose too much image quality we're going to create a file name it's going to be nsuuid constructor.uuid string then we're going to create a reference we're going to say storage dot storage dot reference with path parentheses slash profile images or let's make it underscore images slash we use some string interpolation file name so this is the path to the image we're creating you guys are going to see a folder show up here that says profile images and then within that profile image folder we're going to have some unique file name this is just swift's way of generating some unique string okay then we're going to say ref.put data actually we need a different one put data with a completion now here we pass in the image data metadata is nil completion you guys notice that we get back a callback of a potential error or some storage metadata so you go here

Second subtitle

Main subtitle

underscore

error

we're going to handle our error we're

going to say if let error

equals error

then we're going to say print debug

failed to upload

image with error

error dot localized description

and then return so if we have an error

uploading the image it's going to give

us that error message and it's going to

return

now we need to get the actual image url

so we can execute our completion handler

so we're going to say ref.download url

completion and this gives us back a url

or an error so you hit enter on that guy

say image url

and then we're not going to handle the

error here guys because

honestly if something goes wrong it's

probably going to be here we don't

really have to worry about it here if

you guys want to be super super safe you

could just

do that for the download url as well

but we're going to now say guard let

image url

equal

image url dot absolute string

alex return

and then we're going to execute our

completion of image url

alright guys so this is

going to be our image uploader helper

now we actually have to go back to the

Second subtitle

Main subtitle
auth view model**Second subtitle**

and be able to upload that profile image
so
we're going to create a helper function
to do this for us
so i'm going to go down here and say
private
func
upload profile image
or actually it's not going to be private
we need to actually access this through
the profile photo selector view
and we're going to just do underscore
image ui image
just like that
we're gonna say guard let uid equal
um
okay so this actually gets a little
tricky
um because
we obviously want to be able to say user
session dot uid else return
but if you guys remember
in our register function right we can't
set this user session to have a value
until
we want to be taken to our main user
interface
right so we actually need to create
another property
so i'm going to go up to the top
and i'm just going to create a private
variable
called temp
user session
which is firebaseauth.user
just like that and this is just going to

Main subtitle

be like a temporary user session that we can set in this register function

so

this is like the actual user session and once we set this

it's going to take us over to the main interface

but we have this in between step now right

where we want to upload a profile image so we're going to go here and say guard that uid equal temp user session dot uid and up here

we can uh delete this guy we can actually delete this too we don't need that anymore

after this guard let user equal result.user statement we're going to say temp user session

equals user

hit command b just to make sure everything is okay

we should get a

we need a self here

so now we can store this user session in some like temporary property that's not going to take us to the user the main user interface right

so now we want to upload this profile image we actually do need access to this uid

because why

we need to be able to go into our firestore database and update the correct

user structure so we need to go to the users collection

Second subtitle

Main subtitle

and then go into the correct user id
which is going to come from this temp
user session guy
once the user gets created right
we're going to need to know
where to upload their profile image url
right we can't just go into some random
users data structure and put a different
user's profile image that would be all
kinds of bad

so

um that's the reasoning behind this guys
and then we're just going to go here
and we're going to call upon our image
uploader

we're going to say upload image

we're going to pass in the image and

then we're going to say here we have the
profile image

url and now here's where we can upload
the information so we're going to say
firebase.firebaseio.collection

users

dot document

and this is where this comes in

uid

so this takes us to users and then goes
to the correct user id and now we can
upload their profile image url

document uid

uh

dot update data

with a completion

of um

let's see we need to say profile
image url

and then pass in that profile image url

Second subtitle

Main subtitle**Second subtitle**

completion we don't need to worry about
the error
now we can say self.user session
equals
self.temp current temp user session
right so
guys if this isn't smooth i
don't know what is right this is smooth
like suave super suave ricky suave
smooth bruh
right
so what this is going to do is
upload our profile image and then it's
going to go into our database update the
user structure and then once that
process completes
then we set the temp users or the user
session
to the temp user session and then it's
going to take us over to our main user
interface right so that's a lot of
going on
but
it's uh
that's about as smooth as it gets to me
man if you guys got a better way of
doing this you can let me know
but anyway
we need to actually call this function
in our profile photo selector view
so here we have finish registering user
and we're actually going to need to
get that uh
if let going equals profile image
now we say view model
oh we need access to our view model
so we need an environment variable

Main subtitle
our view model**Second subtitle**

off view model so now we can go down here and say view model dot upload profile image profile image and that's actually wrong right because we need to be able to upload a ui image so if we go back to this image uploader um the only way we can convert this image to actual image data is through this ui image guys if you tried to replace this as a regular image like for the for from a swift ui thing or using swift ui the swift ui image component you're going to notice that it doesn't have that jpeg data transformer so we cannot do that we need to pass in the ui image but it's a good thing we have one right this selected image guys a ui image so we're actually going to say if that selected image equals selected image and then pass that selected image right there now let's go ahead and run this oh we got build failed huh it's asking me for a view model that's weird at environment object var view model so that should be good to go

Main subtitle

and

let's see um we actually need to go back
to our app file guys

and

delete this and go back to our original
content view

run the app and test this the whole way
through and see if it works

and i'm confident it's going to i don't
think this is going to mess up i think
i'm uh

i think this is going to work so let's
go ahead and log out

hit sign up

just do like test

9876 at gmail.com

test9876 as the username

test9876 password qqq

sign up

awesome

so

first of all

we should notice that we have a new user
here

so we see that we already have this user
existing in the database right

that

uh transition from this that's previous
screen to this screen actually involved
authenticating the user and uploading
their information the last piece of this
puzzle is uploading their profile photo
so now we can hit plus photo

let's do a ferrari
and we hit continue
and bam look at that bro let's go
that is the actual

Second subtitle

Main subtitle

profile image url's location so if you guys were to copy and paste this right let's copy and paste this url you paste it and look it downloaded that image and now i can click on it in my downloads and look it downloaded that image to my computer how slick is that right so that's what that whole process involved guys we store the image in firebase storage look at that right there boom profile image folder showing up here's the image that we just uploaded to firebase and we see here that this is the actual location of that image and what we see in firestore is a pointer to the location of that image and when we access that url it downloads the image to our device so this is looking absolutely amazing okay and we were in addition taken back to the main interface of our application once we hit that continue button right so that's a lot of moving parts that we just brought all together to magically work in unison okay so that's a full like authentication sign up flow where you uh have like a separation of like

Second subtitle

Main subtitle

concerns where you have the user fill out a form and then upload a profile photo and you guys could take it a step further and make it so that it's optional to upload that profile photo but we're not going to be doing all that because we want to get into the actual fun stuff of being able to like upload a tweet

and replace all these boring blue images right with actual profile images from the application so that's the next step guys this is looking really really good um

in the next one our chapter we're going to continue with fetching

user data and displaying all the actual user data in their profile and then we'll be able to search the app for all the existing users that we have and ultimately be able to click on their profile and have it display real life information so get excited for that guys the app is really really coming together and i'm super pumped about it hope you guys are too

thanks for sticking with me up until this point i think we're like four or four and a half hours in right now and it's been a lot of fun i'm glad you guys are along with me on the ride on this one let's keep going

all right guys so essentially we've just uploaded a bunch of data to the database we have it saved somewhere on the internet but what we ultimately want to

Second subtitle

Main subtitle.

be able to do is download this information and display it in our application right so we can see a bunch of different users showing up and download the information from our central data storage or api so we're actually going to go and create a service class and it's going to be this file is going to be called user service and it's going to be dedicated to fetching and downloading user information from the api so we're going to go to this service

folder we're going to create a new swift file and we're going to call it user service

user

service cool

we're going to import firebase up at the top

and i'm going to create a structure called user service

and this is going to have a function for now called fetch user with uid uid string

and we're just going to add a print statement here that says print debug fetch user info

so i want us to figure out where exactly we're going to call this and how we're going to utilize this user service

before we actually go over how to fetch our user information

so we're going to hop back into our off view model okay guys

and we're going to go up at the top and

Second subtitle

Main subtitle

we're going to say private let service
equal user service
and in our initialization function
we're going to go ahead and just delete
this print statement
and we're going to say
service
dot fetch user with uid
and we need to pass in a user identifier
here
so
um
actually guys what we can do
is in this init
or sorry we're actually going to create
a
helper function here
that's called fetch user
and we're going to say guard let
uid equal
self.user session dot uid else return
and then within this function we're
going to say service dot fetch user with
uid uid
and then up at the top here we can say
self.fetch user
okay so this fetch user function
is a function that's within our auth
view model
and this guy calls upon the service
fetch user function which we have to
pass in the uid to
and in this case it's just going to be
the current user id
all right so let's just go ahead and run
our application and see if we get back
that print statement that we wrote in

Second subtitle

Main subtitle

our user service file fetch user info

that's all we want to see here

right this way we know that we are
calling the function in the correct

place

and it's actually giving us back some
data signifying to us that it's being
called

so

bam we get fetch user info so now we
know we're good to go so this is where
we can actually fetch the current user
information

so

we're going to go ahead and say

dot firestore.firebaseio.com

users

dot document

uid

dot get document

completion we're gonna have a snapshot

oops snap shot

underscore and then under here we're

going to say guard let

uh

what's it called snapshot

equal snapshot

else return

and i think we can actually say snapshot

dot data

right

so we can say guard let data equal

snapshot.data and let's just go ahead

and say print debug user data is

data

and let's see if we're actually getting
something back from this fetch user

Second subtitle

Main subtitle

function

right so this ideally should give us back the data of the user that we are currently logged in as

so this is going to be pretty exciting guys big moment of truth here

oh

look at that baby

right we're getting the full name we're logged in as test9876

email bam and we also have the profile image url and the username so this is us getting information back from our api about the user that we are currently logged in as this is absolutely amazing guys

this really coming full circle here where we've uploaded information to the database we are now fetching that information from the database and we just have a couple more steps to go to actually be able to display this information in our app and have it really come to life okay so the first thing we're gonna have to do is build out a data model okay we have to be able to map this data into an actual object okay

so we're gonna hop up to this model folder here and we're going to create an object so we're going to just call this user

and up at the top we are going to import firebase firestore swift and i'll explain why in just a little bit but what we're going to do guys is create this struct it's going to be a user and

Second subtitle

Main subtitle

it's going to conform to this identifiable protocol and i'll explain this as well but essentially we're going to create something called a document id var id string and it's going to be optional then we're just going to create a couple other parameters yet let user name string let full name string let profile image url string let's uh what's it called email string all right so this is going to be our user object if you guys don't understand the concept of an object in programming i highly recommend you go and look up some object oriented programming fundamentals i will have a youtube series on that coming very very shortly which i will link to the description when it's ready so at the time you're watching this video it might be ready so just check the link to the description if you guys don't have any idea what i'm doing when i'm creating something called a user object but i'm assuming most of you know what is going on here but essentially what we're doing guys is mirroring um or creating an object in our code that we're going to map this data into so we can access it all through this object so we can say like user dot username or full name or profile image url right so let's talk

Second subtitle

Main subtitle

about what we're importing up at the top
here it's firebase firestore swift so
you guys are probably familiar familiar
with the decodable protocol this is just
what we have to import to work with the
decodable protocol through the firebase
code okay so we'll be able to just
decode our user object um through use of
importing firebase firestore swift here
so this is our data model guys this is
looking pretty good
so what i want us to do is hop back into
this user service and actually map this
data into our user object okay
so let's go ahead and try to do that now
so i'm actually going to go here and
delete this uid field we don't need this
guys
and that ties into why we created this
document id variable so basically what
it's going to do
is
it's going to read the document id here
and store that in the id property so
that we don't have to duplicate that
data in the actual body of the object
okay so it's going to say hey the user
this is the document id
that represents all this data here it's
going to grab that and store it in this
id property unfortunately it has to be
optional
but that's not too bad to work around so
let's just go ahead and hop back into
this user service and see if we can map
this data into that
so up at the top guys let's go ahead and

Second subtitle

Main subtitle
import**Second subtitle**

firebase firestore swift because we need
to do this to actually be able to decode
this data

um

let's actually just go ahead and delete

that dot data and

rename this property snapshot

and we're going to say

guard let user equal try

um snapshot.data

and you guys notice that when you say

data as it has this throws option or

like that means it can throw an error

that's why we have to mark it with this

optional try

but we don't have to introduce the catch

if we in if we have a question mark next

to the try and wrap it in a guard

statement so we're just going to go here

and say user dot self

else return

and let's just go ahead here and say

like print debug

user name is

user dot username

print

debug email is user.email okay so you

guys notice through use of this object

we can say like

um

what is this yelling at me for

requires that user conform to decodable

i guess we actually have to conform to

the decodable protocol that would make

sense

so go back to your user struct guys and

Main subtitle

conform to the decodable protocol
let's try to build that one more time
and that should be good to go looking
good build succeed please build succeed
please let's go
all right so let's run this and see if
we get these two print statements back
you guys notice the beauty of using an
object is that we can we get this dot
notation and can access all the
properties associated with that user and
that is absolutely beautiful
right guys we get username is test9876
email is test9876 gmail.com so
couple things are happening here we are
successfully fetching information from
the database
and then we are successfully mapping
that information into our user object
and then
we can access that user object and
access certain properties associated
with that user object through use of
this dot notation which is a thing of
beauty okay
so
now what we want to do
is actually
um link this user service stuff
to our
auth view model okay and we're gonna do
that through the use of something called
combine
okay so
combine is something that works very
very well with swift ui
and we're gonna see how exactly we can

Second subtitle

Main subtitle

use that here guys to ultimately be able
to
display
information about our user in things
like our side menu and our profile and
all that good stuff
but before we do that um you guys notice
that i did manually modify some data in
my database and you got to be really
careful with that because
i deleted that uid property because if
it was still there i'm pretty sure the
decode would fail right it might not but
um essentially what this decodable
protocol is doing is it reads a data
dictionary and it tries to look for
the exact name of these keys or these
values or these property names in the
data dictionary so you guys will notice
that these things match up identically
right username profile imagery full name
email all match up directly with what we
have here if i were to simply change
this to like
full uh
you know capital like that
you guys are noticed that this decode is
going to fail
and i'm actually not going to get those
print statements back about my user so
that's the power of the decodable
protocol guys i love it
and it makes life a lot easier when
you're working with objects and
downloading information from an api so
you guys notice there's no information
about there's no print statements coming

Second subtitle

Main subtitle

back that's because the decodable protocol or the decode failed there you guys notice if i change it back so full name matches full name here and all the other fields match that i'm going to get my print statement back in the console there meaning the decode was successful okay so that's uh something that is really important to understand guys a lot of errors that i get messages about come down to you guys spelling things wrong in that in that process so just watch out for that

um

but essentially now what we want to do like i said before is link that user service to this off view model so that we can pass that user object around to different views right so let's go over how we're going to do that alright guys so we're going to hop back into our auth view model and we're going to

start off by adding a published property a published var current user which is going to be the uh of the type the user object that we created and it's going to be optional the reason it's optional is because we have to reach out to our api to fetch this user data before we can set it so when the app launches initially it will be nil even if it takes a quarter of a second to fetch that information the app launches before the current we can fetch that data right it always takes time to

Second subtitle

Main subtitle

execute an api call so we have to
initialize this as an optional property

so

um basically we need a way of getting
the user

from this fetch user function back into
this file so that we can set this
property here

so what we're going to do is go back to
the user service and we're going to go

here and add a completion

it's going to be at escaping

parentheses user

void just like that

and we are going to go here and just
after this guard statement we're going
to say completion

user

and you guys are going to notice that if
we build the project

we're going to get an error

and that's because in this completion

or in this function call here we're

missing a completion block

so let's just go ahead and delete that

fetch user guy let's just retype the

function we're going to pass in that uid

and now you guys have noticed that we're

going to have access to a completion

handler so if we hit enter on this it's

like oh hey

when this function call completes you're

gonna get access to a user

and then we just have to go here and say

self dot current user equals user

okay

and that's gonna set this published

Second subtitle

Main subtitle

current user property to
the user that we fetch from the database
all right so that's pretty cool
now what we want to do uh first thing i
want to do is actually see this user's
information showing up in the side menu
okay
so we're gonna go back to the side menu
collapse all that stuff side menu side
menu view
and we already have access to this off
view model right here right
and through this off view model
we have access to our currently logged
in user and it's a published property so
we know when that guy updates it will
update our
um
side menu view with the new and incoming
data
so
what we're going to do
is just wrap
some of this stuff in an if let
statement okay
so we can basically wrap this whole
thing
right this is one way of doing it we can
just say if let user equals off
viewmodel.currentuser
and paste all that stuff in there
and now we can replace this dummy data
with actual data okay so here i'm going
to go ahead and say user.full name
here i'm going to do some string
interpolation i'm going to say at
user.username

Second subtitle

Main subtitle

and that's all we have for now we're also going to want to get this profile image stuff in there but that's what we're going to do next for now i just want to see if this is actually showing up with real life user data this is going to be really freaking cool right a lot of stuff's about to come together here guys

uh so drum roll please

right let's see if it's working

so we go here

and look at that

i love it when this works man it

makes me feel good about myself right so we see test9876

test9876 as the username so that means

guys that we are successfully fetching our user information from the database

we've created our user object and now we are passing it to

different view components in our app the

first thing we passed it to was our side

menu

all right so

now that that's happening

um

i want us to go ahead and knock out

this uh profile image stuff here so

we're actually going to download a

third-party library or swift package

that is going to help us download images

from the internet and display them in

our app in the most efficient way

possible

so let's go ahead and do that now

so uh we're going to go to file or sorry

Second subtitle

Main subtitle

let's highlight xcode

file add packages

and let's go ahead and find the package

we're going to add guys just go to your
internet browser and type in kingfisher

swift package

kingfisher is one word and you're going
to see this link that says 1vcat

kingfisher

right so this is just a third-party
library that makes downloading images
really really really really simple

i absolutely love this uh

this uh

library

so

we just need to find the install

instructions so let's see

swift

package

so we have package manager so if you
guys just do a find command f and search
for swift package you'll get the url
that we need to download this swift
package

you go here you paste it in there

click add package

and just let that do its thing

all right guys so you should see this
screen after a bit you just hit add
package

it's only one box that you can check and
now we've added the kingfisher package
to our project and let's just go ahead
and see

just how easy it is to use kingfisher to
download an image from the internet

Second subtitle

Main subtitle

so we're going to go back to the side

menu view

we're going to import king fisher

just like so

build your project with command b just

to make sure everything is working

okay no such module kingfisher that's

unfortunate because it's literally right

there

um guys we don't need our preview

anymore so i'm just going to close that

out um typically this error will go away

once the the project actually finishes

building it just takes xcode some time

to recognize that but anyway um you

notice guys here that we just have a

circle um so we're gonna replace this

with something called kf image

and it's gonna ask us for this url okay

um so now we actually have to pass in a

url object

so we're going to say url and we're

going to use the string constructor and

we're going to say user.profile image

url so that's literally all we have to

do then you just go here and add dot

resizable and then like dot scaled to

fill

and uh

i think we have to say dot clip shape

circle or something like that right

because before we were working with the

circle now we're working with an image

we actually have to give it a clip shape

of a circle

so

um the guys the reason we have to do

Second subtitle

Main subtitle

this conversion of like building this url here is because on our user object we just have the profile image url represented as a string the url object is a custom swift object so

we can't say it's like it makes your life more difficult if you try to cast this as a url it's easier to just cast it as a string because that's what it comes back as from your database or your api so

then we just go here and construct that url using the user profile image url string

and that's literally all we have to do so i'm going to go ahead and run this now

and

we should see
in our side menu
right

that user profile image url showing up
and it's going to look super freaking
smooth

and we're going to see

oh

check that out boy

let's go

we see our profile image url showing up
beautifully there guys
and

in addition to this test data
so this is looking really really good so far
um the next step i want us to go over

Second subtitle

Main subtitle

how to do is implementing that up here as well and then we're going to go over how to implement that in the profile as well

so

next step guys is we need to hop back

into this content view here

so that we can replace this circle

image that we have as our

leading navigation bar button item with the actual user's profile image url so if you guys go back to the content view and you scroll down to the main interface view we have in the extension we have this button here that's a part of our toolbar

and right now the label is just this circle

so what we're going to do is replace that with our profile image url

so

um on this toolbar guys we can go here and we can just say if let user equals.viewmodel.currentuser

and so basically this won't get created unless we actually fetch our user then we're going to replace the label with

first we got to go up to the top and import

kingfisher and you guys will notice that we have access to the auth view model through this environment object that's why we can go say if let user equals.viewmodel.currentuser and then here we're going to just say kf image

Second subtitle

Main subtitle

url

string

user dot profile image url

and then we're going to say dot

resizable dot scale to fill

dot frame

is 32 by 32

and then we're going to say dot clip

shape of a circle

all right so let's go ahead and run this

now guys

and we should be able to see

our side menu also showing up with our

user profile or sorry our toolbar

showing up with our user profiles image

url or profile image that we are

downloading

and

that is exactly what twitter does and

you guys notice that it takes just a

second to show up right but then

as soon as it does boom we have that

stuff loading absolutely beautifully so

that's uh the next piece of the puzzle

that we have conquered guys the next

step that i want us to go over how to do

is how to integrate this uh

user object into our profile and then we

are going to segue into actually

fetching multiple users

and displaying them in this user list

here and we're going to fix some bugs as

well you guys notice it looks like we

were doubled up on

some navigation views here we're going

to fix that stuff

but this is looking really good guys so

Second subtitle

Main subtitle

let's go ahead and figure out how we're going to add or integrate our user into our profile here that's going to be fun so we are actually going to take a different approach on how to sort of get the user object into this profile view and ultimately display their information so we're going to use something here called a dependency injection or a custom initializer so we're going to initialize this view with a user and then up at the top we're just going to create a user property we're going to say private let user is a user and then we're going to go here and just say self.user equals user and if you guys try to build that you notice that you're going to get an error and that's because it says hey you need to initialize this profile view with a user

so

um you guys notice in our preview now we're missing a parameter for the user so if we want we could go ahead and just sort of add like a dummy user and it's going to ask us for a bunch of this stuff uid or id

you know and we're just going to say like

nsuid uuid string username is like batman full name is bruce wayne profile image url is just blank and email is batman gmail.com

Second subtitle

Main subtitle

all right so this is like i said guys
sort of like a dummy user
that we can add to our preview just to
make sure that that still works
um we don't really need the preview
anymore

but you guys notice the uh you know this
comes back to this initializer that we
just created um we tried to create the
profile view but it was like hey you
need to initialize this thing with a
user

so we're going to try to do it one more
time

and i'm pretty sure we're going to get
an error again

and that's because we're creating this
profile view

i think in two places in the app right
when we click on
this profile button in the side menu
and also when we go to the explore page
and click on a user from there

so

let's just go to those two places and
see if we can figure out how to figure
fix this problem so we're going to go to
side menu first we can go to side menu
view and here is where we go to the
profile view with this navigation link
so we're just going to go here and say
user

user right because we've wrapped this
whole thing in an if-let statement and
we're getting
access to the current user that's on the
view model so now we can just pass it

Second subtitle

Main subtitle

into the profile view

um alternatively you guys could add the auth view model back to the profile view as an environment object and do the same thing we did here and i don't think there's anything wrong with that right i think uh both ways are completely acceptable here so i'm just showing you a different way of doing it you guys can pick which one you like better or which sort of technique fits your app better but anyway we got one more place to do this in so let's go to the explore folder go to views explorer view and then here we need to pass in a user as well you guys notice however that we don't have access to a user here so i'm actually just going to comment this out for now until we actually go over how to fetch real users from this we could make some dummy data but it would kind of just be a waste of time because we're just going to end up fetching that stuff from the database anyway so now i could go back to my profile view and i could now take this user object and start replacing all that dummy data with actual data about the user so let's go ahead and do that so first thing we're going to do up at the top is import kingfisher and in the header view we want to replace this circle with an actual uh

Second subtitle

Main subtitle

profile image so we're going to replace

circle with kf image

url

string

user.profile image url

and then we're going to say dot

resizable

dot scaled to fill

dot clip shape

circle

and then i guess it also has that offset

all right so that's the first thing guys

excuse me

um

now we just keep scrolling down and

anywhere we see some dummy data like

this is uh you know dummy data right

here where we have heath ledger we're

just going to replace this with

user.full name

and then here we go to the at symbol and

we do some string interpolation and we

say user.username

uh we don't have a bio for the user yet

so we're just going to leave this there

we also don't have location and website

yet so we're going to leave that stuff

there as well

and then we don't have

follower data yet either so all that

stuff is going to remain the same for

now

let's just go ahead and run this really

quickly let's see what my problem is

cannot find circle

oops i spelled circle wrong

so we're going to run this

Second subtitle

Main subtitle

and see if we're getting some actual
data
in our
user profile

and once again guys this has been done
through something called a dependency
injection where we're not using the view
model we're actually just requiring that
we initialize this view with a user
just another technique of how to get
data from one place to another so we go
here

click on profile and look at that guys
that's looking super smooth right we got
our profile image here we got our test
uh you know username and full name
um

and what i'm going to do guys is
i'm actually going to go ahead and
delete all these dummy users and create
a bunch of actual like users so the app
looks a little cleaner but we'll do that
in a second um a couple things i want to
do

we need to adjust the spacing of these
two things like we want this arrow to be
a little bit closer to the top here this
looks a little messy
so let's actually scroll down to
our header view and we're going to
change this y offset to negative 16 okay
guys
um

positive offsets move it down negative
offsets move it up when you're on the
y-axis and that's because
the origin point of the

Second subtitle

Main subtitle

coordinate system for a phone starts in the upper left so anything positive moves it down anything negative moves it up

so it's a little weird and you guys notice that looks a lot better

um you guys could move it up even a little bit more if you want like we could make this like negative 20.

and i'm sure that would look great you guys notice we click it and everything is still working just fine

so that's looking really good guys uh we have our user injected into our profile now and next up um like i said i'm gonna do this off camera but i'm gonna just go ahead and delete

all of those uh sort of custom test users that we made like all this stuff here

um you guys make sure that if you do the same if you have just a bunch of dummy users and you want to delete them as well that you delete them in the authentication guide as well so you just go and you click on this and you just click delete account

but let's make sure we sign out before we do that

so i've signed out and then i'm just going to go ahead and delete all these accounts

one by one and then you go and also you can just delete this whole user collection you just say delete collection you type in the name of the

Second subtitle

Main subtitle

collection you hit delete and then you go to storage and you can delete that as well so that's sort of just how you wipe your database clean if you ever need to do that um and i'm going to go ahead and register some actual users so the app just looks a little cleaner and then what we're going to do is go over how to fetch all the users that we have on the application and display them in that list be able to click on them and take and then navigate to the profile and also be able to search for users which is going to be a lot of fun guys so get excited for that

okay guys so i've created a couple more users um i went with f1 drivers for fun so we can see here that this is looking pretty good i did however notice a couple bugs that we have in the application um first off we can't see it because of the color

but the if you guys hold down right by where that arrow is there's actually still a navigation uh button there so on the profile view we're gonna go here and say

dot navigation oops not there but on the v stack rather and say dot navigation bar hidden is true

and let's go ahead and run this and see how that looks and the other bugs were authentication bugs and i'm going to go through the process of signing up a user if you guys did the same thing i'm sure that you notice these bugs as well um and we're going to go over how to fix

Second subtitle

Main subtitle

them now

so let's let the app launch
and let's just go back and check out the
profile

so now i'm logged in as danny ricardo go
to my profile and you guys notice that
now the offset of the button is like a
little jacked up so the sizing of this
was messed up because the navigation bar
was actually still there so we need to
go back to the header and adjust the
y offset so let's just do like negative
four

and run that one more time
and hopefully that looks better
because it was definitely kind of
confusing as to why the frame was only
96 pixels but it looked so much bigger
right now everything actually looks
as it should

so we go to the profile and that looks
pretty good there guys right and that is
working really smoothly

awesome okay so the next bug i'm just
gonna go ahead and show you guys what it
is and then we'll uh go over how to fix

it so if i were to log out of a profile
and then go and sign a user up so i'm
gonna sign up a new user

um let's see

who do i want to do

uh let's do lando norris

so just do lando at gmail.com

uh lando norris

full name is lando norris

one two three four five six so you guys
remember that i was logged in as danny

Second subtitle

Main subtitle

ricardo okay so that's just let's call that user a and this is user b so i hit sign up that's working just fine go to my profile photos and i select lando norris there i hit continue and it's going to take me back to the main interface of the application but you guys notice that it still looks like i'm logged in as danny ricardo like i go to my profile and everything is from the previously logged in user that's not what we want so what we need to do is go to our authentication view model and on this register function guys after we authenticate the user we need to call this fetch user function again okay and what that's going to do is it's going to fetch the user with the new data so this user session is going to have a different value okay and then once that has a different value or actually we might want to call it in this upload profile image yeah that's what we want to do so after we upload the profile image then we're just going to go here and say self.fetch user right so now you can see make sure you call it after you set the user session to the new temporary user session right because now this will be equal to the newly created user's id and then we go to this fetch user function and it says oh hey we have a new user i'm going

Second subtitle

Main subtitle

to go ahead and fetch that information

now

sweet

all right so

um now we also need to do this in one more place we need to do it in the login function as well so self.user session is user and then we go here and just say self.fetch user so that means now that once the user logs in it will be logged in with the new user data

so that looks pretty good let's just go ahead and run this and

you guys will notice that once i run the app then it's going to be showing me with the lando norris profile

and i'm just going to create one more user just so we can verify that this is working

so now you guys will notice that it's lando norris up at the top that's looking good we can see that it's all lando norris here this button looks good up here so now let's go ahead and log out

create a new user

so it's going to be valtteri

gmail.com this is val teri bottas

oops

val terry

botos cool one two three four five six i

hit sign up

i add a photo

i select my valtteri botox photo i hit

continue

and you guys will notice that now it's valtteri botasa's information in the

Second subtitle

Main subtitle

application and it's not the previously logged in user so that's looking really really good we go to the profile everything is fine and dandy right so this is looking awesome guys um the next step for us is going to be fetching all of the users and displaying them in this explore view and also implementing some search functionality and being able to go click on one of these users and have it navigate us over to that

user profile so that's going to be a lot of fun and that's what's up next now that we have fixed those bugs we also have one more bug to fix here where we look like we're doubled up on some navigation bars so that's what we're going to be doing in the next chapter guys get excited

all right guys so the first thing i want us to fix is this issue we see here where we have this double uh navigation view right so

this is actually a really easy fix we just need to hop into the explorer view and remove this navigation view here so let's just highlight or all this stuff with that's the v stack so pretty much everything except this navigation view and its brackets cut it delete the navigation view and paste that back in there

so let's go ahead and run this and let's go to the explore page and you guys will be able to see that that uh second navigation item is gone um there is

Second subtitle

Main subtitle

another bug that we are gonna fix uh later on guys you guys notice that the tab title or sorry the navigation title is the same for each one of our tabs and we're going to be going over how to fix that later on you notice that even though i have navigation title explorer here it uh doesn't change we're going to fix that bug like i said a little bit later in the course i want us to get more into the fun stuff here where we're gonna actually get started with how to fetch all the users in our application and also search for them and filter through them so we can check out different people's profiles that's going to be really cool and really bring sort of the social media aspect of this app together so

let's go and hop into um i think it's going to be our user service okay guys so right now we have a function for fetching one user we need a function that's going to help us fetch all the users so we're going to write a function here called fetch users and it's going to have a completion of at escaping void and we're going to get back an array of users just like that and all we need to do is go into this collection firestore.firestore.collection users and then just say dot get documents

Second subtitle

Main subtitle

and that's just gonna give us back all the users in that users collection which is uh really simple so now we just have to say snapshot and underscore for the error because i don't really bother handling the error guys with snapshot fetches um because it pretty much never fails unless you just don't have an internet connection um which is something you can handle separately but anyway let's say guard let documents equal snapshot dot documents else return so just to make this clear guys what we're doing in the database is we're going to this user's collection and we're just getting all of these documents and each one of them will have a different sort of data packet associated with it like different values for all of these keys you can see this is kimmy this is max for stop in this is lewis hamilton blah blah blah and what we're going to do in this function is map all of that into a user's array so that we can ultimately display it in this list so right now we have all of these documents now all we need to do is map it into our user object very similar to what we did here so we're going to say let so i'll show you guys the long way of doing this and sort of the swifty way of doing this so we're going to say documents dot for each

Second subtitle

Main subtitle

document

then

we would have to go up here and create a
users array

like this

and then we would have to say let user
equal and we would just grab this line
of code guard let user equal blah blah
blah

try and it would just be uh document
dot data

as user dot self and you guys could
build that and notice that it works then
you could say users.append

user

and then outside of this for loop after
it's all done you would say completion
users okay

so this would work but it is uh like a
little bit

uh it could be a lot cleaner but um
first of all guys let's go ahead and see
if this uh

function works and where we're gonna
call it and then we will come back and
clean up this code a little bit we can
actually shorten this down to one line
of code using swift so um let's see
where do we want to call this function
so i think it's going to be in our
explore view model which we do not have
yet so we need to go to this explore
folder go to the view models folder and
create a new file so we're going to call
this explore view model or we could call
it like search view model doesn't matter
we should probably call it explorer just

Second subtitle

Main subtitle

to stay consistent with our naming
explorer view model we're going to go
here we're going to create a class
called explore view model
it's going to be
it's going to inherit from observable
objects
we're going to have a published property
of our users which is going to be this
user array
um and then you know we're going to have
an init function
actually i don't think we need the init
let's keep it there we're going to have
funk fetch users
and then we also need a service so we're
going to say let service equal user
service
and we're going to say service dot fetch
users
[Music]
and this will give us back our users and
we could say self.users equals users
but let's also just go ahead and say
print
users
are
the users we just delete that so we're
going to call this function when we
initialize this view model and it's
going to call that fetch users function
which is going to go into our service
class and call that fetch users function
which is what actually reaches out to
the database
you guys might be wondering why we have
a separate service class um

Second subtitle

Main subtitle

why not just put the fetch users function in here that way we don't have to worry about this completion stuff i'm going to explain that a little later and go over some improvements for how to improve upon using this user service so that it's more reusable and we don't have to initialize it in multiple places in the app i'll go over that in just a little um for now i just want to see if this is working so we're going to hop back into the explore view and create our view model so we're going to say at observed object

var view model equals explore view model so this is where we actually initialize this view model so once it initializes it's going to call that fetch users function and

let's just go ahead and see if we are getting back that print statement that gives us back all those users from the database right so this is going to tell us two things one that we're successfully fetching the information two that we're successfully mapping the information into our users object and it's

mapped into an array of users so that we can ultimately display those users in a list using that array

so this looks pretty good guys right we have all of our users in the app coming back right here so all of that stuff is successful you see we have that in the

Second subtitle

Main subtitle

name of the app dot user and then it gives us all of the data associated with each one of those users so displaying this in a list is now going to be actually really really simple we're just going to replace this for each with viewmodel.users and replace this underscore with the user and now we can actually go into this user row view and we can pass a user into this [Music] and the text here is going to be the user's username so we're going to say user.username here we're going to say user.fullname right and then we need to replace this with our kf image so we're going to say import kingfisher and we're going to say kf image with the url string user.profile image url let's actually make this a tad bit bigger guys i looked at the twitter app and the image does look a little bigger than what we have then we're going to just say dot resizable dot what else is it clip shape circle and then we're also going to say dot scaled to fill so this should look really good we're going to go back here we're going

Second subtitle

Main subtitle

to pass in this user
to our user row view so now this is
actually being populated with user data
and now we can even uncomment this line
of code and pass
the user into our
profile view as well
so let's go ahead and run this guys and
see if it works
conform to hashable so i think we
actually can just delete this guys that
id backslash self
build failed
what are you complaining about xcode
oh so yeah this needs um
a user now uh like i said guys we are
done with building the ui so we don't
really need that preview anymore
um what i like to do
when i'm at when i'm like building an
app sometimes is create mock data so i
would have some like sort of helper
class that had a bunch of mock objects
for like users and tweets and stuff but
uh for those purposes of this course we
don't really need to do that
um but look at that guys that is
freaking awesome right we got all the
users showing up in our application it's
in this awesome scrollable list that
looks really smooth and you guys will
notice that if i click on one of them
it'll display
real profile information about them or
real information about that user so if i
go to daniel ricardo it displays his
information if i go to lando norris it

Second subtitle

Main subtitle

displays his information we there's a couple more things we have to do on the profile like obviously we want this to be like a follow button we only wanted this be edit profile if we're looking at our currently logged in user so i'm logged in as valtteri botas i would want to go to valerie botos and have it say edit profile but if i went to like lewis hamilton then i would want this to say follower following so we have to implement some logic there but other than that guys that is looking absolutely amazing um i did say that we were going to go back to this service class and clean this code up a little bit so i'm going to show you guys how to make this a little cleaner we're going to say let users equal documents.compactmap so this applies like a mapping to the documents right returns an array containing the non-null results of calling the given transformation with each element of this sequence so basically it just allows us to shorten up this for loop so we can add some curly brackets there and we're going to say dollar sign 0 data as user dot self and then i believe we have to mark this with a try and we can literally just delete all this code and delete this code up at the

Second subtitle

Main subtitle

top and we could say completion
users which we already have right there
so go ahead and build that
and that should succeed so you guys
notice that we shorten this up to one
line of code so let me explain this
really quick this is what's known as a
higher order function in swift so it
allows us to pretty much just shorten up
like for loops and stuff like that so
basically guys what we want to do is
take data from this documents array and
map it or transform it into user objects
so we do so with this compact map we
have to use our try statement and then
this dollar sign zero just represents
each element in that documents array so
we can say dollar sign zero dot data as
user dot self and store that in this
users array and then execute our
completion handler so let's run this one
more time and see if our change or
refactor works there
and then after this we are going to get
into building a search functionality for
this guy so you guys can see this still
works beautifully
the only thing we have to do
is now add a search bar up at the top so
we can filter through our users and
that's going to be fun and exciting as
well guys so this is looking absolutely
awesome we're coming up on the point
where we're gonna uh sort of get into
the final
like um
what's it called pieces of the app where

Second subtitle

Main subtitle

we're gonna be uploading a tweet and displaying them in this list here um so it's really coming together guys like i said next up is search then uploading tweets and then some cleanup of the app and it's gonna be a pretty sweet project for you guys to have i hope you're enjoying this so far guys make sure you smash that like and subscribe button if you haven't already and also check out my other online courses that i have the links are in the description as i'm sure i've mentioned in this video already but let's keep going

all right guys so let's hop back into this explore view and figure out what we got to do um first off swift ui does have their own like custom search implementation

but you have to do it with their list component and i don't really like lists uh for a number of reasons um so i do like to keep this scroll view and lazy v stack formation

so that comes at a cost we do have to build our own custom search bar but it's really not that hard um

so let's just go ahead and do that now so we're going to hop into our components folder

and we're going to create a swift ui view

and it's going to be a search bar we're going to call it yeah we could just call it search bar um

so we're gonna have a binding property

Second subtitle

Main subtitle

up at the top guys text string
and let's go ahead and get our preview
opened up just so we can sort of see
this coming together
um
and in the preview just go ahead and say
dot constant
and give it some open quotes
and let's go ahead and just resume on
that preview
should come up pretty soon uh missing
label uh text
just like that
so hit resume
now that our preview is good and let's
just go ahead and build out this search
bar
all right guys so it's going to be
an h stack
so we're going to create this h stack
here we're going to create a text field
oops
text field
we're going to add search as the
placeholder and our text
is going to be that binding text
property
and we're going to give it a padding of
8
another padding of dot horizontal
24 i'll explain that in a sec dot
background is
color open parenthesis dot system gray
six
right there
so let's see how this looks so far
so try again on the preview

Second subtitle

Main subtitle

why are you yelling at me
that's really oh why did i do that no
open brackets stefan you dummy
all right so that looks pretty good so
far now we just need to stick a
magnifying glass in there guys we're
actually going to do that with an
overlay so but before that give it a
corner radius of 8 and then we're going
to say dot overlay
and hit enter on that and we're going to
say h stack on the overlay
image system name
magnifying glass
dot foreground color
of dot gray
dot frame and we're going to say min
width
is 0
max width is dot infinity
and the alignment is dot leading okay so
you're going to notice that moves the
overlay all the way over to the
left which is really nice
and then we're going to give it a
padding of dot leading 8 and that's
going to move it just a tad over right
there
um and then on the whole thing guys on
this whole h stack
we're just going to give it a padding of
dot horizontal
four
right so that's our search bar right
there it looks pretty smooth
um to get a better look at it you could
say dot preview layout

Second subtitle

Main subtitle

dot size that fits

and that will shrink the preview down to show you exactly what the search bar looks like there so that is uh pretty simple guys that was pretty quick and easy now we're just going to go and add this search bar into our uh

our explorer view here

so let's go back into our explore view and in this v stack you guys can literally just go here and above the scroll view add a search bar it's going to ask you for some text though so we need to provide it with some text so we're going to say state private var search text

equals blank

and then pass in that text search text and let's just give it some all around padding

and go ahead and run this on our simulator

to see how that's looking um shouldn't really need the preview anymore so let me just close that out

and let's see what this guy looks like

and then we can actually add the search functionality which isn't going to be that hard guys

um so we go here

and check that out boys and girls

um

so that looks great guys and now we just want to be able to like type stuff into the search bar

and

you know have it be able to filter

Second subtitle

Main subtitle

through these users for us

so

let's go ahead and get started on that
now

so basically what we want to do guys is
hop back into our view model
and we are going to want to filter this
user's array

based on

the user search text that they type in
so they type in some search text we want
to filter this array
to only return
users that where the username or the
full name contains that search text or
query so we're going to make a computed
property here we're going to call it
searchable users

it's going to be an array of users
but you guys notice here that we need
access to this search text so we're
actually going to place the search text
on the view model we're going to say
published var search text
equals a blank string
and we're going to go back to the
explorer view delete this state property
and for this guy uh delete that and say
dollar sign view model dot search text
okay so then we can now go back to the
view model and compute this searchable
user's property based on this search
text so we're just going to say if
search text dot is empty
and else
if it's empty we just want to return the
user's array if it's not we want to

Second subtitle

Main subtitle

filter it somehow

so really quickly guys we're going to
lower case the query we're going to say

let lowercase

query equal search text dot lowercase

then we're going to say return

users.filter and open up some curly
brackets

and let's say

zero

dot or dollar sign zero dot username dot
contains

oops dot contains

lowercase query

uh or

dollar sign dot zero dot full name dot
contains

lowercase query

so

now we should be able to build this and
it should work

and then we can just go back to our
explore view

and replace.viewmodel.users with
viewmodel.searchable

users

and we go and let's run the code

and this should

work guys

if my calculations are correct

so let's see moment of truth baby

let's go i'm excited i'm excited

so we go here and i'm going to start
typing out

daniel oh check it out baby let's go

uh okay well hold on

that should

Second subtitle

Main subtitle

it looks like our filtering lando norris
okay so it looks like the filtering is a
little off but it's pretty good right
we see carlos and then let's say like
dot signs right
um oh okay so i see the problem
uh let's go back to the view model
and let's say full name dot lowercase
dot contains query so this is case
sensitive guys and you want to just make
sure everything is lower case so we're
lower casing everything in the query
just in case the user would go like
carlos signs you know or
lando norris
um
right so and you can see that there's
auto capitalization on the first letter
which could potentially return like
nothing
um and you guys notice when we delete
everything everything just goes back to
normal which is great um
but we also have to full lowercase the
full name so if i type in like daniel
space
um
let's go ahead and just run this and see
if it works that space shouldn't affect
anything
so let's see
it's like 95 percent there right
and if there's a bug we can always come
back to it and fix it towards the end so
i want to see
lando
norris okay yeah now it's working

Second subtitle

Main subtitle

awesome

so i can now say daniel re

yep it's working so if i if i type in

like an x okay yep so guys that is

looking absolutely phenomenal

and let's make sure that when we click

on one of those users it actually takes

us over to their profile that takes us

over to carlos signs that's great

so that looks absolutely awesome right

now guys you notice we can still access

our side menu from anywhere in the

application and you know do all these

good things like go to our profile log

our user out log our uh and go to the

login page so this is looking freaking

awesome man i'm really proud of this

hope you guys are too the next step is

gonna be probably the funnest part of

the app we saved the best for last where

we're gonna get into

uploading tweets to the database and

displaying them in our user feed that's

going to be absolutely amazing guys

let's go ahead and hop to the next

chapter and get started with that all

right guys so let's take a look at this

new tweet view that's what i have opened

up in my xcode project here so first

thing we need to do is actually get some

user information into this view right we

have to display the user's profile image

and then allow them to tweet so after

that we're going to go over how to

upload a tweet to the database and then

how to download that information from

the database and display it in our feed

Second subtitle

Main subtitle
so**Second subtitle**

we have two options for how to get the user into this new tweet view guys we could create an environment object environment object var view model or we could we should call it off.viewmodel is of type off.viewmodel and we could do it this way and then you know say do like the if-let statement so let's see where is that circle so on this circle you could just go here and say like if let user equals authviewmodel.currentuser then we go up to the top import kingfisher and then replace this guy with kf image and then it's url string user dot profile image url bam all right then we say dot resizable dot scaled to fill dot clips or clip shape is a circle just like that so if you build that it should work right it looks good so let's actually just go ahead and run this and see what that looks like in our simulator that should be our first thing checked off the list there the other option you guys would have is to uh do what we did with the profile where we use a dependency injection right where

Main subtitle

we don't um

we didn't access the off view model here

we just initialized this with a user so

like i said before guys i think either

way is fine it's really just personal

preference to you

so let's go ahead and go here and you

notice that that looks really good right

that's our user profile image showing up

there beautifully so now let's go over

what we need to do to actually upload a

tweet to the database

so

i want us to go and create a new service

class so this is going to be called

tweet service

so it's going to be a swift file

hit next

tweet service

and this is just going to be a struct

called tweet service

we need to import firebase up at the top

and we're going to have a function

called upload tweet

just like that

and we need to look at exactly what we

need to upload to the database for

our tweet right so what i mean by that

is what data are we going to be

uploading to the database

um and how is that going to look so

basically i want us to create a new

collection that's going to be called

tweets and then each tweet will be its

own document very similar to this uh how

this user's guide is structured and then

we're just gonna have the associated

Second subtitle

Main subtitle

metadata of the tweet like

um

the you the user id of the owner of the
tweet the tweet id here

uh the tweet caption

and when that tweet was made okay so
that's pretty simple

um so let's just go ahead and figure out
what we need in this function

so it should just be the caption of the
tweet right so that's something that we
need to grab from this view and pass to
this function when we want to upload it
so that should be it right there

so

we also just we need a couple more
things but we can grab them inside of
this file so we can say guard let uid
equals auth.auth.currentuser.uid
else return so this is just going to get
the current user id because we need
information about who uploaded the tweet
right because if you guys look at tweets
it has information about the user that
uploaded it so we're going to use this
user id to eventually grab that user
data for our tweet

so then we're going to create this data
dictionary we're going to say let data
equal

and it's going to be a dictionary so
it's going to be uid and it's just going
to be uid just like that

and then we're going to have the caption
and it's going to be caption just like
that

then we're going to um

Second subtitle

Main subtitle

have a likes property as well that's going to be it's going to default to zero obviously and then we are going to have uh the timestamp just like this of the tweet and that's going to be this timestamp object that we get from firebase and we just pass in the date so that will give us the exact moment that the tweet was uploaded so that we can display how long ago that tweet was uploaded

okay

so let me just clean this up a little bit guys i'm pretty sure that's all we need

and let's go ahead and build that and we need to cast this as a string any dictionary like this because we have different data types as our values um

so that should be fine

now yeah okay now we just have to upload this guy

so

we're going to say let

we could just say

firestore.firestore.collection

tweets

dot document

and we're actually just going to auto generate a document guys so we just say document

and then we can say dot

set data

with a completion here

Second subtitle

Main subtitle

let me actually get that document guy
up there
and then we'll do the set data on a new
line
just like this so now it's going to ask
us for the data we want to upload
and then we have some completion
right so for now we can just add a print
statement that says like hey
did upload tweet
and when we get this print statement we
know that we'll have successfully
uploaded our tweet and we're actually
going to end up doing something here
a little bit later on guys so for now
let's just go ahead and figure out where
we want to implement this guy so
in our new tweet view
we want it to be so that when we hit
that upload tweet button right this guy
right here we upload our tweet
so we actually need to create a view
model for this guy
to uh
put that sort of logic in there so we're
gonna create a file in the view model
folder we're going to call it upload
tweet
view model
and i'm going to say class upload tweet
view model
it's going to be observable object
and then we're just going to have um
our function here called upload tweet
we're going to initialize our service
we're going to say let service equal
tweet service

Second subtitle

Main subtitle

and

uh

we could say with caption caption string

because we do need that caption data

then we're just gonna say service dot

upload tweet

and we're gonna pass in the caption

so that looks pretty good now we can

just hop back to this new tweet view

and

we can initialize our view model so

we're going to say at observed object

var.viewmodel

equals

upload tweet view model

and then right down here guys we're just

gonna go and say view model dot upload

tweet with caption and we are gonna pass

in the caption because we have access to

that inside of our new tweet view so

that's pretty smooth

let's go ahead and run this and see if

it actually uploads that information to

our database properly

and there's a little bit more work we

have to do there

and i'm also gonna give you guys some

more breakdown

of why

we use an extra service class there's a

couple reasons and i'm gonna give you

some pros and cons

and we're gonna like tie that back into

our user service class as well

so let's see we go here

what's happening

uh first test tweet right so then we say

Second subtitle

Main subtitle

tweet and bam it uploaded that looks great so let's go back here whoa oh okay that's weird um and let's see we refresh our page and guys we can now see some tweets and we see this tweet id it has our caption our number of likes um the timestamp and the owner of the user that uploaded it so let's go ahead and just double check that this is right i'm logged in as val terry bottas this was the uid and we can see that this provides us with all of valtteri bottas's information so that looks great right but we do have some issues here first off once that tweet has finished uploading we want to dismiss that view right so let's go over how we want to do that so we could go back to our tweet service and we could add a completion parameter here so we could just say completion at escaping parentheses arrow void and put a boolean variable in here and guys we could handle the error in this case because uploading a tweet is uh something where if it fails you do want to let the user know so let's go ahead and and see what this is going to look like we're going to say if let error equals error we're just going to say completion false and we're going to print debug failed to

Second subtitle

Main subtitle

upload tweet with
error error dot localized description
and then we return
okay and then
we can say uh
else basically just completion true
right so if the completion is true we
know we successfully uploaded it if it's
false we know that we failed so now if
we build our project you guys are going
to notice we get an error because we
have to implement this completion
handler so this is actually this is
going to be pretty sweet what we're
going to do here guys so basically we
want to dismiss this screen once the
tweet has uploaded right
so let's
uh figure out how we're going to do that
so first of all let's recall this
function
upload tweet
we pass in the caption
and now we have our um
what's it called completion handler and
we can just say success and then here we
can say if success
and else right so
here we want to say dismiss
screen
somehow right but we're inside of the
view model so it's like okay well how
the hell are we going to do that
right
and here we want to maybe like show
error message to user
okay

Second subtitle

Main subtitle
so**Second subtitle**

basically guys we want to have some published property um inside of our view model that's going to allow us to you know set that somehow and once that changes we could then go back to our new tweet view and execute this uh wrapped value.dismiss guy so let's go ahead and add a publish property we're going to say at published var uh did upload tweet and it's gonna be false and then here we if we're successful we just say self dot did upload tweet equals true and here we're not going to really handle the error guys that is going to be left in like the completed source code of the project right now i just want to get this functionality down so now that we've done that we have to go back to our new tweet view to actually figure out how we can listen for changes on that um did upload tweet uh published property okay so let's figure out how we're going to do that so we're going to double click on that v stack bracket guys the like the parent v stack and we're going to go all the way down to the bottom here and we're going to use this dot on receive

Main subtitle

function so

this adds an action to perform when this view detects data by emitted by the given publisher right so that's exactly what we want

we're going to publish that

did upload tweet property to true and then this is where we're going to receive that notification and be able to execute some code so we're going to say on receive

we're going to say view model

dot dollar signed did upload tweet and then this is where we're going to perform something and we get access to the publisher's output in this case it's a boolean so we're going to go here and say success and then we're going to say if success

presentation mode dot wrapped value dot dismiss

so

this is where we get access to that uh property like the actual output of the publisher and we want to make sure it's true before we actually dismiss the uh that sheet or that view because if we dismiss it when it's false that's not what we want to do right if it's false we would ideally display some error message to the user and we could also have that logic in here

so let's just go and make another tweet

we're gonna say test tweet to see if

this dismisses the view

and then we're gonna hit tweet and bam

look at that guys it dismisses it and we

Second subtitle

Main subtitle
can go back to

our tweets uh

collection

and this is the tweet that i just made

right

if you guys can't see that let me just

zoom in on it

test tweet to see if this dismisses the

view so that is absolutely perfect

that's dismissing our tweet our new

tweet view as expected now all that's

left to do is actually fetch these

tweets okay so the first thing we gotta

do to in order to get that done is go

back into our tweet service and write a

fetch tweets function so we're gonna go

here and say funk

fetch tweets

and we're gonna go back into this tweets

collection right here

and then we're gonna say dot get

documents

so we're going to say snapshot

not worried about the error and now

let's just go and say guard let

documents

equal

snapshot.documents else return

and let's just go ahead and say

documents

dot for each doc

let's say print doc dot data

and guys we're printing out all these

users here we don't want to do that

anymore

so let's go back to the user service and

just uh

Second subtitle

Main subtitle

where are we printing that guy out oh
that should be in our um
explore view model
so let's see explore
view model
and yeah that's where we're printing out
the user so we don't want to do that
so now we just need to go
and
call this fetch tweets function
using this tweet service okay so let's
figure out where we want to do that
um so we want that in our feed
i don't believe we have a feed view
model yet
so let's see yeah we don't so let's go
ahead and create that now
swift file
feed view model
and it's the same exact principles guys
we're going to create a class called
feed view model
it's going to be an observable object
we're going to initialize it
and when we initialize it we're going to
call this funk fetch tweets
we need access to our service we're
going to say let service equal tweet
service
and we're going to say service dot fetch
tweets
and then call our fetch tweets function
right here
so now guys we need to create this view
model inside of our feed view so we're
gonna say at observed objects
var view model

Second subtitle

Main subtitle

is equal to our feed view model
and we're going to go ahead and run this
i remove that navigation title i don't
think we need that down there
um but anyway guys yeah just go ahead
and run this and we should notice that
we get a print statement back showing us
those two tweets that we've created
and then all that's left to do is create
our data model
let's see
right so here they are let's go
um guys they should be down at the
bottom of your console i see my caption
that says first test tweet
and then i see my caption that says test
tweet to see if this dismisses the view
awesome
so
um very similar to what we did with our
user object guys we need to create a
tweet object so that we can map this
into some sort of object in our code map
all that data to an object so that we
can ultimately display them in this list
very similar to what we did
with our explore view so let's see
explore view we have this list or lazy v
stack whatever and we loop through
some array of data in this case it was
users in this case for the feed it's
just going to be tweets right
but we need that tweet object so let's
go ahead and create it now
so we are going to say new swift file
we're going to say it's a tweet
and we're going to create first of all

Second subtitle

Main subtitle

guys go ahead and import firebase
firestore swift we're going to create a
struct which is a tweet
it's going to be identifiable and
decodable
we're going to have a caption which is a
string we're going to have a timestamp
which is a timestamp
just like that
um then we're also going to have a uid
which is a string
and this user id is different than the
id guys the uid is the author of the
tweet that's their user id
so we need this document id bar
string which is optional
and i think we need to import firebase
to get access to that timestamp property
so let's see let's run that again
and those errors should go away
and let's see and then we also have this
likes property guys so we need to do
that as well
so we're going to say var likes is zero
excuse me the reason likes is a variable
guys and the rest of these things are
constants is because the likes can
change obviously right
oh this actually needs to be an integer
so
that looks pretty good for now
um we're going to work later on how to
get the actual user information so
actually let's just go ahead and add one
more property
it's going to be an optional user okay
and i'll explain this in a little bit

Second subtitle

Main subtitle

and don't worry guys i haven't forgotten
about explaining the purpose of these
service classes either those
explanations are coming
so really quickly let's go back into our
tweet service

and now we want to map this information
we're getting back
into a tweets array so first things
first let's add our completion of at
escaping

void and it's going to be an array of
the tweet object we just created
and then

we can just remove this docs dot for
each guy and use that nice fancy compact
map we used before so we're going to say
let tweets equal documents dot compact
map

then we'll say dollar sign zero dot data
as

tweet dot self
and then we out we can't forget to mark
that with a try
and then we're gonna say completion
tweets

now if we run our project
we're gonna notice we get an error and
that's going to be back in our view
model

because we need to
have this
recall this function so that it has the
tweets in the completion

and we also need guys a publish property
up at the top
so when this guy changes we can have

Second subtitle

Main subtitle

that take effect in our actual tweet
or our feed right so now we go here and
say self.tweets equals tweets
go ahead and hit command b
and
that should be good to go now all we
have to do is go back to our feed view
and
remove this sort of static
loop of 0 to 20 and say view model dot
tweets
and then we'll go tweet in
and in this tweet row view let's just go
ahead and do this now
we're going to hop into that to do that
guys i just hold down command and click
and then it says jump to definition
we need to initialize this with a tweet
i know i'm moving kind of quickly now
guys but it's because we've done this
stuff before and we're coming up on like
six hours of video time so i'm trying to
get this done a little bit faster um but
it should be nothing new at this point
right so now
we can't do anything with this circle
image yet because we don't have the user
data we haven't fetched that yet we need
to do that in a little bit
um
but we do okay so we can't do this
either but we can change the tweet
caption
so we can go here and say tweet.caption
and that looks pretty good um so let's
just go ahead and run this and see if
that is showing us with the actual tweet

Second subtitle

Main subtitle

caption that we have and then we're gonna go over how to get the user data um okay so we don't need this preview anymore

go back to your feed view we can delete this id backslash dot self

and then here we need to pass in the tweet

to the tweet row view

and let's run that oh we got one more error

var tweets view um okay

oh so this is happening in our um profile view so let's fix this fix this as well

uh actually guys let's just go ahead and comment this out for now

we're going to go over how to fix that in a little bit i want i really want to get this stuff working in the feed

so we can move on to sort of finishing the app up and adding in all these nice little extra touches

oh look at that baby let's go

so we have first test tweet and this is the second tweet that we have so now guys all we need to do

is focus on how we're going to get this user data for these tweets so let's figure out how we're going to do that so that we can actually display the correct user information there

then we are going to go over how to display the user's tweets in their profile view so we'll be able to go to like a user profile

and i'll be able to see carlos sanz's

Second subtitle

Main subtitle

tweets and i'll be able to see valtteri bottas's tweets and all that good stuff so this is really coming close to coming together guys get excited let's go ahead and figure out this next step okay guys so basically what we need to do is hop back into this feed view model and we're going to reuse one of our user service functions where we can fetch a user using a uid to get the tweet user information so you guys remember we have access to the uid of the user that owns a tweet the tweet so this serves as a pointer to the user id we need to go to in our user structure to fetch that user data and then essentially set that optional user property on the tweet so let's figure out how we're going to do it so go back to your feed view model and here we can say tweets.4 each tweet and we're going to say let uid equal tweet.uid then we can go here and we can create a user service and we can say user service dot fetch user with uid uid and now we have access to the user from this function so we can see how nice it is to have a reusable function like this right so we don't have to have that code in multiple places this is one of the main reasons we create a separate service class for these functions because if we just stuck all this code into a view model we wouldn't be able to reuse it we have to have it

Second subtitle

**Main subtitle
in a****Second subtitle**

sort of central location that allows us to reuse these functions so that's a prime example of why we create separate service classes guys it's for reusability

so um we actually need access to the index of the tweet that we're looking at guys so imagine we have an array of tweets right so let me just add a comment here we have like tweet one tweet two

tweet three tweet four

okay so each one of these tweets has a user property that we need to set on it so we're gonna loop through each one we're gonna get the uid property associated with that tweet and then we're going to fetch the user but now we need to be able to say hey tweet1 dot user equals user well we the only way for us to do that is to have access to the index so we're going to write this for loop a little differently we're going to say 4i

in

xero

up to tweets.count

we're going to say

let

tweet or let uid

equal tweets

i dot uid

and then we're going to copy and paste this guy

and now

we have for this particular iteration

Main subtitle

we're going to say
tweets
i dot user
equals user
so now whenever it sets that property
because this is a published variable
it's going to republish that
ooh tweets is a let constant what are
you talking about
this is a var
oh we need to say self.tweets
build
that right
okay so it was confusing these two guys
uh
the tweets and the completion and the
self(tweet so it's very important to
distinguish between those two things
so
now guys when this fetch tweets
functions runs it goes and gets all of
our tweets and displays the tweet data
and then sort of as a separate process
goes and gets the user information sets
the user property on the tweet object
and then displays will be able to
display the user information but we got
to go back to that tweet row view so
let's open that up tweet row view
and we need to go and now we can replace
some of this stuff with actual user
information
so
on this h stack guys we could just go
and cut
all this code out
or let's see

Second subtitle

Main subtitle

yeah so we could actually just cut this

uh

this h stack

and say like if let user equals tweet

dot user

paste it in there now we can do this we

can say hey text is uh

user dot full name

and this guy is going to be

user.username

and we're going to worry about that

timestamp in a little bit right now the

most important part is actually

displaying the user information on the

tweet and to see if our code is actually

working

oh snap and then we got to do the

profile image as well but we'll do that

in just a second

um by this point you guys should know

how to do that by now we've done it

quite a few times but we're gonna go

over it anyway

oh look at that boys and girls

so we noticed that we see valteri

bottas his full name and his username

and the caption so now all we need to do

is get this profile image figured out

and that's going to be pretty easy so

actually guys

uh

we put this in the wrong spot

so let's just go ahead and cut this h

stack information back

and uh put it back on its own we need to

wrap the we need to place the if let

inside of like or higher up in the code

Second subtitle

Main subtitle

so

let's see where do we want to place it
we have this h stack right so we want to
do it on this whole h stack right here
and i don't think there's any more user
information that we need to have right
so that's fine so let's go ahead and
where it says profile image plus user
info plus tweet let's just go ahead and
cut this whole thing out and then wrap
that in an iflat user equals tweet dot
user

and then paste that in there and now we
go up to the top we import kingfisher
and we say kf image
um what's it called

url right string user.profile image url
bam

and now we can remove this foreground
color we don't want that then we'll say
dot clip shape
circle

we need to make it dot resizable
and we have this frame
let's put that right there oops
and we also need to make it dot scaled
to fill

cool

all right so let's now go ahead and see
if this works guys
and it should if we did everything
correctly
and
that will have our app looking pretty
darn good right
moment of truth
let's go

Second subtitle

Main subtitle
let's go baby**Second subtitle**

all right we got our tweets showing up
looking really really good guys what you
should do is actually sign in as a
couple different users and
tweet uh make a couple different tweets
from some different accounts so that
this tweet will actually sorry this feed
will actually look like a real-life
twitter feed with information coming
from different users so that's looking
awesome the next step or piece of the
puzzle

is to be able to go to the user's
profile and actually see tweets that are
unique to them okay and then we're also
gonna fix some other stuff in this
profile with like this edit profile

button and all that good stuff so get
excited guys we are almost done i'm
super proud of how far you guys have
come i hope you're learning a lot and
that you're really loving this course
and there's a lot more stuff like this
to come but let's get this finished up
before i start talking too much

all right guys so i actually uh off
camera uploaded some tweets from some
different accounts i think you guys
should do the same

so the app is looking really really
great right now but there are a couple
things i want to take care of before we
go back into the profile view so we do
notice that the tweets are not in the
correct order so this was the most
recent tweet that i made but it's all

Main subtitle

the way at the bottom of the feed so we're going to want to sort our tweets by the most recent okay so we're going to go back to our tweet service and luckily we can handle this sort on the back end or server side so in this fetch tweets function let's go ahead and just uh indent that line right there and then we could say dot order by field and then we're gonna use this descending property so by field and descending we're gonna say timestamp descending is i believe it's false um and let's just go ahead and highlight this hit command or sorry control i just to indent everything properly and let's go ahead and run this and see if that fixes our problem if not we'll just change this value to true i can't remember which one it is um and then after that guys there is one other thing i want to do that uh is involved with the scrolling of the feed it's a very simple fix so that is not right so let's actually make this descending is true run that again and you guys notice that we do get a little bit of like a flicker on the tweets before they come into view that's because we're fetching the tweets first and then the user information but i think it's fine

Second subtitle

Main subtitle

um okay guys so that looks pretty good

right

um

now we have the most recent tweets on
the top and we didn't have to do any
sorting on the client side which is
really nice like we didn't have to sort
this array we told firestore to sort the
stuff for us which uh takes
more str which uh like removes stress
from the client side or doesn't add
unnecessary load or computation to the
client side which is what we want right
typically anytime you can put some load
or complex tasks on the back end that's
what you should do because on the client
side you want to preserve things like
battery life and limit the data usage
and stuff like that so that's looking
really good guys um i don't know if this
is what you guys app looks like but you
should have it so that like as you
scroll you can sort of see the
navigation bar get grayed out a little
bit so if that's not what yours is
looking like um i think to fix it all
you have to do is go on to this feed
view and then you can say dot navigation
bar
display mode and say dot inline
and that should fix the problem
because if if the navigation bar stays
translucent it kind of looks messy and
doesn't look super clean so those are
just some quick fixes that i wanted to
get done before we moved on to fetching
user profile tweets right so you guys

Second subtitle

Main subtitle

noticed that that looks really clean
and this app is seriously coming
together guys it's almost done
so now i want us to go to the profile
and figure out how to fetch tweets that
this user has made right so our feed is
just a collection of uh tweets from
multiple users obviously when you go to
a user's profile you want to be able to
see the tweets that they have made
uniquely so let's figure out how to do
that so we're gonna hop back into our
tweet service
and
we're gonna create another function here
called fetch
tweets
for uid
and that's going to be a string
and it's going to be really similar to
this
guy right here
so we're we can pretty much copy and
paste this
and we're just going to add something to
the query guys um so once again this is
going to have a completion of at
escaping you know
pretty much just copy this guy
bam
all right we just add another
parameter to our query
unfortunately
firestore is limited in this way you
can't order by one field
and then do another filter um that we're
going to do here so we actually have to

Second subtitle

Main subtitle

delete this order by and then we're going to say dot where field is equal to so just start typing out like where field is equal to there we go so we're gonna say where field uid is equal to um the let's see um we need the current user id or the yeah so i accidentally deleted that parameter so we're going to say 4 uid uid which is a string and then we pass that into here so basically what this is going to do guys is it's going to go into the firestore collection of tweets and it's only going to get us tweets where this uid field right let's see this uid field is equal to the user that we want to fetch tweets for so like let's say i go to daniel ricardo um he's gonna have his own set of tweets that like where the uid is equal to his uid firestore will filter all that data for us and only give us back daniel ricardo's tweets so that is absolutely amazing and it makes our life a lot easier and once again puts that stress on the back end instead of the client side which is what we want but the downside to this is like i said guys you can't do where field is equal

Second subtitle

Main subtitle

to and an order by on the timestamp so
we have to sort those ourselves but
luckily that's not too bad
so
we're going to say
actually go ahead and make this a var
and we're going to say completion tweets
dot sorted by
uh and we're going to go here and say
dollarsign0.timestamp.date
is
greater than
dollarsign1.timestamp.datevalue
so the sort is pretty simple and it
doesn't cost us a lot and unfortunately
it's just something we have to do here
so now let's go ahead and figure out
where we're going to call this
fetchtweets4uid function
and it's going to be back in our profile
view model
so we don't actually have a profile view
model
so we're going to create one
profile view model
and guys actually go back to that file
we didn't actually mutate this
so we can actually just keep it as a let
constant but anyway go back to your
profile view model now we're going to
say class profile
view model
observable objects
we're going to say at published var
tweets
is equal to an array of tweets we're
gonna say let service equal

Second subtitle

Main subtitle
tweet service**Second subtitle**

let user service

equal user oh no we don't need the user
service actually

uh we might um

we'll see no actually we don't that's

awesome we'll figure that out in a
second

so let's see we have the tweet service

when we initialize this guy we want to
fetch those user tweets so we're going
to say

funk

fetch user tweets

we're going to say service dot fetch
tweets for uid and we need access to the
user id so we're actually going to
initialize this profile view model with
the user id guys we're going to say init
we can actually just initialize it with
the user

and we're going to go up here and say
private let user be a user we can
actually make this service
private as well

and we're going to go here say self.user
equals user

and then we're going to say self.fetch
user tweets

so now we'll pass in our user id we'll
say user.uid
or a dot id

sorry and it's optional so let's just
give it a default value or we could say
guard let uid equal user.id
else return
and just pass in this uid

Main subtitle

and now we get back our array of tweets
and we can say self.tweets equals tweets
so
that's pretty freaking smooth um i am
going to oh let's delete that in it i'm
going to explain this in more detail
once we see it working guys so we're
going to get some errors
um
namely oh no we have it we actually
haven't initialized this profile view
model anywhere so let's go back to our
profile view and initialize this bad boy
so i'm going to go up to the top and i'm
going to say at observed
object bar view model
equals profile view model just like that
and
here we don't really
need the user anymore because we can
access it through the profile view model
and you guys will notice that this gives
us an error because we need to pass in a
user to it so this is actually how we
have to do that we just cast it just
like this as a profile view model then
we can say self dot view model equals
profile view model
and pass in that user and let's actually
just go ahead and delete this user
and that's going to throw some errors
and we can just replace user with view
model dot user
build again
oh so we actually need to make this user
a public property that's my bad
boom

Second subtitle

Main subtitle

the service remains private because this
is the only class we're going to access
that service through
so we build and here we just say view
model dot user dot that and view model
dot user dot full name
and that should be good guys
so
now that we've sort of
introduced that functionality
here's the last piece of the puzzle
so now we can go here and replace this
for each zero to nine with view model
dot tweets
use this tweet in
and now we create our tweet row view and
we have a tweet to pass into it
and this is our tweets view property all
the way down at the bottom below your
tweet filter bar
so
guys
i really think
that is going to work
and as promised
i'm going to give a full breakdown of
everything we just did
because i know we went through it kind
of quickly um even though i think i did
a pretty good job explaining it i'm
going to go over it one more time just
in case you guys are a little lost
well let's go ahead and see if this is
working all right so i'm going to go to
my profile
and check that out so
we're getting back lewis hamilton's

Second subtitle

Main subtitle

tweets right but we still need to fetch
the user data for those tweets
luckily we already have access to the
user from the profile so we know all of
these tweets are going to belong to
lewis hamilton so we have that user on
the view model already
so if we go back to our profile view
model
we can say
um
tweets
or let's see
we gotta loop through the tweets
and set the user property so we'll say
self or
dot tweets dot for each
tweet
tweet uh
um or actually guys we need to use we
need to do the four in loop so we're
gonna say four i in 0
up to tweets
we're going to say self dot tweets i dot
user
equals user
xerotweets.com there we go
uh it should equal self.user
yup
okay so that should be good to go
and now that the user data is all there
it's going to set that information up
for us nicely
but we do know this is working um
because we know that we made two tweets
with lewis hamilton
and there were two items showing up

Second subtitle

Main subtitle

there so let's go back to his profile
and bam look at that only lewis
hamilton's tweets are showing up and
let's go and check out some other guys
so i made some tweets from carlos sanz
forza ferrari bam that's looking awesome
and i made some tweets with valtteri
bosata's account yep
so that's looking absolutely amazing
guys amazing right

we are fetching tweets unique to users
um and then displaying them in their
profile and then in the feed we have a
list of tweets from everyone that's
absolutely amazing so as promised before
we move on i'm going to do a quick
breakdown of that feature

so i think you guys might be really
starting to see the beauty
of having these service classes separate
because

we have to do um
you know

things with these service classes in
multiple places in the app right like
fetching tweets or up maybe you want to
upload a tweet from a different place in
the app mainly uh the user service as
well right we want to be able to fetch
user information in multiple places in
the application so these service classes
give us reusable code where we can you
know

reuse these functions in multiple places
across the app instead of copy and
pasting the same exact code in multiple
places that's a

Second subtitle

Main subtitle

huge principle of programming don't
repeat yourself the dry principle but
anyway

um so in this tweet service we wrote a
function to fetch tweets for a
particular user id

and then we get back um a completion
handler of the tweets that belong to
that user so

we wrote a query to filter our tweets
where that field uid is equal to the
user id that we pass in in this function
so when i go to lewis hamilton's page
the the user id that i pass into this
function is lewis hamilton's user id it
goes and gets me all of the tweets that
belong to him and gives them back to me
and once again that filtering is
happening on the back end not on the
client side imagine
that i went and fetched all the tweets
that exist in the app and then
filtered down
to
to just get back two of lewis hamilton's
tweets out of potentially millions or
billions or trillions of tweets right
that would be extremely inefficient to
do on the client side and extremely
unnecessary so it's really awesome that
we can do this server side okay guys a
little bit of back-end engineering for
you guys

um

and then we just get the documents we
map them and then we sort the tweets
here once again we have to do that

Second subtitle

Main subtitle

client-side because firestore does not give us the capability to do this where field is equal to and order by timestamp all in a single query kind of annoying but it is what it is firestore is great so i'm not complaining so then we create a profile view model guys and we did something a little tricky here we initialized it with a user and that's because this view model needs to know something about the user that we are fetching data for right so we need to be able to get access to the user id of whose profile we're looking at so when i go to my current user's profile this is the current user id and we could get that uh another way but what happens when i go to you know someone else's profile i have i'm passing a user into this profile and then i'm passing that user into this view model as well and um then i can get access to the user id that i need to call this service fetch tweets for uid right then we get the tweets back and we know that this is like another like efficiency improvement we don't have to go and fetch the user for these tweets right because we know that we already have that data so we don't actually have to go and fetch that user for these tweets we can just say hey the user for the tweet is equal to self.user so that's just a quick

Second subtitle

Main subtitle

explanation of what we just did guys
there's a couple more things i want to
wrap up on the profile like we got to
fix this edit profile follow button
and other than that this has really come
together nicely um i am gonna show you
guys how to like tweets as well and then
we're gonna go over how to uh
display the users liked tweets in the
profile as well as a nice little bonus
feature for you guys um if you guys want
the rest of this stuff like how to
implement notifications and messages
like direct messaging and stuff and some
other really cool features just go to my
website as i'm sure i've mentioned the
link is in the description and you guys
will see how to get access to that
source code with all of the completed
features so let's keep going guys we're
almost done couple more features and
then we're going to wrap this up and but
it looks really great so far and i'm
really happy with what we got so let's
keep going
so let's talk about what's going to
happen when we want to like a tweet what
we have to do
so we want to click this button have
some process happen and then we want to
update the ui to show that this tweet
has been liked and then we want this
heart to be filled in with a red color
so we also want it to be
such that if we were to close the app
out like say imagine i just closed the
app and relaunched it that if i liked

Second subtitle

Main subtitle

this tweet i would want that data to persist and i would want the app to load with this tweet already liked right and that's going to be unique to specific users right so for example if i log in as lewis hamilton and i like carlos signs his tweet

when i close out the app and i'm still logged in as lewis hamilton i want to show that i've liked the tweet but if i were to log in as a different user that hasn't liked the tweet obviously we don't want to show that they've liked that tweet so we're gonna be uploading some information to the database here so i want us to hop into our tweet service guys and we're gonna write a function for us to like a tweet

and

before we go into what we have to do i always like to just add a print statement

like tweet here

and let's figure out where exactly we want to call this function so this like button if you guys remember exists in that tweet row view file

so we just go to core i believe we go to components and then tweets and we have tweet row view right and we go down here and we have all of our buttons right here and this is the like button so we can see that we left a comment that says action goes here so this is where we're going to want to call our function but it's like how do we actually get access to that like tweet function right

Second subtitle

Main subtitle

this is inside of a service class
you don't want to initialize a service
class inside of your inside of a view
you want to do that inside of a view
model so in this components folder guys
we're going to go ahead and create a new
folder two new
folders we're going to say view model
and name this guy views
and we're going to place that
tweet row view into the view
folder and then we're going to create a
view model here
so this is going to be a swift file
and we're going to call it tweet row
view model
and this is just like any other view
model it's going to be we're going to
name it tweet row view model
it's going to be an observable object
and let's go ahead and initialize this
service so we're going to say private
let service equal tweet service
and then we'll have a function called
like tweet
and we'll say service dot like tweet
and now we can initialize this view
model back in our view
so that we can call that function and
what we're actually gonna do guys is
replace
this uh
initializer here
uh with a tweet with an action with the
view model so we're actually gonna say
let view model
tweet row view model

Second subtitle

Main subtitle

and we're gonna initialize this guy with
a tweet
and then we're going to say
self.viewmodel equals tweet row
viewmodel
and we're going to pass in this tweet
here and we actually need to make that
initializer in our tweet
row viewmodel
so we're going to go here and say init
tweet
tweet
and then we can go here and say
let tweets
tweet
and self dot tweet equals tweet and guys
this helps us sort of stay consistent
with this mvvm architecture
um let me put this below the tweet
so
technically you're supposed to
initialize your model inside of the view
model and then initialize the view with
your view model and use the view model
to configure the view that's like
standard mvvm architecture
oops
tweet there we go
um so now everywhere we that we were
using just the tweet we can just say
viewmodel(tweet.user
uh
view
model(tweet.caption
and that should be good to go
and just to clarify guys there is like
it's okay to initialize your view with

Second subtitle

Main subtitle

like your model

um you don't always have to create a view model if it's just adding unnecessary complexity like sometimes the view model just adds an unnecessary layer

of uh of like communication between views and models

but in this case we do sort of need this view model because we are going to be making api calls from this view so and that's why we're uh introducing this view model here

and nothing really changes that much right so now

we you guys will notice that we have access to our view model here we can say view model dot like tweet which is awesome

so now let's actually go ahead and run this code guys

and see if that we're getting that print statement back

so if you guys remember in our tweet service it just says like tweet here and then we'll go over what we actually need to do to upload this information from the database and update our ui once the tweet has actually been liked

so i click this and you guys are seeing that we have this uh sorry my toolbar's in the way here guys let me see if i can get that out of the way i can't but you guys should be seeing that print statement show up in your console so that's looking really good all right now let's go over what we actually want to

Second subtitle

Main subtitle

do to get this information in the database

so guys what we're going to do is we want to keep track of what users have liked what tweets right so that'll make more sense in a little bit but essentially imagine we go to the profile and we want to fetch all the tweets that a user has liked that's what this like section is going to be for so on our users guy we're going to have another collection so the awesome part about firestore is that you can have sub collections on documents so basically we would create a sub collection here that's going to be called like user likes and then we'll just store all of the tweet ids that that user has liked and that's going to help us do a couple things which we'll see here in a little bit

so let's just delete this line of code um we're going to need our current uid so guard let uid equal off the auth.currentuser.uid else return and this is because we need to obviously you know store this in the currently used logged in users liked tweets right you wouldn't want to like a tweet and then not use the current user id for who liked that tweet so then we're gonna say let let's just create a reference here user likes ref equal firestore.firestore.collection

Second subtitle

Main subtitle

users.document

uid dot collection user dash likes so
you guys notice this is a pretty lengthy
uh sort of query

um

huh

it's weird that it's

like indents it weirdly we i mean you
can kind of leave that on one line
doesn't matter

all right now we're going to say uh
and then oh also guys i forgot

we also have on this tweets guy we have
to keep track of the number of likes so
we're going to have to go there update
the like count and then

um we are going to have to go into the
users list and update the the user's
list of like to tweets so a couple
things we got to do so now we're going
to say firestore dot firestore
dot collection
tweets

and you guys are going to notice that we
need the tweet id of the tweet that's
been liked right so in my database i
have to know which tweet to go to to
update the light count so what we're
going to do is add an input parameter

here like tweet

which is a tweet

and then we can go here and say dot
collection or sorry dot document
tweet.id just like that so that's
awesome and then we can say
dot

update data with a completion handler

Second subtitle

Main subtitle

and we're just going to say

likes

and then we're going to say tweet

dot likes plus one

so let's imagine the tweet likes was

like five it's going to go into the

structure

and this is uh

just give that a default value guys but

it should always be there

if you want we can say guard let tweet

id equal tweet id else return just to be

super safe

and then we could replace this with

tweet id

anyway we're just gonna go into that

structure and update the like so this is

going to go into the tweet id it's going

to find the likes guy and it's going to

increment it by 1.

so now we can just hit enter on this

completion block and then within this

completion we can update our user likes

reference so we can just go here guys

and say user likes ref

dot document

tweet id

dot set data

and this is a little tricky

so we're going to use a completion we're

actually just going to set it with a

blank dictionary and i'll show you guys

why in a little bit but on this

completion

we're going to just go ahead and say

print

debug

Second subtitle

Main subtitle

did like tweet

and now we should update ui okay so like
once the tweet has been liked and we
know that all of the back end processes
have completed then we can update our ui
and we're going to see this print
statement show up when all that is done
and we're going to update our model
object here

in order to make that happen but for now
i just want to see if this is uh if our
backend functionality is working and i'm
going to explain it to you guys in a
little bit more detail so that
it makes more sense so we're going to
get an error guys and that's because we
need to actually pass this tweet in to
this function and you guys will notice
that our initializer with this tweet on
this view model is now coming in handy
because we have access to the tweet that
we need to like so we're going to say
like tweet

and then here

we're just going to pass in that tweet
so now it knows what tweet to update the
likes property on and then it's going to
know how to update that user like
structure that we're going to create so
let's go ahead and run this
and see what we got boys and girls
so let's keep an eye on our back end
and let me go to like let's see
this tweet from lewis hamilton mercedes
is the best team on the planet bar none
i'm the best driver
um so that's this guy right here so

Second subtitle

Main subtitle

watch pay attention to this likes
parameter guys let me
zoom in on this a little bit
so i hit this button
bam and you guys notice that it updated
that likes property and then here let's
find lewis hamilton's uh
document now you notice we have a user
likes collection
and this is going to be a pointer to the
tweet id that we just liked guys and all
we need is the tweet id we don't need
any actual document data we just want
this to serve as like a list
of the user the the tweets that a
user has liked and we can take these ids
and go back into the tweets collection
and fetch them you guys will notice that
this is the tweet we just liked so that
matches up perfectly so
that's going to help us one figure out
whether or not we've already liked to
tweet and then it's also going to help
us display the list of the tweets that a
user has liked
but uh before we get into that we need
to actually update our ui
so
what i'm gonna do guys is hop into this
model our tweet model
and i'm gonna add another optional
property it's going to be called did
like and it's just a boolean
excuse me
um
typically guys i would just default this
to false so it doesn't have to be

Second subtitle

Main subtitle

optional

but um

that would cause this decode to fail

which is kind of annoying um but i guess

firebase just needs to update their code

but essentially you just make it an

optional boolean or optional property

and that won't cause the decode to fail

what i mean by that is we don't have

this did like property in the database

because um it doesn't make sense to keep

that like property in the database

because whether or not we liked it is

unique to each individual user we want

this to be sort of like a global data

store

that doesn't contain information unique

to specific users because some people

will have liked it some people won't but

all of this stuff is universal

information about the tweet this is more

so like a user relation to the tweet

so we're just going to make it this

optional property i'm pretty sure we

could give it a default value of false

but it does have to be optional

so just go ahead and run your code one

more time just to make sure that we're

still successfully fetching these tweets

and

and

loading our data correctly

so that should be fine

you guys notice this is still okay

but we noticed that hey this is the

tweet we liked and it's not showing we

we need like i don't want to hit this

Second subtitle

Main subtitle
like button again**Second subtitle**

and increment the like count a second time

each user can only like a tweet once

so

now what we're going to do is use this

model property of did like

and um figure out how we're going to adjust uh what we want to do with our uh like function

so back in our tweet service we can't modify this from here meaning we don't want to update our tweet model from this service class right that's just not how the architecture is supposed to work we need to do that from the view model so what we're going to do is add our own completion handler to this like tweet function so that we can access it in the view model so the view model will know when all of these back end processes have completed so we can update our model object and have that in turn update our ui that's how this lifecycle is supposed to work so we're going to say completion at escaping

bool

void

and here guys we're just going to go into this completion block and say completion true

and you can go ahead and build that and just to make sure we don't have any errors in our service class and we don't we're going to get one in our view model because we need to actually implement

Main subtitle

this completion handler now so just go ahead and delete that like tweet and retype it pass in the tweet and we don't actually need that boolean variable we just need to know that this process has completed so if you guys want you could actually go back to the tweet service and just delete that and make make this completion just execute just like that so now we can go back to our tweet view model or a tweet row view model and um let's actually just retype this one more time and you guys notice it just opens up a block for us now we can modify this tweet object okay so it actually has to be a variable and it actually has to be a published variable because when that tweet gets modified we want to publish that change and update our ui so we're going to say published var tweet which is a tweet here we set it and now we can say self(tweet.did like equals true and let's just go ahead and run this code or build it just to make sure there's no errors and we can continue on and it should be fine the last thing we need to do is update our view to make it so that when that did like property is true this is a red heart and when it's false it's this blank heart alright guys so let's hop back into our tweet row view

Second subtitle

Main subtitle

and we're going to want to
do some modification on this button here
so
uh what i mean by that is now this image
name is going to depend on whether or
not the tweet has been liked so we're
going to go here and we're just going to
cut that and we're going to say view
model
oops view model dot tweet dot did like
yes
is going to be heart dot phil
no is just going to be our heart
and then we're also going to need a
foreground color so we're going to go
here and say dot foreground color
view model dot tweet dot did like
and it looks like this needs a um
it needs to be unwrapped so let's just
give it a default value of false
because it it should default to false
and then uh we'll say the same thing
here
you know default value of false is going
to be
uh
color
dot system red oops
actually we guys we can just say dot red
guys and no is dot gray
so that should be the last piece of that
puzzle there's a couple other things we
have to do
um
oh we need a question mark right there
so build
that and let's go ahead and like a tweet

Second subtitle

Main subtitle

and see if that works but there are like
i said going to be a couple other things
we have to do to really complete this
feature so i'm going to run this
and the two things that we we have to do
guys are we have to write the code to
unlike a tweet
and then we have to write the code out
to check and see if a user has liked the
tweet when the app launches so you can
see here that i have liked one of these
tweets but i'm not i don't have any code
in my application to check and see if
i've liked all these tweets so
as far as the app is concerned none of
these tweets have been liked yet even
though that information exists in the
database we have to do a check so really
quickly i'm going to go ahead here like
this
huh
so i know why that didn't work and that
is because this is not an observed
object so
we have to go here and say observed
object var view model
and now we can run this again and that
should work
so guys if your view model is not
observed
um as we just saw there it does not
observe changes that publishers make
so you have to have this relation of uh
you know having the view model be
something that is observable and then in
the tweet in the view file you observe
the view model and the changes that

Second subtitle

Main subtitle

happens to it so let's see there and
boom let's go that's working
oh man i really love it when comes
together like that guys it pumps me up i
guess that makes me a dork but who cares
so that is working and it looks
beautiful we're liking a tweet and we
can go back to our database and we can
see that if we go to our users
let's see lewis hamilton should now have
a list of three tweets that he's liked
so let's just look at the ids mine is
lwr m9k and se
so i go back here and we should see that
we have lwr that's one of the tweets i
liked se that's another one and m9k is
the other one and you guys can see here
that we are accurate re accurately
reflecting the like count of those
tweets
so the next thing i want to do guys is
write some code to check and see if we
have liked a tweet and then
we will then update our ui
um to like make this information
actually persist right so if you guys
notice like right now it says i've liked
the tweet so pay attention seven time f1
world champion has entered the chat
and if i run the app again guys you guys
are going to notice that when it shows
up again
that tweet
isn't liked anymore even though that
information exists in the database and
this is pretty buggy right now because
if i hit this like button again it would

Second subtitle

Main subtitle**Second subtitle**

increment the like count to two right
which is not something we want to happen
so we need to write the code to unlike a
tweet and check and see if a tweet has
been liked before we can move on to the
profile stuff so we're almost done so
let's go back into our tweet service and
we're going to write a function called
func check
if
user liked tweet
and we need access to the tweet
and we're going to need another
completion of at escaping
void
but this time we actually need a boolean
variable guys because we want this
completion handler to tell us whether or
not the tweet has been liked
so we basically just need to go
into the user's uh
data structure
and we need to check this user likes
collection and if the tweet id that
we're looking at is exists in this list
we know that the user has liked that
tweet if not they haven't so that's
pretty simple to do we're just going to
say firestore
firestore dot collection
users
dot document and we need our uid here so
let's just copy and paste this guy right
there and we need the tweet id as well
so let's copy that in there as well
um we're going to say document uid
dot

Main subtitle

collection user dash likes right so

let's uh

make this a little bit more readable

just like that

dot

get document or sorry let's uh let's see

dot document

tweet id

dot get document

snapshot

and error and this is pretty smooth guys

we can just say completion

snapshot dot exists

so let's actually

say guard let snapshot

equal snapshot

else return because we don't want it to

be optional

and then we can say

completionsnapshot.exists

so this dot exists guy

is just going to tell us whether or not

that document exists right so if it

exists we get back true if not we get

back false and that will ultimately tell

us whether or not we liked that tweet so

let's go ahead and figure out where

we're gonna call this okay

so

back on the tweet row view model

when we initialize this view

or this view model with a tweet

we can just go ahead and call that you

know that function that we just wrote so

let's write a function check

if user liked tweet

and we call service dot check if user

Second subtitle

Main subtitle

like tweet we pass in our tweet and this completion will say did like in and we're just going to say self(tweet) or let's actually only update it if did like is true right because it's automatically false and if it once we get this back the only way the only time we want to actually update that that tweet is if it's actually been liked if it's false we don't want to unnecessarily publish that just to have it show the same information that the tweet hasn't been liked so we're just going to say if did like self(tweet).did equals true and let's build that

should be good to go then we're going to go here and say check if user like tweet and we're going to run our code and we should see all the tweets we've liked show up with a red heart if we did this right and that's going to be pretty freaking awesome man so let's hope for the best let's go yes let's go let's go right these are all the tweets that this currently logged in user has liked right so that's awesome let's like another one let's like first test tweet and let's just go ahead and run the app again and see if that persists and then the last thing we got to do guys before we move on to the profile oh

Second subtitle

Main subtitle

check that out that is so nice right
there let's go
um we just got to write the function to
unlike a tweet and then make it so that
if the tweet has been liked if we click
it again we unlike the tweet and delete
that information in the database so
let's hop back to the tweet service and
write this unlike tweet function
so we're just gonna say funk unlike
tweet
tweet
tweet
and we can just copy this completion
we need the same information
that we have here
and we uh
let's see
guys i think we don't actually need
to
uh make this a separate function but
let's go ahead and type it out and then
refactor it so basically we're going to
go into this
tweets collection and we're going to
decrement the like count
so we're going to say
firestore dot firestore dot collection
tweets
dot document tweet id
dot update data
and we're just gonna go
grab this dictionary right here
but we're gonna to say tweet.likes -1
and we're also going to add a guard
check here guys that says guard let
or we're just going to say guard

Second subtitle

Main subtitle

tweet.likes

is greater than or equal to zero else

return

or sorry greater than zero because if
the tweet like count is zero we don't
want to be able to unlike the tweet and
have the likes go into the negative so
this is just an additional logic check
or logic gate to make sure that we you
know keep our data

valid in the application

so then we're going to hit completion on
this error or enter on that completion
block

and we need to go and delete the
user likes ref

tweet id so we're going to copy that
and we are going to say user likes ref
dot

document

tweet

dot id dot delete

with a completion

and we just do underscore error in and
then we execute our own completion
handler

so that's our unlike tweet function

um it's pretty similar to this stuff

guys but

it's uh not similar enough to where i

would care to refactor it like

we have some different functionality

like we're deleting data

and stuff like that so i'm just going to
leave these as two separate functions
and it'll probably make our life a
little bit easier actually

Second subtitle

Main subtitle
so**Second subtitle**

now what we need to do is go to this
tweet row view

and we need to update how we call this
or you know
uh

the logic for this button so we're going
to say

view model dot tweet dot did like yes
and then we're gonna say uh view

model dot

unlike tweet

oh we need that function in the view
model

so really quickly guys let's go back to
the view model and add this unlike tweet
function we're going to say funk unlike
tweet

service dot unlike tweet

tweet

completion self(tweet

dot did like equals false

so

we want to be able to have the user
unlike the tweet and reset the like back
to its original state

so now we have this unlike tweet
function we're gonna go back to our
tweet row view and we're gonna say
unlike tweet no

view model dot like tweet

so

that should be good guys so what this is
saying um we need to give this a default
value again which is kind of annoying
yes
no

Main subtitle

let's go ahead and build that
and that should be good to go so what
this is saying guys is if the tweet has
been liked right so right now it's in
the liked state like did like is true
then when we click it we want to unlike
the tweet and if the tweet has not been
liked then we want to like the tweet so
that's what this is doing right here so
now let's go ahead run this test it out
and then we'll be able to move on to
fetching um
the tweets the users liked and
displaying them in the profile and we'll
be pretty much done with the app so it's
getting pretty exciting so
first off guys let's go ahead and
see
if this data actually gets deleted so
i'm going to just unlike this and you
guys notice that we had a value
disappear
and then if i go back to the tweets
let's see if we can find that tweet
seven time f1 world champion
boom the likes is now zero so let's
unlike another one
i believe it's this one you guys noticed
that the like count decremented to zero
and that you know now it's only two
liked tweets in my feed and if i were to
run this code again
we should notice that those other tweets
that we just unliked
are not showing up as liked anymore and
that the other two
uh persist as liked so that is looking

Second subtitle

Main subtitle

absolutely phenomenal guys that's um
you know just how you introduce a liking
system for any kind of content
so the next step here is to finish up
this profile so we want to one
adjust this button so that it shows the
right data or the right title and then
we want to be able to click on likes and
have it show a different list of tweets
so that's going to be really exciting
and that is going to wrap it up for the
course so almost done guys stick with me
through this to the end this last part
is going to be fun and add a little bit
more flair to the app so let's go ahead
and get it done now
all right guys so we are on the home
stretch here let's hop back into our
tweet service and we are gonna add
another function here to fetch our users
liked tweets so before we do that guys i
actually just want to grab all these
like functions so we have those here
let's create an extension on our service
so we're just gonna say extension tweet
service
and paste them all in there and we're
going to add a mark
and say likes here so this just helps us
like compartmentalize or segment our
service to make it a little cleaner
because you can see here that we're
adding quite a bit of functions and you
know if you guys want you could uh sort
of categorize these as well
um you could into like fetching and
uploading and stuff like that i just

Second subtitle

Main subtitle

wanted to show you how to sort of clean that up a little bit so we're going to add something to this extension we're going to say funk fetch liked tweets for uid uid which is a string and then we're going to have a completion of at escaping void and it's going to be an array of tweets just like this so basically guys the reason we have these this input parameter here is because when we go to a user's profile we want to fetch the likes for that specific user so for example here i would fetch lewis hamilton's suites and here you know if i go to like the search page i would want to fetch carlos sanz's tweets here right so that's why we pass in this uid and we're going to be calling this function in the profile view model so let's just go ahead and say print debug fetch liked tweets here and let's go ahead and see if we can figure out where to call this so we're going to go back into our core folder we're going to go to profile view models profile view model and you guys see that we already have our tweet service set up here so now we're just going to go here and say funk fetch liked tweets and we're going to say service dot fetch

Second subtitle

Main subtitle

liked tweets for uid user.id

tweets in and we're gonna need to unwrap
that so we're gonna go here
and then we are going to pass in that
uid to this function and let's just go
ahead and

um

call this in our init we're going to say
self.fetch like tweets and we're going
to create another published variable
called liked tweets

and it's going to equal
a tweet array so here basically guys
what we're going to do is make it so
that

when we select this tweets filter it
gives us back these tweets when we
select this likes filter it gives us
back these likes or liked tweets array
and you know

we're just going to add a function in in
here in just a second to get that done
right now i just want to make sure that
our function is actually getting called
and we get that print statement back
so let's see

it should show up when our profile view
initializes so let's go to our profile
and we see here that we have fetch liked
tweets here so that's looking really
good

now we just need to write the code to
fetch that stuff

so

it's basically what we have to do guys
is we have to do a chained fetch here so
what that means is we have to go into

Second subtitle

Main subtitle**Second subtitle**

this user likes structure grab the tweet

ids

take the tweet id

go back to this tweets guy and then grab
use that id to get the associated tweet
data so that we can actually construct
or create our tweet object in our code

so

uh it's not that bad let's just go ahead
and figure out how to do it so i'm gonna
say uh

first off

firestore dot firestore.collectionusers
dot document so i need my user id which

i have right here

so let's enter there and then we're

gonna say dot collection

user dash likes

dot get documents

snapshot and we don't need to worry

about the error

then what we're going to say is guard

let documents

equal snapshot documents else return and

these guys are a bunch of blank

documents

right you guys remember all we have is
this list of

ids but these are actually still treated
as documents they're just blank so we
just need to grab all of the ids that we
have here so basically what we can say
is documents

dot for each

doc

in

then we're going to say let tweet id

Main subtitle

equal doc dot document id just like that
so that's going to give us the actual
tweet id and once again that represents
what this id is right here this is the
document id and the way we set our back
end up is that this is a pointer to the
actual tweet that we need to get so now
um

i don't believe we have a fetch tweet

function like a single fetch tweet

so we're gonna have to write that out

so we just need to say firestore dot

firestore dot collection

tweets

dot document tweet id

and let's go ahead and press enter there

and then we just say get document

completion snapshot error

and now we just need to map our tweet or

that document data into a tweet guys so

we're gonna say

uh guard let tweet equal

um

hold on

we need to get the document first so

we're going to say guard let document

equal snapshot dot document ooh no that

is the document awesome

so one sec

bear with me so now we're just going to

say guard let tweet equal try

snapshots dot data as tweet

dot self

else return

and we need to be able to append this to

some sort of array so we're going to

create that up at the top we're going to

Second subtitle

Main subtitle

say var tweets equals tweet
and then we're going to say tweets dot
append
oops
tweet right
and then guys um
i believe
then we say completion
tweets just like that
so
that is looking pretty good
now we just need to uh go ahead and set
that liked tweets guy in our profile
view model so here we just go ahead and
say self dot liked tweets equals tweets
and now we need to add our filter
function right so
basically guys we're just going to add a
function here we're going to say funk
tweets
for filter filter
tweets uh filter view model
and it's gonna give us back an array of
tweets
just like this and we're going to say
switch filter
case tweets we're just going to return
tweets
case replies we can also just return
tweets because we don't have replies and
then for the likes we're going to return
liked tweets so this is super super
smooth guys we're using this filter to
help us you know select the correct data
source that we want our list to be
populated with and then all we have to
do is go back to our profile

Second subtitle

Main subtitle

view and instead in the tweet list view

so right down here we just say view

model dot tweets

for filter

uh

and i believe we can just say self dot

selected filter

so that is super super smooth let's go

ahead and run this

and see if that is showing up correctly

and by my calculations

it should be

and then the last thing we got to do is

update that edit profile button and this

app is freaking done

let's go i'm pumped man

so let's go to our profile

boom

likes oh

oh wait dang we still gotta fetch the

user information

almost there guys we are almost there

let's keep it going so we do actually

have to fetch the unique users for this

um

because we uh we can like other people's

tweets and display them there those

these tweets aren't always going to be

from us

so in our tweet service

let's see let's see

we could do that here

or we could do it in our profile view

model

so let's see um i think we did that

somewhere else

in our feed view model maybe

Second subtitle

Main subtitle

yes so we did it here
so we can literally just go here
copy this
in your fetch tweets function in the
view model guys
no wait yeah yeah we need to copy that
and then we can go here
and say for i in tweets.com let that
equal uid self.user service so we need
our user service in here
with uid uid user in self.litetweets
i dot user equals user so now we just
need to create this user service so
we're going to say
private let user service equal user
service
and we can build this
and that should be good to go guys
so that is looking pretty good
i think that's gonna solve our problem
and then we'll be done
so let's go to lewis hamilton's profile
we go here and boom
check that out baby let's go
so we can see here that this is lewis
hamilton's list of like tweets and now
guys we kind of have that really awesome
animation too where like it like sort of
reloads that tweet list for us right so
that is looking
absolutely phenomenal right there um app
is looking super super smooth the last
thing we need to do guys is update this
button right here and that is going to
be a super simple fix so pretty much
what i want us to do is hop into our
user model file

Second subtitle

Main subtitle

and we're going to create this computed property we're going to say is current user

which is a bool

and we're going to return and we actually need to import firebase up at the top

we're going to say

auth.auth.currentuser.uid

equals id

so basically if the currently logged in user's id is equal to this userid property on our object we know that they're the current user and that's going to help us figure out this button title so let's actually hop back to our profile view model and we're going to create a computed property here so i'm going to say

var

action button title

and it's going to be a string

and we're going to say if

self dot user dot is current user

uh wait we can actually just wrap this in a ternary we can say return user dot is current user yes

edit profile

no

follow

and we'll just replace where we see edit profile with this view model dot action button title so profile view

let's see edit profile we just go here

guys and we say view model

dot action button title

so that's in this action buttons

Second subtitle

Main subtitle

property
and we can run this
and
that should work and guys
now we are finally done with this app
right that's almost seven hours of
content give yourself a huge pat on the
back man this is an amazing project to
add to your portfolio you guys learned
some amazing skills we pretty much just
built a full scale application with
nothing but swift ui so let's go ahead
and check and see if this works we're
seeing edit profile there which is great
we go to someone else's profile and we
see that follow button showing up there
that's awesome and we notice that this
likes or this guy here is also
successfully uh showing up
so man uh what a ride that has been guys
thanks so much for joining me on it once
again if you guys want the completed
source code for this um go ahead and
check out the link in my description
also have some links in my description
to a bunch of other courses that i have
with both ui kit and swift ui so make
sure you guys check that out as well
like and subscribe to the channel tons
more free content coming your way just
like this i'm probably going to do
either instagram or chat app or maybe
even a crypto app next so get excited
for that guys big things coming to the
channel once again thanks so much for
taking time out of your day to code with
me see you later guys peace out

Second subtitle

Main subtitle

Second subtitle