

## (49) COMPLETE User Login / Sign Up App | Swift UI + Firebase | Async / Await - YouTube

### Main subtitle

what's going on everybody Stefan here

from App stuff welcome back to the

channel hope you guys are doing well man

I had a lot going on the last couple

months it's been a while changed

Apartments changed jobs that's a video

for another time right now we got an

awesome video for you guys here today

where we're going to be building a full

authentication Suite using Swift UI and

Firebase now this has all the latest and

greatest Technologies man with async

await navigation Stacks the latest

version of Swift UI and Firebase all of

that good stuff it's going to help you

sign users up sign them in sign them out

we're going to have a custom user

profile all of that awesome stuff so

without any further Ado guys let's go

ahead and check out what the app's going

to look like and see what you're going

to be building before we jump into the

code

[Music]

so we have our beautiful login page here

you guys see we have an icon some custom

### Second subtitle

chuyện gì đang xảy ra vậy mọi người Stefan ở đây

từ Nội dung ứng dụng chào mừng các bạn quay trở lại

kênh, hy vọng các bạn đang làm tốt.

Tôi đã có rất nhiều

việc phải làm trong vài tháng qua, đã lâu rồi các căn hộ đã thay đổi.

video tuyệt vời dành cho các bạn ở đây hôm nay,

nơi chúng ta sẽ xây dựng một

Bộ xác thực đầy đủ bằng cách sử dụng Swift UI và

Firebase, giờ đây, bộ này có tất cả những Công nghệ mới nhất và

tuyệt vời nhất với

điều hướng không đồng bộ. Xếp chồng

phiên bản mới nhất của Swift UI và Firebase, tất cả những thứ

đó thứ tốt nó sẽ giúp bạn

đăng ký người dùng đăng ký họ đăng nhập đăng xuất

chúng tôi sẽ có

hồ sơ người dùng tùy chỉnh tất cả những thứ tuyệt vời đó vì vậy

không cần thêm bất kỳ Ado nào nữa, hãy tiếp tục và xem ứng dụng sẽ

trông như thế nào và xem những gì bạn sẽ xây dựng trước khi chúng tôi bắt đầu viết mã

[Âm nhạc]

để chúng tôi có trang đăng nhập tuyệt đẹp của chúng tôi ở đây,

các bạn thấy chúng tôi có một biểu tượng, một số

**Main subtitle**

input fields we also have some form

validation so this button is not  
clickable right now until the user fills

out the form here clicking this down at

the bottom we have our sign up page so

this is where a user would create their  
account if they don't have one we also

have a really awesome feature here guys  
that I love which is confirming your  
password

so you guys can see here that  
it will indicate to me whether or not my  
passwords match which I think is really

cool and let's go ahead and take a look  
and see what it looks like to log in so  
if we say test gmail.com this is just a

test account I created pay attention to

this button guys

once the password has a certain amount  
of characters and both Fields have you  
know the correct inputs our button  
becomes validated and allows us to click

on that and then if we hit sign in it

will log Us in and take us to our custom

profile page here so this is information  
that is actually being fetched from

Firebase so Firebase is a back end  
system that we're going to set up so

**Second subtitle**

trường nhập tùy chỉnh, chúng tôi cũng có  
một số

xác thực biểu mẫu nên nút này không thể  
nhấp được ngay bây giờ cho đến khi người  
dùng điền

vào biểu mẫu ở đây, nhấp vào biểu mẫu này  
ở

dưới cùng, chúng tôi có trang đăng ký của  
mình, vì vậy

đây là nơi người dùng sẽ tạo

tài khoản của họ nếu họ không có tài khoản,  
chúng tôi cũng

có một tính năng thực sự tuyệt vời ở đây,  
tôi love đang xác nhận mật khẩu của bạn

để các bạn có thể thấy ở đây rằng

nó sẽ cho tôi biết liệu

mật khẩu của tôi có khớp hay không, điều  
mà tôi nghĩ là thực sự

tuyệt vời và hãy tiếp tục

và xem giao diện đăng nhập sẽ như thế nào  
nếu chúng ta nói thử nghiệm gmail.com, đây  
chỉ là

tài khoản thử nghiệm mà tôi đã tạo, hãy chú  
ý đến

nút này nhé các bạn

khi mật khẩu có một số lượng

ký tự nhất định và cả hai Trường bạn đều  
biết đầu vào chính xác, nút của chúng tôi  
sẽ được xác thực và cho phép chúng tôi  
nhấp

vào đó và sau đó nếu chúng tôi nhấn đăng  
nhập, nó

sẽ đăng nhập Chúng tôi và đưa chúng tôi  
đến trang hồ sơ tùy chỉnh của chúng tôi

tại đây, vì vậy đây là thông tin

thực sự được tìm nạp từ

Firebase, vì vậy Firebase là

hệ thống phụ trợ mà chúng tôi sẽ thiết lập  
để hệ thống

**Main subtitle**

this has full back-end support which is what you want for an app like this you

want to be able to save users information in a database and their authentication information like their user their their username and password and database and we're going to be doing

all of that with Firebase which is really awesome and you guys will see here we also have the functionality to be able to close out of the app and relaunch it and it will keep us logged

in unless we sign out which is what happens right here so that's just a brief rundown of the app guys you can once again drag and drop this into any

one of your apps that needs user authentication so with that being said let's go ahead and hop into xcode now

and get started with the code all right guys so let's get started by

opening up xcode and creating a new project I'm going to hit next here I'm

going to call this Swift UI

auth tutorial

and obviously make sure Swift UI is

selected as your interface language is Swift and we don't need core data or

tests uh let's go ahead and hit next

**Second subtitle**

này có đầy đủ hỗ trợ -end, đó là

những gì bạn muốn cho một ứng dụng như thế này, bạn

muốn có thể lưu

thông tin người dùng trong cơ sở dữ liệu và thông tin xác thực của họ như

người dùng của họ, tên người dùng và mật khẩu

và cơ sở dữ liệu của họ và chúng tôi sẽ thực hiện

tất cả điều đó với Firebase, điều này

thực sự tuyệt vời và các bạn sẽ thấy

ở đây, chúng tôi cũng có chức năng

có thể đóng ứng dụng và

khởi chạy lại ứng dụng và nó sẽ giữ cho chúng tôi đăng

nhập trừ khi chúng tôi đăng xuất, đó là điều

xảy ra ngay tại đây, vì vậy đó chỉ là một

tóm tắt ngắn gọn về các ứng dụng,

một lần nữa, bạn có thể kéo và thả ứng dụng này vào bất

kỳ ứng dụng nào của bạn cần

xác thực người dùng, vì vậy, như đã nói,

hãy tiếp tục và chuyển sang xcode ngay bây giờ

và bắt đầu với mã

nhé mọi người, vì vậy hãy bắt đầu bằng cách

mở xcode và tạo một

dự án mới mà tôi sẽ thực hiện tiếp theo ở đây, tôi

sẽ gọi đây là hướng dẫn xác thực giao diện người dùng Swift

và rõ ràng là đảm bảo rằng giao diện người dùng Swift được

chọn làm ngôn ngữ giao diện của bạn là

Swift và chúng tôi không cần dữ liệu cốt lõi hoặc

kiểm tra uh, hãy tiếp tục và nhấn tiếp theo

**Main subtitle**

and go ahead and get started with  
setting up our project directory  
[Music]

so I'm going to go here and create a  
couple groups if you guys have watched  
my tutorials before you know I always

start every xcode project like this I  
like to organize my files so you can

right click here and select new group  
alternatively you can hit command option  
n and it will create some groups for you  
so let's go ahead and create like five I

think we need  
so the first one is going to be called  
app  
the next one is going to be called  
components  
and this is where we're going to put all  
of our reusable components

um this next one is going to be called  
The Core  
and this is where the majority of our  
views and view models will live this

next one is going to be called  
uh let's see model I believe  
and I think that's all we need so let's

go ahead and just delete that last group  
we only need four app components core

and model  
and we are going to start by moving this

**Second subtitle**

và tiếp tục và bắt đầu  
thiết lập thư mục dự án của chúng ta  
[Âm nhạc]

vì vậy tôi sẽ đến đây và tạo một  
vài nhóm nếu các bạn đã xem  
hướng dẫn của tôi trước khi bạn biết tôi  
luôn

bắt đầu mọi dự án xcode như thế này, tôi  
muốn sắp xếp các tệp của mình để bạn có  
thể

nhấp chuột phải vào đây và chọn nhóm mới.  
Ngoài ra, bạn có thể nhấn tùy chọn lệnh  
n và nó sẽ tạo một số nhóm cho bạn,  
vì vậy hãy tiếp tục và tạo như năm nhóm.  
Tôi

nghĩ chúng ta cần  
cái đầu tiên sẽ được gọi là  
ứng dụng,

cái tiếp theo sẽ được gọi là  
các thành phần

và đây là nơi chúng ta sẽ đặt tất  
cả các thành phần có thể tái sử dụng của  
mình, cái

tiếp theo này sẽ được gọi là  
Cốt lõi

và đây là nơi chứa phần lớn chế độ  
xem và chế độ xem của chúng tôi sẽ hoạt  
động, chế độ xem

tiếp theo này sẽ được gọi là  
uh hãy xem mô hình tôi tin

và tôi nghĩ đó là tất cả những gì chúng tôi  
cần, vì vậy hãy

tiếp tục và chỉ cần xóa nhóm cuối cùng,  
chúng tôi chỉ cần bốn thành phần ứng dụng  
lỗi

và mô hình

và chúng tôi là sẽ bắt đầu bằng cách di  
chuyển

**Main subtitle**

app file up into our app folder

and then we are going to uh

within our core folder guys we're going to create another one

called root

and we are going to put our content view

there so this is like the root view of our application okay so that's our content View

and now what I want us to do is also

create another folder here

called Authentication

and one more folder called profile

and let's adjust these so that they are in alphabetical order

uh yep that looks good so let me break this down really quick all of our authentication stuff like this login view sign up view or auth authentication

view model is all going to live here

um we're going to have a profile page as you saw in the demo that's going to go here and then our content view is

obviously inside of our root and then

our model folder is going to have all our data models and stuff so now that

we've got our project directory

organized guys let's go ahead and get started with actually creating some of the user interface components for our

**Second subtitle**

tệp ứng dụng này vào thư mục ứng dụng của chúng tôi

và sau đó chúng tôi sẽ vào uh

trong thư mục lõi của chúng tôi, chúng tôi sẽ tạo một tệp khác

có tên là root

và chúng tôi sẽ đặt chế độ xem nội dung của mình

ở đó để đây giống như chế độ xem gốc của ứng dụng của chúng tôi được rồi, đó là Chế độ xem nội dung của chúng tôi

và bây giờ, điều tôi muốn chúng tôi làm cũng là

tạo một thư mục khác ở đây

có tên là Xác thực

và một thư mục nữa có tên là hồ sơ

và hãy điều chỉnh chúng

theo thứ tự bảng chữ cái

uh vâng trông ổn đấy Hãy để tôi phân

tích vấn đề này thật nhanh, tất cả

nội dung xác thực của chúng tôi như chế độ xem đăng nhập, chế độ xem đăng ký hoặc

mô hình chế độ xem xác thực xác thực đều sẽ tồn tại ở đây,

chúng tôi sẽ có một trang hồ sơ như

bạn đã thấy trong bản demo sắp ra mắt ở

đây và sau đó chế độ xem nội dung của chúng tôi

rõ ràng là bên trong thư mục gốc của chúng tôi và sau đó

thư mục mô hình của chúng tôi sẽ có tất cả các mô hình dữ liệu và nội dung của chúng tôi, vì vậy bây giờ

chúng tôi đã tổ chức thư mục dự án của mình,

hãy tiếp tục và bắt

đầu thực sự tạo một số

các thành phần giao diện người dùng cho ứng dụng của chúng ta

**Main subtitle**

app here we're going to start off with this login view alright guys so let's hop up to our authentication folder and create a new file just hit command n it's going to be a swift UI View and we're going to hit next and we are going to call this login View and you guys might have guessed it this is going to be the file that we build our login view with and let's go ahead and just get started right away so immediately guys we want to note that we want to wrap this inside of a navigation stack and that's because we want to be able to move back and forth between our login and sign up views okay we need to use a navigation stack to do that keep in mind I'm saying navigation stack not navigation view Swift UI recently released a new API that replaced the old navigation views with something called a navigation stack if you guys want to learn more about that I have a tutorial for that the link is in the description make sure you check that out now if you want to learn more about it with a fun tutorial

**Second subtitle**

ở đây, chúng ta sẽ bắt đầu với chế độ xem đăng nhập này nhé các bạn, vì vậy hãy chuyển đến thư mục xác thực của chúng ta và tạo một tệp mới chỉ cần nhấn lệnh và nó sẽ là một Chế độ xem giao diện người dùng nhanh chóng và chúng ta sẽ sẽ nhấn tiếp theo và chúng tôi sẽ gọi đây là Chế độ xem đăng nhập và các bạn có thể đoán nó sẽ là tệp mà chúng tôi xây dựng chế độ xem đăng nhập của mình và hãy tiếp tục và bắt đầu ngay lập tức vì vậy chúng tôi muốn ngay lập tức lưu ý rằng chúng tôi muốn bọc cái này bên trong ngăn xếp điều hướng và đó là vì chúng tôi muốn có thể di chuyển qua lại giữa chế độ xem đăng nhập và đăng ký của mình, được rồi, chúng tôi cần sử dụng ngăn xếp điều hướng để thực hiện điều đó, hãy nhớ rằng tôi đang nói ngăn điều hướng không phải chế độ xem điều hướng Giao diện người dùng Swift gần đây đã phát hành một API mới thay thế chế độ xem điều hướng cũ bằng một thứ gọi là ngăn điều hướng nếu các bạn muốn tìm hiểu thêm về điều đó. Tôi có một hướng dẫn về liên kết đó trong phần mô tả, hãy đảm bảo bạn kiểm tra điều đó ra ngay bây giờ nếu bạn muốn tìm hiểu thêm về nó với một hướng dẫn thú vị,

**Main subtitle**

so let's go ahead and get started by wrapping this inside of a navigation

stack

and within that we are going to have a v-stack because obviously we have a vertically stacked list of view

components here so we're going to say

v-stack and let's go ahead and just create a quick outline for this so we're

going to say this is our image here is where we're going to put our form fields

and here's our sign in button

here's our sign up button

so now let's go ahead and just get started with what we need to do guys to start rendering this out so get started with the code so before we do I'm going

to drag an asset into my project you guys don't have this it's part of the completed source code you don't need it

you could literally just go to Google

download any image and drop it into your xcode project but if you guys do want

access to the source code just head over

to my website at stefanocode.com and check out my pricing plan so any one of these plans will get you access to the

YouTube courses which this will be added

**Second subtitle**

vì vậy hãy tiếp tục và bắt đầu bằng cách gói phần này bên trong ngăn xếp điều hướng

và bên trong đó chúng ta sẽ có một ngăn xếp v vì rõ ràng chúng ta có một danh sách xếp chồng lên nhau theo chiều dọc xem

các thành phần ở đây, vì vậy chúng tôi sẽ nói

v-stack và hãy tiếp tục và tạo một phác thảo nhanh cho điều này, vì vậy chúng tôi

sẽ nói đây là hình ảnh của chúng tôi, đây là nơi chúng tôi sẽ đặt

các trường biểu mẫu của mình

và đây là nút đăng nhập của chúng tôi

đây là nút đăng ký của chúng tôi,

vì vậy bây giờ hãy tiếp tục và bắt đầu với những gì chúng tôi cần làm để bắt đầu kết xuất cái này, vì vậy hãy bắt đầu với mã vì vậy trước khi chúng tôi thực hiện, tôi sẽ

kéo một nội dung vào dự án các bạn không có, đây là một phần của mã nguồn đã hoàn thành, bạn không cần nó,

theo nghĩa đen, bạn có thể chỉ cần truy cập Google để

tải xuống bất kỳ hình ảnh nào và thả nó vào dự án xcode của mình nhưng nếu các bạn muốn

truy cập vào mã nguồn thì chỉ cần hãy truy cập

trang web của tôi tại stefanocode.com và xem gói giá của tôi để bắt kỳ

gói nào trong số các gói này sẽ giúp bạn truy cập vào

các khóa học YouTube mà gói này sẽ được thêm

**Main subtitle**

to but the other plans give you upgraded

benefits right so the diamond plan is the most popular right now it gives you access to everything on the site for 40

bucks a month that's YouTube courses Pro

courses and all of the source codes so that's thousands of dollars worth of value for just 40 bucks a month so make sure you guys go ahead and check that

out and sign up to become a member on the site to get stuff for free and get

discounts on awesome stuff

so let's go back to xcode and I'm going to now create my image here let me

minimize this really quickly so let's see we're going to create an image Firebase Dash logo and guys for those of

you who don't know this is obviously just the name of whatever is in that your assets folder we need to make this

dot resizable

dot scale to fill

and that's still going to look terrible

so we need to give it a frame of a width

and a height of a hundred by 120.

and that looks great right and then

we're going to say dot padding

dot vertical

32 pixels okay

so awesome and at the bottom of all this

**Second subtitle**

vào nhưng các gói khác cung cấp cho bạn quyền lợi được nâng cấp,

vì vậy gói kim cương là lựa chọn tốt nhất phổ biến nhất hiện nay, nó cho phép bạn truy cập vào mọi thứ trên trang web với giá 40

đô la một tháng, đó là các khóa học YouTube Các khóa học chuyên nghiệp

và tất cả các mã nguồn, do

đó, giá trị hàng nghìn đô la

chỉ với 40 đô la một tháng, vì vậy hãy

đảm bảo rằng các bạn sẽ tiếp tục và kiểm tra điều đó

và đăng ký để trở thành thành viên trên trang web để nhận nội dung miễn phí và được

giảm giá cho những nội dung tuyệt vời,

vì vậy, hãy quay lại xcode và

bây giờ tôi sẽ tạo hình ảnh của mình ở đây, hãy để tôi

giảm thiểu điều này thật nhanh, vì vậy hãy

hãy xem chúng tôi sẽ tạo một hình ảnh

logo Firebase Dash và các bạn dành cho những người bạn

chưa biết đây rõ ràng

chỉ là tên của bất kỳ thứ gì có trong

thư mục nội dung của bạn, chúng tôi cần tạo tỷ lệ chấm

có thể thay đổi

kích thước dấu chấm này để lấp đầy

và đó là vẫn sẽ trông rất tệ,

vì vậy chúng tôi cần cung cấp cho nó một khung có chiều rộng

và chiều cao là 100 x 120.

và điều đó trông rất tuyệt và sau đó

chúng tôi sẽ nói phần đệm chấm chấm

dọc

32 pixel được,

thật tuyệt vời và tại dưới cùng của tất cả



**Main subtitle**

guys in between sign in and sign up

let's go ahead and add a spacer  
and that's just going to move this up to  
the top which is exactly what we want

now we're going to create these form  
Fields guys so typically what I do is I

will I would create this like the bad

way the Newbie way of like copy and  
pasting pretty much the same code  
um for these form fields and you guys

notice they're extremely similar the  
only things that differ between them are  
the titles and the actual text that gets  
passed in so instead of doing it the bad  
way first and refactoring it I'm just  
going to skip straight to the good way  
so we're going to create a reusable

component for these input views so we're

going to go up to our components folder  
here and create a new file and it's  
going to be a swift UI View and we're

going to call this input View  
and having this as a reusable component

is going to help us out a lot guys  
because especially as we get to the  
registration page and we see four of  
these fields we don't want to copy and

paste pretty much the same code four  
times we're going to be able to reuse

**Second subtitle**

những người này ở giữa đăng nhập và  
đăng ký,

hãy tiếp tục và thêm một dấu cách  
và điều đó sẽ di chuyển cái này lên  
trên cùng, đó chính xác là những gì chúng  
ta muốn

bây giờ chúng ta sẽ tạo các

Trường biểu mẫu này, những người này  
thường như thế nào Tôi làm là tôi

sẽ Tôi sẽ tạo cái này giống như cách tệ hại  
mà

cách của Người mới giống như sao chép và  
dán gần như cùng một đoạn

mã cho các trường biểu mẫu này và các  
bạn

để ý rằng chúng cực kỳ giống nhau,

điều khác biệt duy nhất giữa chúng là

tiêu đề và văn bản thực tế được

chuyển vào nên thay vì thực hiện theo

cách xấu trước và tái cấu trúc nó, tôi

sẽ chuyển thẳng sang cách tốt, vì vậy

chúng tôi sẽ tạo một thành phần có thể tái  
sử dụng

cho các chế độ xem đầu vào này, vì vậy  
chúng tôi sẽ

sẽ đi tới thư mục thành phần của chúng tôi

ở đây và tạo một tệp mới và nó

sẽ là Chế độ xem giao diện người dùng  
nhẹ và chúng tôi

sẽ gọi Chế độ xem đầu vào này

và việc sử dụng tệp này như một thành  
phần có thể tái sử dụng

sẽ giúp chúng tôi rất nhiều

vì đặc biệt là khi chúng tôi đến

trang đăng ký và chúng tôi thấy bốn trong số

các trường này mà chúng tôi không muốn  
sao chép và

dán khá nhiều mã giống nhau bốn

lần, chúng tôi sẽ có thể sử dụng lại Chế

**Main subtitle**

this input View and make our code way

more efficient and we're just going to  
set it up so that we can pass different  
things into it when we create it you

guys have probably probably seen me do  
stuff like this before unless you're new

to the channel so it's also going to  
help us here so anyway let's go ahead

and create this input view we need to

add a couple input or initialization

properties first we need a binding  
VAR text which is oops

which is a string

title which is a string let placeholder  
which is also a string

and we're going to say VAR is secure

field and default it to false so we're  
going to see how we're going to use all  
of these in just a little bit let's go

and create the actual view now  
so we are going to have a v stack guys  
and let's actually go ahead and give it

the input parameters that it needs  
so just go down to your preview struct

and hit fix text you can say dot

constant and just make it blank  
title is email address

and placeholder can be name  
at example

**Second subtitle**

độ xem đầu vào này và làm cho mã của  
chúng tôi

hiệu quả hơn và chúng tôi sẽ  
thiết lập nó để chúng tôi có thể chuyển  
những thứ khác nhau vào đó khi chúng tôi  
tạo ra nó,

có lẽ các bạn đã từng thấy tôi làm  
những việc như thế này trước đây trừ khi  
bạn là người mới

của kênh nên nó cũng sẽ  
giúp ích cho chúng tôi ở đây, vì vậy, dù sao  
đi nữa, hãy tiếp tục

và tạo chế độ xem đầu vào này, trước tiên  
chúng ta cần

thêm một vài thuộc tính đầu vào hoặc thuộc  
tính khởi tạo,

chúng ta cần một

văn bản VAR ràng buộc, đó là rất tiếc, đó là  
một

tiêu đề chuỗi là một trình giữ chỗ cho chuỗi  
cũng là một chuỗi

và chúng ta' Tôi sẽ nói VAR là trường an  
toàn

và mặc định nó là sai, vì vậy chúng ta  
sẽ xem cách chúng ta sẽ sử dụng tất  
cả những thứ này chỉ trong chốc lát, hãy bắt  
đầu

và tạo chế độ xem thực tế ngay bây giờ  
để chúng ta sẽ có một v stack guys

và chúng ta hãy thực sự tiếp tục và cung  
cấp cho nó

các tham số đầu vào mà nó cần,

vì vậy chỉ cần đi xuống cấu trúc xem trước  
của bạn

và nhấn văn bản sửa lỗi, bạn có thể nói  
hàng số dấu

chấm và chỉ cần để trống tiêu

đề là địa chỉ email

và trình giữ chỗ có thể là tên

chẳng hạn

**Main subtitle**

.com

you spell that right okay so we got all

of our input params there

so this is basically what it's going to

look like when we create an input view

guys we're going to pass in some text

parameter in our case it'll be a state

property later the title is what you see

here for the input View and the

placeholder is what you're going to see

here and we'll get into this is secure

guy in a little bit so we're going to

say text

and we're just going to pass in the

title

so in this case it would be email

in this case it would be password

and we're going to say dot foreground

color is color

dot dark gray

dot font weight dot

semibolds.font is dot footnote okay

and here's where we create the actual

text field so we're going to say if is

secure fields and else

so basically guys we want to be able to

use this for a password field as well so

if we have this secure Field property

set to true it's going to create a

password or secure field for us if not

**Second subtitle**

. com

bạn đánh vần đúng rồi đó, vì vậy chúng tôi có tất

cả các thông số đầu vào của mình ở đó,

vì vậy về cơ bản, nó sẽ

trông như thế nào khi chúng tôi tạo một chế độ xem đầu vào,

chúng tôi sẽ chuyển một số

tham số văn bản trong trường hợp của chúng tôi, nó sẽ là một thuộc tính trạng thái

sau đó, tiêu đề là những gì bạn thấy

ở đây cho Chế độ xem đầu vào và

trình giữ chỗ là những gì bạn sẽ thấy ở

đây và chúng ta sẽ tìm hiểu về anh chàng an toàn này

một chút vì vậy chúng ta sẽ

nói văn bản

và chúng ta 'Chúng tôi sẽ chuyển vào

tiêu đề

vì vậy trong trường hợp này nó sẽ là email

trong trường hợp này nó sẽ là mật khẩu

và chúng tôi sẽ nói màu nền trước của dấu chấm

là màu chấm màu

xám đậm dấu

chấm phong chữ trọng lượng dấu chấm phẩy.

và đây là nơi chúng tôi tạo

trường văn bản thực tế, vì vậy chúng tôi sẽ nói liệu có phải là

trường bảo mật hay không và

về cơ bản, chúng tôi muốn có thể

sử dụng trường này cho trường mật khẩu, vì vậy

nếu chúng tôi đặt thuộc tính Trường bảo mật này

thành true nó sẽ tạo một

mật khẩu hoặc trường bảo mật cho chúng tôi nếu không

**Main subtitle**

it's just going to be a regular old text

field so we're going to say secure field  
this is going to be our placeholder  
and this is going to be our text  
and we're going to say dot font dot  
system size of 14.

and we can pretty much copy and paste  
this down in the else block and just  
make it a text field and these  
properties are going to be the exact  
same  
and lastly we need a divider

and then we need to adjust some  
things about our v-stack we need to make

the alignment dot leading to align this

all to the Leading Edge and we are also  
going to make the spacing 12 pixels okay  
so that looks really good right there  
don't worry about the padding just yet  
we're going to handle all of that in the

implementation view or the login view so

now that we have this awesome reusable

component and I reuse this all the time  
in my projects guys it's a really nice  
clean input field with a title so we can

go back to login View

and we can now

um first off let's create two State

Properties

for email and password

**Second subtitle**

nó sẽ chỉ là một trường văn bản cũ thông  
thường,

vì vậy chúng tôi sẽ nói rằng trường bảo mật,  
đây sẽ là trình giữ chỗ của chúng tôi  
và đây sẽ là văn bản của chúng tôi  
và chúng tôi' sẽ nói

kích thước hệ thống chấm phông chữ chấm  
là 14.

và chúng ta có thể sao chép và dán  
phần này xuống khối khác và chỉ cần  
biến nó thành trường văn bản và  
các thuộc tính này sẽ giống hệt  
nhau

và cuối cùng chúng ta cần một dấu phân  
cách

và sau đó, chúng tôi cần điều chỉnh một số  
thứ về ngăn xếp v của mình, chúng tôi cần  
tạo

dấu chấm căn chỉnh dẫn đến căn chỉnh tất  
cả những thứ này

với Cạnh hàng đầu và chúng tôi cũng  
sẽ tạo khoảng cách 12 pixel ổn

để nó trông thực sự đẹp phải

không lo lắng về phần đệm nhưng

chúng tôi sẽ xử lý tất cả những thứ đó trong  
chế độ

xem triển khai hoặc chế độ xem đăng nhập  
để

bây giờ chúng tôi có thành phần có thể tái  
sử dụng tuyệt vời này

và tôi luôn sử dụng lại thành phần này

trong các dự án của mình, đó là một

trường nhập liệu sạch đẹp thực sự với một  
tiêu đề để chúng tôi có thể

quay lại Chế độ xem đăng nhập

và bây giờ chúng tôi có thể

ừm, trước tiên hãy tạo hai Thuộc tính trạng  
thái

cho email và mật khẩu,

**Main subtitle**

cool now we are going to on this form

Field section create a v-stack

with spacing of 24 pixels and we're

going to say input View

text is going to be dollar sign email

title is going to be email address

and the placeholder is going to be name

at example.com

and let me clean this up a little bit

cool

and we're also going to set another

property here guys we're going to say

dot Auto capitalization

is going to be dot none

and that's because it's an email field

and we don't want it to automatically

capitalize things because email

addresses shouldn't have capital letters

okay so once again don't worry about the

padding just yet we're going to fix that

in just a second now I'm going to create

my input View

for my password field so the text is

going to be dollar sign password so I

hope it's clear how that's working with

the state and binding properties guys if

that's confusing to you once again guess

what I got a tutorial for that the link

is going to be in the description if you

are confused about the differences

between State properties and binding

properties so we have a state property

in the main View and a binding property

**Second subtitle**

bây giờ chúng tôi sẽ chuyển sang phần  
Trường của biểu mẫu này

tạo một ngăn xếp v

với khoảng cách là 24 pixel và chúng tôi 'sẽ

nói đầu vào Chế độ xem

văn bản sẽ là ký hiệu đô la,

tiêu đề email sẽ là địa chỉ email

và trình giữ chỗ sẽ là tên

tại example.com

và hãy để tôi làm sạch điều này một chút

và chúng ta cũng sẽ để đặt một

thuộc tính khác ở đây các bạn, chúng tôi sẽ  
nói

dấu chấm Viết hoa tự động

sẽ là dấu chấm không

và đó là vì đó là trường email

và chúng tôi không muốn nó tự động

viết hoa mọi thứ vì

địa chỉ email không được viết

hoa. một lần nữa, đừng lo lắng về

phần đệm, chúng tôi sẽ khắc phục điều đó

chỉ trong giây lát, tôi sẽ tạo

Chế độ xem đầu vào

cho trường mật khẩu của mình để văn bản

sẽ là mật khẩu ký hiệu đô la, vì vậy tôi

hy vọng nó làm rõ cách nó hoạt động với  
các

thuộc tính trạng thái và ràng buộc nếu

điều đó làm bạn khó hiểu một lần nữa hãy  
đoán xem

tôi có hướng dẫn gì về liên kết đó

sẽ có trong phần mô tả nếu bạn

bối rối về sự khác biệt

giữa các thuộc tính Trạng thái và thuộc tính  
ràng buộc,

vì vậy chúng tôi có một thuộc tính trạng thái

trong Chế độ xem chính và một thuộc tính

**Main subtitle**

on the child view I'm going to call it  
 or the the custom view that we just  
 created and we pass those along from the  
 main view here which is our login view  
 as State Properties into the input view  
 which is a binding property  
 uh so now our Pat our title is going to  
 be password  
 and this guy is going to say enter your  
 password  
 so we can see how this nice reusable  
 component makes our life way easier  
 right instead of copy and pasting all of  
 this code that we see here  
 because all of this is pretty much going  
 to be the same the only thing that's  
 going to be different is the title and  
 the text parameter and whether or not  
 it's it's a secure field speaking of we  
 need to go ahead and add that here so  
 you guys will notice that it comes up  
 and we can say true and the reason that  
 we don't need to pass it in here is  
 because we gave it a default value of  
 false which is a nice little trick right  
 because you don't want to have to say  
 false every time you just want to create  
 a text field so we're just defaulting it  
 to false and then if we ever want it to

**Second subtitle**

ràng buộc  
 trên chế độ xem con. Tôi sẽ gọi nó  
 hoặc chế độ xem tùy chỉnh mà chúng tôi  
 vừa  
 tạo và chúng tôi chuyển những thuộc tính  
 đó từ  
 chế độ xem chính ở đây là chế độ xem đăng  
 nhập của chúng tôi  
 dưới dạng Thuộc tính trạng thái vào chế độ  
 xem đầu vào,  
 đây là một thuộc tính ràng buộc  
 uh, vậy bây giờ Pat của chúng ta, tiêu đề  
 của chúng ta sẽ  
 là mật khẩu  
 và anh chàng này sẽ nói nhập mật khẩu của  
 bạn  
 để chúng ta có thể thấy thành phần có thể  
 tái sử dụng tuyệt vời này  
 giúp cuộc sống của chúng ta dễ dàng hơn  
 thay vì sao chép và dán tất cả  
 mã này mà chúng ta thấy ở đây  
 bởi vì tất cả những thứ này gần như  
 giống nhau, điều duy nhất  
 sẽ khác là tiêu đề và  
 tham số văn bản và liệu  
 đó có phải là trường bảo mật hay không mà  
 chúng ta  
 cần hãy tiếp tục và thêm nó vào đây để  
 các bạn sẽ nhận thấy rằng nó xuất hiện  
 và chúng ta có thể nói đúng và lý do mà  
 chúng ta không cần chuyển nó vào đây là  
 vì chúng ta đã đặt cho nó một giá trị mặc  
 định là  
 sai, đây là một mẹo nhỏ hay đúng  
 vì bạn không muốn phải nói  
 sai mỗi khi bạn chỉ muốn tạo  
 một trường văn bản, vì vậy chúng tôi chỉ đặt  
 mặc định  
 là sai và sau đó nếu chúng tôi muốn nó

**Main subtitle**

be a secure field in the case of  
creating a password field we can then  
specify that we want it to be true  
that's a nice trick about having default  
values and what I mean by default value

guys is we said equals false so it's  
always going to default to false unless  
we set we specify otherwise okay  
so now we need to adjust the padding so

I'm going to say dot padding  
is dot horizontal  
so that looks really good and then we  
are also going to give it dot padding of

dot top of like 12 pixels just to get a  
little bit more space between that and  
our logo image we are not going to be

doing the forgot password option here  
guys that will be a part of the  
completed source code we are just going  
to skip straight to the sign in and sign

up buttons so  
uh let's go ahead and create this button  
now

so I'm going to create a button  
and it's going to be action and label  
so here we could say print  
log user in  
and my label  
is going to be an H stack and it's going  
to have text sign in  
dot font weight dot semi bold  
and an image of system name  
arrow.right

**Second subtitle**

là một trường an toàn trong trường hợp  
tạo mật khẩu Sau đó, chúng tôi có thể  
chỉ định rằng chúng tôi muốn nó là đúng,  
đó là một mẹo hay về việc có  
các giá trị mặc định và ý tôi là theo giá trị  
mặc định,  
chúng tôi đã nói bằng sai, vì vậy nó  
luôn luôn mặc định là sai trừ khi  
chúng tôi đặt chúng tôi chỉ định khác,  
vậy bây giờ chúng ta cần điều chỉnh phần  
đệm, vì vậy  
tôi sẽ nói rằng phần đệm chấm  
là chấm nằm ngang  
để nó trông thực sự đẹp và sau đó chúng tôi  
cũng sẽ cung cấp cho nó phần đệm chấm  
của

đỉnh có kích thước 12 pixel chỉ để có  
thêm một chút khoảng trống giữa cái đó và  
hình ảnh biểu trưng của chúng tôi, chúng tôi  
sẽ không

thực hiện tùy chọn quên mật khẩu ở đây,  
đây sẽ là một phần của  
mã nguồn đã hoàn thành, chúng tôi sẽ  
chuyển thẳng đến các nút đăng nhập và  
đăng

ký, vì vậy,  
hãy tiếp tục và tạo nút này  
ngay bây giờ  
vì vậy tôi sẽ tạo một nút  
và nó sẽ là hành động và nhấn  
nên ở đây chúng tôi có thể nói rằng  
người dùng đăng nhập bản in  
và nhấn của tôi  
sẽ là một ngăn xếp H và nó sẽ  
có ký hiệu văn bản ở phong chữ dấu chấm.  
chấm nửa đậm  
và một hình ảnh của tên hệ thống  
arrow.right

**Main subtitle**

and on this guy we are going to give

our let's see yeah the button itself

we're going to say dot foreground color

is dot White

and Dot frame is

width and height of UI screen

dot main dot bounds

dot width

minus 32 so that's just going to make it

whatever the width of the screen is

minus 32 pixels so it'll have 16 pixels

of padding on both sides and the height

is going to be 48.

um sorry guys this I think is going to

go on to the H stack itself yeah and

then the button here is going to have a

background of color

dot system blue

and then a corner radius of 10.

and let me just okay so something is

wrong

oop I forgot a parenthesis there we go

okay so that looks good we just need

some padding now right

so let's see

um

how are we going to do that padding I

think we can just say padding dot top

and let's make it like 24 pixels

yep that looks good

and now we need to blow our spacer

create our sign up button so this is

actually going to be a navigation link

guys

**Second subtitle**

và trên anh chàng này, chúng ta sẽ cung cấp cho

chúng ta hãy xem vâng, chính nút

chúng ta sẽ nói màu nền trước của chấm

là chấm Trắng

và Khung chấm là chiều

rộng và chiều cao của màn hình giao diện người dùng

dot chính dot giới hạn

dot width

trừ 32, vì vậy điều đó sẽ làm cho nó

bất kể chiều rộng của màn hình là

trừ 32 pixel, vì vậy nó sẽ có 16 pixel

đệm ở cả hai bên và chiều cao

sẽ là 48.

ừm, xin lỗi các bạn điều này Tôi nghĩ rằng nó sẽ

tự chuyển sang ngăn xếp H vâng và

sau đó nút ở đây sẽ có

nền là

hệ thống chấm màu xanh lam

và sau đó là bán kính góc là 10.

và hãy để tôi ổn vì vậy đã xảy ra sự

cố rất tiếc

Tôi đã quên dấu ngoặc đơn ở đó, chúng ta sẽ ổn thôi

vì vậy có vẻ tốt, chúng ta chỉ cần

một số phần đệm ngay bây giờ, vì vậy

hãy xem chúng ta sẽ thực hiện phần đệm đó như thế nào.

chúng ta cần thổi miếng đệm

tạo nút đăng ký để đây

thực sự sẽ là một liên kết điều hướng nhé các bạn



**Main subtitle**

and let's see navigation link and which option are we going to use I think we're going to just use destination and label and you guys might be saying like hey this doesn't follow along with the new navigation stack API and that's because this is not this doesn't need to be a value-based navigation setup right it's just we're just going from one place to another we're not clicking on like an object that we need to create a navigation destination for so navigation view has been deprecated but this is not so there's a reason Swift UI left this in place and if that's because it's still viable to use this as an option so before anybody starts freaking out and saying that I'm not doing this the right way this is totally okay to do it this way so we haven't set up our registration view yet but we are going to create this label here so let's see it's going to be don't have an account sign up so we're going to say h stack texts don't have an account and then it's just going to be text after that sign up okay and it's going to be dot font

**Second subtitle**

và hãy xem liên kết điều hướng và chúng ta sẽ sử dụng tùy chọn nào. Tôi nghĩ chúng ta sẽ chỉ sử dụng điểm đến và nhãn và các bạn có thể nói như này, điều này không tuân theo API ngăn xếp điều hướng mới và đó là bởi vì đây không phải là điều này không cần phải là một thiết lập điều hướng dựa trên giá trị, chỉ là chúng ta chỉ đi từ nơi này sang nơi khác mà thôi. 'không nhấp vào giống như một đối tượng mà chúng tôi cần tạo đích điều hướng cho nên chế độ xem điều hướng không được dùng nữa nhưng đây không phải là lý do Swift UI để nguyên điều này và nếu đó là vì vẫn có thể sử dụng điều này như một tùy chọn Vì vậy, trước khi bất kỳ ai bắt đầu lo lắng và nói rằng tôi đang làm điều này không đúng cách thì bạn hoàn toàn có thể làm theo cách này vì vậy chúng tôi chưa thiết lập chế độ xem đăng ký của mình nhưng chúng tôi sẽ tạo nhãn này ở đây, vì vậy hãy cùng xem nó sẽ không có tài khoản đăng ký, vì vậy chúng tôi sẽ nói h ngăn xếp văn bản không có tài khoản và sau đó nó sẽ chỉ là văn bản sau khi đăng ký được rồi và nó sẽ là

**Main subtitle**

weight dot bold

and it's going to be dot font

of dot system size 14.

so that looks pretty good right

and we can actually make you see that

that's a little bit too much spacing

there guys we can adjust the spacing of

our components to two pixels and it's

going to bring that in nicely you could

even make it like three

just to play around with it

and my previews now being mean to me

let's see if it loads

yep so that looks pretty good three

Paces uh yep okay cool

all right so now you guys will notice

that that will allow me to go to a new

screen and in a little bit we're going

to implement that registration view that

we see right here but this is looking

really really good guys

um we're also going to implement this

form validation functionality guys so

like

you'll see test gmail.com

and then when my form becomes valid it's

going to actually enable the button and

change the color of it we're going to do

all that stuff as well but next up we

are going to get started with uh writing

**Second subtitle**

dấu chấm phong chữ dấu chấm in đậm

và nó sẽ là phong chữ chấm

có kích thước hệ thống chấm 14.

Vì vậy, nó trông khá đẹp phải không

và chúng tôi thực sự có thể làm cho bạn  
thấy rằng

khoảng cách hơi quá nhiều

ở đó, các bạn, chúng tôi có thể điều chỉnh  
khoảng cách của

các thành phần thành hai pixel và đó là

sẽ mang nó vào một cách độc đáo, bạn

thậm chí có thể làm cho nó giống như ba

chỉ để chơi với nó

và các bản xem trước của tôi bây giờ có ý  
nghĩa với tôi,

hãy xem nó có tải

không, vâng, vì vậy trông khá tốt.

sẽ nhận thấy

rằng điều đó sẽ cho phép tôi chuyển sang  
một

màn hình mới và một chút nữa chúng ta sẽ

triển khai chế độ xem đăng ký mà

chúng ta thấy ngay tại đây nhưng điều này  
trông

thực sự rất tuyệt đấy các bạn

ừm chúng ta cũng sẽ triển khai

biểu mẫu này những người có chức năng  
xác nhận giống

như

bạn sẽ thấy gmail.com thử nghiệm

và sau đó khi biểu mẫu của tôi trở nên hợp  
lệ, nó

sẽ thực sự kích hoạt nút và

thay đổi màu sắc của nút, chúng tôi cũng sẽ  
thực hiện

tất cả những nội dung đó nhưng tiếp theo  
chúng tôi

sẽ thực hiện để bắt đầu với việc viết

**Main subtitle**

out the code to build our registration

View and this is going to go really  
really quick so let's just go ahead and  
get it done now  
and the reason that's going to go really  
quick is because we pretty much have the  
same view that this is in our login page

and we built it so that it's scalable so

that it's easier it's easy for us to  
make that with not a lot of code it's

really efficient right so we're going to  
say

registration View

and we are just going to copy over this  
image

so grab the v-stack as well or let's  
actually just place it in a v-stack

and

grab the image

paste that right there

let's go ahead and grab this v-stack  
paste that there

create our properties so we're going to

say State private VAR email equals that

just an empty string two three four  
we're going to do full name  
full name

password and then we are also going to  
do this awesome confirm password feature

okay and then

we can copy this guy

**Second subtitle**

mã để xây dựng Chế độ xem đăng ký của  
chúng ta

và quá trình này sẽ diễn ra rất

nhanh, vì vậy hãy tiếp tục và

hoàn thành nó ngay bây giờ

và lý do sẽ diễn ra rất

nhanh là vì chúng ta có khá nhiều thứ

giống nhau xem rằng đây là trang đăng  
nhập của chúng tôi

và chúng tôi đã xây dựng nó sao cho nó có  
thể mở rộng để

chúng tôi dễ dàng

thực hiện điều đó mà không cần nhiều mã,  
nó

thực sự hiệu quả, vì vậy chúng tôi sẽ  
nói

đăng ký Xem

và chúng tôi chỉ sẽ sao chép

hình ảnh này,

vì vậy hãy lấy cả v-stack hoặc

thực sự chúng ta chỉ cần đặt nó vào một v-  
stack

và

lấy hình ảnh

dán ngay tại đó,

hãy tiếp tục và lấy v-stack

paste này để

tạo các thuộc tính của chúng ta để chúng ta  
'sẽ

nói rằng email VAR riêng tư của Bang  
tương đương với

chỉ một chuỗi trống hai ba bốn

chúng ta sẽ thực hiện tên đầy đủ

tên đầy đủ

mật khẩu và sau đó chúng ta cũng sẽ

thực hiện tính năng xác nhận mật khẩu  
tuyệt vời này

được rồi

chúng ta có thể sao chép anh chàng này

**Main subtitle**

place it

right here

this is going to say

full name

placeholder is going to be enter

your name

and we definitely want Auto

capitalization for that right guys

unlike email right because we want the

person's name to be capitalized

correctly so we're going to delete that

and then here we're just going to copy

our password guy

and we are going to change this to

confirm password

still going to be a secure field this is

going to say confirm your password and

this is going to be confirm password

yep and we just need to also change this

guy to full name right so just take a

second and make sure that your field

looks like this input view email input

view full name input view password input

view dollar sign confirm password

beautiful and then guess what we can

copy and paste this button

and you guys could make this button

reusable as well you could create like a

subclass for this button like much like

you did with the input view

um I'm just not going to do that in the

interest of time I've already showed you

guys how to create a reusable component

**Second subtitle**

đặt nó

ngay tại đây,

điều này sẽ cho biết

trình giữ chỗ tên đầy đủ sẽ được nhập

tên của bạn

và chúng tôi chắc chắn muốn

Viết hoa tự động cho những người phù hợp đó

không giống như email bên phải vì chúng tôi muốn

tên của người đó được viết hoa

chính xác nên chúng tôi sẽ xóa nó

và sau đó ở đây, chúng tôi sẽ sao chép

anh chàng mặt khẩu của mình

và chúng tôi sẽ thay đổi điều này để

xác nhận mặt khẩu

vẫn sẽ là một trường an toàn, điều này sẽ

nói xác nhận mặt khẩu của bạn và

đây sẽ là xác nhận mặt khẩu

vâng và chúng tôi chỉ cũng cần thay đổi

anh chàng này thành tên đầy đủ, vì vậy chỉ cần dành một

chút thời gian và đảm bảo rằng trường của bạn

trông giống như thế này.

nút này

và các bạn có thể làm cho nút này

có thể tái sử dụng cũng như bạn có thể tạo giống như một

lớp con cho nút này giống như

bạn đã làm với chế độ xem đầu vào

ừm. Tôi sẽ không làm điều đó vì

lợi ích của thời gian mà tôi đã trình bày các

bạn làm thế nào để tạo một thành phần có

**Main subtitle**

but and honestly this isn't too bad to

to just copy and paste it's not a lot of  
code whereas this would have been way  
too much code to copy and paste so  
having this input view is really nice

but anyway this is going to say sign

user

up

and then

this guy is going to say sign up

uh frame and everything is the same  
background and everything is the same  
uh let's see go ahead and rebuild that

just to make sure we're good to go here

and yep sign up then we add our spacer

and this guy down here at the bottom  
guys this is not going to be a  
navigation link we just need to make  
sure that this pops our navigation back

to the original screen so what we can do

for that is use this handy environment

property

so you say backslash dot dismiss VAR

dismiss

and then we can go down here and create

a button

with an action and a label

the action is going to be dismiss

**Second subtitle**

thể tái sử dụng

nhưng thành thật mà nói, điều này không  
quá tệ khi

chỉ sao chép và dán nó không có nhiều  
mã trong khi điều này sẽ có  
quá nhiều mã để sao chép và dán nên  
có chế độ xem đầu vào này thực sự rất  
tuyệt

nhưng dù sao thì điều này sẽ nói người  
dùng đăng ký

và sau đó

anh chàng này sẽ nói đăng ký

uh khung và mọi thứ đều giống nhau  
và mọi thứ đều giống nhau

uh chúng ta hãy tiếp tục và xây dựng lại  
điều đó

chỉ để đảm bảo rằng chúng ta ổn hãy vào  
đây

và vâng, đăng ký, sau đó chúng tôi thêm  
miếng đệm của mình

và anh chàng này ở dưới đây, ở phía dưới,  
đây sẽ không phải là một

liên kết điều hướng, chúng tôi chỉ cần

đảm bảo rằng điều này sẽ bật điều hướng  
của chúng tôi trở

lại màn hình ban đầu để chúng tôi có thể  
làm gì

vì đó là sử dụng thuộc tính môi trường tiện  
dụng này

để bạn nói dấu gạch chéo ngược dấu chấm  
loại bỏ VAR loại

bỏ

và sau đó chúng ta có thể đi xuống đây và  
tạo

một nút

có hành động và nhãn mà

hành động sẽ bị loại bỏ

**Main subtitle**

and the label we can copy and paste from  
our login View  
right here  
and this is going to say already  
have an account sign in  
okay so now let's actually just go ahead  
and test this out let's see if we can  
  
move back and forth between our views so  
  
here in my navigation link we want the  
  
destination to be my registration view  
so let's let our preview load and wait  
  
for the Moment of Truth so we're going  
  
to hit sign up and check that out guys  
it goes right over to our registration  
  
View and then we can hit sign in here to  
  
go back to our login view there is one  
  
cleanup we have to make though I don't  
  
like this back button here because I  
like the navigation bar to be hidden as  
we can see in the completed app here so  
  
that the transition looks smoother right  
  
the logo image is in the exact same  
place on both screens and our back  
  
button serves as this little footer  
button down here whereas if you have  
that back button it shoves everything  
down we don't really need the navigation

**Second subtitle**

và nhãn chúng ta có thể sao chép và dán từ  
đăng nhập Xem  
ngay tại đây  
và điều này có nghĩa là đã  
có tài khoản đăng nhập được,  
vì vậy bây giờ chúng ta hãy thực sự tiếp tục  
và kiểm tra điều này, hãy xem liệu chúng ta  
có thể  
di chuyển qua lại giữa các chế độ xem của  
mình không vì vậy  
ở đây trong liên kết điều hướng của tôi,  
chúng ta muốn  
điểm đến là chế độ xem đăng ký của tôi,  
vì vậy hãy để bản xem trước của chúng tôi  
tải và đợi  
Khoảnh khắc của sự thật, vì vậy chúng tôi  
sẽ  
nhấn đăng ký và kiểm tra xem các bạn,  
nó sẽ chuyển ngay đến Chế độ xem đăng  
ký của chúng tôi  
và sau đó chúng tôi có thể nhấn đăng nhập  
tại đây để  
tiếp tục quay lại chế độ xem đăng nhập của  
chúng tôi, có một  
công việc dọn dẹp mà chúng tôi phải thực  
hiện mặc dù tôi không  
thích nút quay lại này ở đây vì tôi  
muốn ẩn thanh điều hướng như  
chúng ta có thể thấy trong ứng dụng đã  
hoàn thành ở đây  
để quá trình chuyển đổi trông mượt mà hơn  
ngay  
trên logo hình ảnh ở cùng một  
vị trí trên cả hai màn hình và nút quay lại  
của chúng tôi  
đóng vai trò là nút chân trang nhỏ này  
ở đây trong khi nếu bạn có  
nút quay lại đó, nó sẽ đẩy mọi thứ  
xuống, chúng tôi không thực sự cần thanh  
điều hướng

**Main subtitle**

bar up at the top  
so you can just go here  
inside of your login view when we create  
  
the registration view as the destination  
and say dot navigation  
bar back button hidden  
  
and does that require an input parameter  
back button hidden  
um I don't know if we have to pass in  
  
like true or something there yes we do  
so yes we want to say true  
and now let's let this build  
and you guys notice that that looks way  
cleaner to me at least you guys can keep  
  
the back button in there if you want I  
just prefer it like this so that pretty  
much wraps it up for the authentication  
flow UI next up I want us to get started  
with creating this profile user  
interface and actually populating it  
with like some user data so we're going  
  
to create a user model create our  
profile view and then we're going to  
  
move on to all of the networking stuff  
with our view model and all that so  
  
let's go ahead and jump right into it  
guys let's go ahead and go to this  
profile folder and create a swift UI  
  
view we're just going to call this  
profile view

**Second subtitle**

ở trên cùng  
nên bạn có thể chỉ cần vào đây  
bên trong chế độ xem đăng nhập của bạn  
khi chúng tôi tạo chế  
độ xem đăng ký làm đích  
và nói rằng  
nút quay lại của thanh điều hướng dấu  
chấm bị ẩn  
và điều đó có yêu cầu  
nút quay lại thông số đầu vào bị ẩn  
không tôi không biết liệu chúng tôi có phải  
chuyển vào  
như đúng hay không vâng, chúng tôi làm  
như vậy, vâng, chúng tôi muốn nói sự thật  
và bây giờ hãy để bản dựng này  
và các bạn lưu ý rằng nó trông  
gọn gàng hơn đối với tôi ít nhất các bạn có  
thể giữ  
nút quay lại ở đó nếu bạn muốn. Tôi  
chỉ thích nó như thế này để  
  
Tôi muốn chúng ta bắt đầu  
với việc tạo giao diện người dùng hồ sơ này  
và thực sự phổ biến nó  
giống như một số dữ liệu người dùng, vì vậy  
chúng tôi sẽ  
tạo một mô hình người dùng, tạo chế độ  
xem hồ sơ của chúng tôi và sau đó chúng  
tôi Chúng ta sẽ  
chuyển sang tất cả nội dung mạng  
với mô hình chế độ xem của chúng ta và tất  
cả những thứ đó, vì vậy  
hãy tiếp tục và bắt đầu ngay với nó.  
Hãy tiếp tục và chuyển đến  
thư mục hồ sơ này và tạo chế độ xem giao  
diện người dùng nhanh chóng mà  
chúng ta sẽ thực hiện gọi chế độ  
xem hồ sơ này

**Main subtitle**

and let's get our preview up and running

and take a look at how we're going to do this so let's wait for our preview to

load but pretty much guys this is all just going to be inside of a list and each one of these is going to be its own individual section with its own individual set of options so let's go ahead and just start with the header so I'm going to say this is going to be a list

and we are going to create a section here

and we're going to actually create two more sections so this one is going to be called General

oops

and another section for accounts okay so this one does not have a title because it's just like the user info

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

create that now and then we'll create

each one of these individual settings

items so we're going to say text and we can say MJ for Michael Jordan

Undisputed goat if you ask me

and then we'll say dot font is dot title

dot font weight Dot semibold

dot foreground colors dot White

dot frame of a width and height of 72 by 72

**Second subtitle**

và hãy thiết lập và chạy bản xem trước của chúng tôi

và xem cách chúng tôi sẽ thực hiện

điều này, vì vậy hãy đợi bản xem trước của chúng tôi

tải nhưng hầu hết mọi người, tất cả điều này sẽ nằm trong một danh sách và mỗi một trong số này sẽ là phần riêng lẻ với

bộ tùy chọn riêng, vì vậy hãy tiếp tục và chỉ bắt đầu với tiêu đề,

vì vậy tôi sẽ nói đây sẽ là

một danh sách

và chúng ta sẽ tạo một phần

ở đây

và chúng ta sẽ thực sự tạo thêm hai

phần nữa, vì vậy phần này sẽ được

gọi là Chung

Rất tiếc

và một phần khác dành cho tài khoản

được rồi, vì vậy phần này không có tiêu đề

vì nó giống như chế độ xem thông tin người dùng,

vì vậy hãy tiếp tục và xem nếu chúng tôi có thể

tạo nó ngay bây giờ và sau đó chúng tôi sẽ tạo

từng mục trong số các mục cài đặt riêng lẻ này,

vì vậy chúng tôi sẽ nói văn bản và chúng tôi

có thể nói MJ cho Michael Jordan

Con dê không thể tranh cãi nếu bạn hỏi tôi

và sau đó chúng tôi sẽ nói phong chữ dấu chấm là chấm tiêu đề chấm

trọng lượng phong chữ Dấu chấm phẩy chấm

màu nền trước chấm Khung chấm trắng

có chiều rộng và chiều cao là 72 x

72



**Main subtitle**

and then we're going to give it a  
background color

of dot system gray three

and then

just give it a clip shape guys of a

circle

and that's pretty clean right there yeah

that looks awesome and then

so this actually needs to be inside of

an H stack guys

because we see here that we have this

horizontally stacked set of components

so let's just go ahead and command click

on that hold down command

and click on it and say embed in h-stack

and next up we are going to create a

v-stack and that's because these two

text components are stacked vertically

so we have an H stack and within that we

have a v-stack

so I'm going to say uh text

Michael Jordan

and it's going to be dot font weight dot

semi bold dot padding

dot top of four

cool and then text

test at gmail.com

.font.footnote

let me see and then we can say dot

accent color

is dot gray and once again guys you have

to specify that because when Swift UI is

**Second subtitle**

và sau đó chúng ta sẽ cung cấp cho nó  
màu nền

của hệ thống chấm ba màu xám

và sau đó

chỉ cần tạo cho nó một hình dạng chiếc kẹp.  
một

vòng tròn

và khá rõ ràng ngay tại đó, vâng,

nó trông tuyệt vời và

vì vậy, cái này thực sự cần phải ở bên trong

ngăn xếp H

bởi vì chúng tôi thấy ở đây rằng chúng tôi  
có

tập hợp các thành phần được xếp chồng lên  
nhau theo chiều ngang này,

vì vậy hãy tiếp tục và ra lệnh nhấp

vào nút giữ đó lệnh down

và nhấp vào nó và nói embed in h-stack

và tiếp theo chúng ta sẽ tạo một

v-stack và đó là vì hai

thành phần văn bản này được xếp chồng  
lên nhau theo chiều dọc

nên chúng ta có một ngăn xếp H và bên  
trong đó chúng ta

có một ngăn xếp v

Vì vậy, tôi sẽ nói uh, hãy nhấn tin cho

Michael Jordan

và nó sẽ là dấu chấm, trọng lượng, dấu  
chấm, dấu nửa

đậm, dấu chấm, dấu chấm,

trên cùng là bốn,

tuyệt vời và sau đó

kiểm tra văn bản tại gmail.com

.font.footnote

để tôi xem và sau đó chúng ta có thể nói  
dấu chấm

màu nhấn

là chấm xám và một lần nữa, các bạn

phải chỉ định điều đó bởi vì khi Swift UI

**Main subtitle**

actually smart enough to recognize emails which is pretty cool and it will default it to the accent color of your application so right now the accent color I think defaults to that blue color so if you want to modify that or override it you have to use this accent color modifier that might confuse some people you might think it has to be foreground color which is what I thought until I Googled it but that's why we use accent color there on this v-stack guys we need an alignment of dot leading and a spacing of four pixels so let's go ahead and see how that looks awesome and I think I want to give this a font of Dot subheadline yeah that looks better cool and yeah so that's our sort of our user profile header uh we'll go over how to actually like um populate this with user data in a little bit we haven't created like a data model for that yet we're going to get there later um now we are going to go down here and create these settings row views so uh once again guys we're going to create a reusable component for this so we're

**Second subtitle**

thực sự đủ thông minh để nhận ra email, điều này khá tuyệt và nó sẽ mặc định nó là màu nhấn của ứng dụng của bạn, vì vậy ngay bây giờ tôi nghĩ màu nhấn là mặc định thành màu xanh lam đó, vì vậy nếu bạn muốn sửa đổi hoặc ghi đè lên màu đó, bạn phải sử dụng công cụ sửa đổi màu nhấn này có thể khiến một số người nhầm lẫn, bạn có thể nghĩ rằng đó phải là màu nền trước, đó là điều tôi nghĩ cho đến khi tôi tìm kiếm nó trên Google nhưng đó là lý do tại sao chúng tôi sử dụng màu nhấn ở đó trên v-stack này các bạn, chúng ta cần căn chỉnh các dấu chấm ở đầu và khoảng cách bốn pixel, vì vậy hãy tiếp tục và xem nó trông tuyệt vời như thế nào và tôi nghĩ rằng tôi muốn đặt phong chữ này cho dòng tiêu đề phụ Dot, vâng, nó trông đẹp hơn. và vâng, đó là loại tiêu đề hồ sơ người dùng của chúng tôi, uh chúng ta sẽ tìm hiểu cách thực sự thích um điền dữ liệu người dùng này vào một chút, chúng tôi chưa tạo giống như một mô hình dữ liệu cho điều đó nhưng chúng tôi sẽ đạt được điều đó sau, ừm, bây giờ chúng ta sẽ đi xuống đây và tạo các chế độ xem hàng cài đặt này, vì vậy, một lần nữa, chúng ta sẽ tạo một thành phần có thể tái sử dụng cho việc này, vì vậy chúng ta

**Main subtitle**

going to go up to our components folder and once again that's because these are basically the same thing it's got an icon a title and an optional like thing

at the end of it like some sort of description parameter so we're going to create a reusable component for that

Swift UI view settings row View and once again that's in your components folder and this is going to be pretty simple right so let's start with our input

params we're going to say let image name is a string

title is a string

let tint color

be a color

okay and

let's go ahead and pass these in here so

this is what it's going to look like

when we create it so this one we can

literally just say like gear title

uh version tint color

uh color dot system gray or something like that

and let's just go ahead and wait for our preview to load

cool

um now we can create our actual row View

and use these input params to populate

it with what we want

so it's going to be an age stack spacing of 12 pixels

**Second subtitle**

sẽ đi tới thư mục thành phần của mình và một lần nữa đó là vì những về cơ bản là giống nhau, nó có một biểu tượng, một tiêu đề và một thứ tương tự tùy chọn

ở cuối giống như một số loại tham số mô tả, vì vậy chúng tôi sẽ tạo một thành phần có thể tái sử dụng cho Chế

độ xem giao diện người dùng Swift

đó. thư mục thành phần của bạn

và điều này sẽ khá đơn giản,

vì vậy hãy bắt đầu với các tham số đầu vào của chúng ta,

chúng ta sẽ nói tên hình ảnh

là một chuỗi

tiêu đề là một chuỗi,

hãy để màu sắc là

một màu

được rồi

hãy tiếp tục và chuyển những thứ này vào đây, vì vậy

đây là giao diện

khi chúng ta tạo nó, vì vậy cái này chúng ta có thể

nói theo nghĩa đen như tiêu đề bánh răng

uh phiên bản màu sắc

uh màu hệ thống chấm màu xám hoặc đại loại như thế

và hãy tiếp tục và đợi

tải bản xem trước của chúng ta

tuyệt vời,

bây giờ chúng ta có thể tạo Chế độ xem hàng thực tế của mình

và sử dụng các tham số đầu vào này để điền vào

đó những gì chúng ta muốn,

vì vậy nó sẽ là khoảng cách ngăn xếp tuổi của

**Main subtitle**

image with a system name

of our image name so this is really just

made to work with the xcode system

images guys if you want to work with

custom images you'd have to implement

that functionality yourself but pretty

much for all these little like icons the

xcode system images that you get from

that SF symbols app are pretty

comprehensive and I haven't found the

need to download outside things for a

while so and they keep on adding to it

which is great so we're going to say dot

image scale is dot small.font is dot

title

dot foreground color is dot is uh the

tint color

okay cool and then next to that we're

going to say text

title

dot font Dot subheadline

dot foreground color of dot black

cool

um and that's it right we just didn't

want to have to copy and paste this a

bunch of times and you know obviously as

your settings page your profile page has

a bunch of actual like real options it's

going to be a lot more than this you

would get really tired of copy and

pasting this constantly

**Second subtitle**

hình ảnh 12 pixel với tên hệ thống

của tên hình ảnh của chúng ta, vì vậy điều này thực sự chỉ được

thực hiện để làm việc với

các hình ảnh hệ thống xcode nếu bạn muốn làm việc với

các hình ảnh tùy chỉnh, bạn phải tự triển khai

chức năng đó nhưng đối

với tất cả các biểu tượng nhỏ giống như các biểu tượng này,

hình ảnh hệ thống xcode mà bạn nhận được từ

ứng dụng biểu tượng SF đó khá

toàn diện và tôi đã lâu rồi không thấy

cần phải tải xuống những thứ bên ngoài

nên họ tiếp tục thêm vào đó,

điều này thật tuyệt vì vậy chúng tôi sẽ nói

tỷ lệ hình ảnh chấm là chấm nhỏ. Phong chữ là chấm tiêu đề

chấm màu nền trước là chấm uh

màu sắc

ổn và sau đó bên cạnh đó chúng ta

sẽ nói tiêu đề văn bản phong chữ chấm Dấu chấm

dòng tiêu đề phụ

màu nền trước của chấm đen

tuyệt vời

ừm và đúng vậy, chúng ta không

muốn phải sao chép và dán cái này

nhều lần và bạn biết rõ ràng vì

trang cài đặt của bạn, trang hồ sơ của bạn có

rất nhiều tùy chọn giống như thực, nó

sẽ còn nhiều hơn thế này, bạn

sẽ thực sự mệt mỏi với việc sao chép và

dán liên tục,

**Main subtitle**

um so let's go back to our profile view

and Implement that so here we're going to pass it or create a settings row View gear

version

and then it's going to be color dot system Gray

and let's clean that up a little bit

and then the other two

are going to be our account options so

oh and guys to get that uh 1.0 oh on the right here's what we can do for that

right this is pretty smooth we can actually just create an H stack paste that there

put a spacer here and say text 1.0.0

and we can make that dot font Dot

subheadline

dot foreground color of dot Gray

all right so that looks pretty good

and next up the account options so

we're going to want a button here

because this sign out guy is an actual button and clicking it would sign me out

we can see that in just a little bit

so we're going to want an action and a

label

you can just say print sign out

label is going to be a settings row View and

**Second subtitle**

ừm, vì vậy, hãy quay lại chế độ xem hồ sơ của chúng ta

và Thực hiện Vì vậy, ở đây chúng ta sẽ chuyển nó hoặc tạo một hàng cài đặt Xem

phiên bản bánh răng

và sau đó nó sẽ là

hệ thống chấm màu Xám

và hãy làm sạch nó một chút

và sau đó hai cái còn lại

sẽ là tùy chọn tài khoản của chúng ta.

và các bạn để có được điều đó uh 1.0 ở bên phải đây là những gì chúng ta có thể làm cho điều đó. Điều

này khá suôn sẻ, chúng ta

thực sự có thể tạo một ngăn xếp H dán ở đó

đặt một dấu cách ở đây và nói văn bản 1.0.0

và chúng ta có thể tạo dấu chấm đó phong chữ Dấu chấm

dòng tiêu đề phụ

dấu chấm màu nền trước của dấu chấm Xám

được rồi trông khá đẹp

và tiếp theo là các tùy chọn tài khoản, vì vậy

chúng tôi sẽ muốn có một nút ở đây

vì anh chàng đăng xuất này là một

nút thực và nhấp vào nó sẽ đăng xuất cho tôi,

chúng tôi có thể hãy xem điều đó chỉ trong chốc lát,

vì vậy chúng tôi sẽ muốn có một hành động và một

nhãn mà

bạn có thể chỉ cần nói nhãn đăng xuất in

sẽ là một hàng cài đặt Xem

và

**Main subtitle**

let me just uh copy this name really quickly

oops gotta put quotes around that

so it's `arrow.left.circle.fill`

title is sign out tint color is dot red

cool right and

let's get that there

and then one more button

for deleting our accounts

and I believe this is going to be `xmark.circle.fill`

and this is going to say delete account

so let's see let's let this load

and yeah that looks really good right guys so that's our profile page

um next up guys uh what I want us to do

is create like a user data model so that we can actually populate this with a

user and then we're going to get into the sign in and sign up stuff and we're going to need that user object to do

that stuff as well so that's what's coming next

so let's hop into our model folder guys

and once again this is going to be our data model for our user that we're going

to be signing in and out of our app and

creating and all of that good stuff so this is just going to be a regular Swift file and we're just going to call this

**Second subtitle**

để tôi sao chép tên này thật nhanh

Rất tiếc, phải đặt dấu ngoặc kép xung quanh `Vì`

vậy, đó là

tiêu đề `arrow.left.circle.fill` đang đăng xuất có màu sắc chấm đỏ

mát mẻ bên phải và

hãy đến đó

và sau đó thêm một nút nữa

để xóa tài khoản của chúng ta

và tôi tin rằng đây sẽ là `xmark.circle.fill`

và nút này sẽ nói

xóa tài khoản,

vì vậy hãy xem nào, hãy để cái này tải

và vâng, nó trông rất đẹp phải không các bạn, vậy đó là trang hồ sơ của chúng ta, tiếp theo các bạn uh, điều tôi muốn chúng ta làm

là tạo ra một mô hình dữ liệu người dùng để chúng ta có thể thực sự điền vào điều này với một

người dùng và sau đó chúng tôi sẽ

đăng nhập và đăng ký nội dung và chúng tôi sẽ cần đối tượng người dùng đó để thực hiện

công việc đó, vì vậy đó là những gì sắp tới,

vì vậy hãy nhảy vào thư mục mô hình của chúng tôi

và một lần nữa, đây sẽ là

mô hình dữ liệu của chúng tôi dành cho người dùng mà chúng tôi sẽ

đăng nhập và đăng xuất khỏi ứng dụng của mình, đồng thời

tạo và tất cả những nội dung hay đó, vì vậy đây sẽ chỉ là một tệp Swift thông thường và chúng tôi sẽ chỉ cần gọi

**Main subtitle**

user

and for any app that you guys create

that requires authentication you're going to need to create a user object like this it's how you represent your user programmatically in your code as a

data object right you're signing someone up that person is referred to as a user and you're going to need to refer to that user in probably almost every surface of your application right so in this example it's just going to be populating this profile with the user's

data so that when you log in as different users it displays the correct user data but yeah let's go ahead and

just create this struct so it's going to be a called user and it's going to conform to the identifiable protocol and the codable

protocol okay and we're going to say let ID and that's going to be a string let full name

that's going to be a string and let email which is also a string

and we're also going to have this initials property guys so we're going to

write or create a computed property here

to get the user's initials so that you see Michael Jordan is going to display MJ Steph and Dallas would display SD so on and so forth so we're going to create

**Second subtitle**

người dùng này

và đối với bất kỳ ứng dụng nào mà các bạn tạo yêu cầu

xác thực, bạn

sẽ cần tạo một đối tượng người dùng như thế này, đó là cách bạn đại diện cho người dùng của mình theo chương trình trong mã của bạn

dưới dạng đối tượng dữ liệu.

lên người đó được gọi là người dùng

và bạn sẽ cần đề cập đến

người dùng đó ở hầu hết mọi

bề mặt của ứng dụng của bạn, vì vậy trong

ví dụ này, nó sẽ

điền vào hồ sơ này với dữ liệu của người dùng

để khi bạn đăng nhập với tư cách

những người dùng khác nhau, nó sẽ hiển thị

dữ liệu người dùng chính xác nhưng vâng, hãy tiếp tục và

chỉ cần tạo cấu trúc này

để nó trở thành người dùng được gọi và

nó sẽ tuân theo

giao thức có thể nhận dạng và giao thức có thể mã hóa

được và chúng tôi sẽ nói let

ID và đó sẽ là một chuỗi

let full name

sẽ là một string và let

email cũng là một string

và chúng ta cũng sẽ có

thuộc tính chữ cái đầu này nhé các bạn, vì vậy chúng ta sẽ

viết hoặc tạo một thuộc tính được tính toán ở đây

để lấy tên viết tắt của người dùng để bạn

thấy Michael Jordan sẽ hiển thị

MJ Steph và Dallas sẽ hiển thị SD,

v.v. vì vậy chúng tôi sẽ tạo

**Main subtitle**

an initials guy here

it's going to be a string apple or Swift

actually has a component to help us with

this we're going to say let formatter

equals person name components  
components

formatter

we're going to say if let components

equals formatter dot person name

components from full name okay so it's

going to look at the full name and this

object here can divide it up into

components and give us back what we need

from the user's full name which is

really really nice right saves us a lot

of trouble from having to do it we're

going to say formatter.style equals dot

abbreviated

and we're going to say return formatter

dot string from components

and if it's nothing we're just going to

return a blank string

in a real app you guys might want to

return like an image or something like

that but we're not going to be doing

that here where it would be like a blank

profile image view but you know

everything goes according to plan the

user has to submit a full name when they

sign up even if it's just like one

character

**Second subtitle**

một tên viết tắt ở đây,

nó sẽ là một chuỗi apple hoặc Swift

thực sự có một thành phần để giúp chúng  
tôi với

điều này, chúng tôi sẽ nói hãy để bộ định  
dạng

bằng tên người, bộ định dạng thành phần  
tên người, bộ định dạng thành phần,

chúng tôi sẽ nói nếu để thành phần bằng bộ  
định dạng,

dấu chấm thành phần tên người

từ tên đầy đủ, vì vậy nó

sẽ xem xét tên đầy đủ và

đối tượng này ở đây có thể chia nó thành

các thành phần và trả lại cho chúng tôi  
những gì chúng tôi cần

từ tên đầy đủ của người dùng, điều này

thực sự rất hay giúp chúng tôi tiết kiệm rất

nhều rắc rối khi phải làm điều đó, chúng tôi

sẽ nói formatter.style bằng dấu

chấm viết tắt

và chúng tôi sẽ nói rằng trình định dạng trả  
về

chuỗi dấu chấm từ các thành phần

và nếu không có gì, chúng tôi sẽ

trả về một chuỗi trống

trong một ứng dụng thực, các bạn có thể  
muốn

trả về dưới dạng hình ảnh hoặc thứ gì đó  
tương

tự nhưng chúng tôi sẽ không để làm

điều đó ở đây, nơi nó sẽ giống như một chế  
độ

xem ảnh hồ sơ trống nhưng bạn biết đấy

mọi thứ diễn ra theo đúng kế hoạch,

người dùng phải gửi tên đầy đủ khi họ

đăng ký ngay cả khi nó chỉ giống như một

ký tự,



**Main subtitle**

um you know we're going to do some form validation as well then this won't be a

problem so now that we have this user

object let's go ahead and just create like a mock user so I can say extension

user

static VAR mock

underscore user

equals user

and the ID is just going to be an NS

uuid dot uuid string full name

let's make it Kobe Bryant

test at

gmail.com okay so this is our mock user

right guys

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

just populate this profile page with like actual user data so let's go back

to our profile view

and here instead of MJ we can say like

mock user

or sorry we would say user dot mock user

Dot

initials just like that

so you see now it's KB for Kobe Bryant

which is really cool

now we're going to go here and say user

dot mock user dot full name

and we can copy and paste this

**Second subtitle**

bạn biết đấy, chúng tôi sẽ thực hiện một số xác thực biểu mẫu thì điều này sẽ không thành

vấn đề vì vậy bây giờ chúng ta có đối tượng người dùng này,

hãy tiếp tục và chỉ cần tạo

giống như một người dùng giả để tôi có thể nói người dùng tiện ích mở rộng

người dùng dấu gạch dưới VAR tĩnh bằng người dùng

và ID chỉ là sẽ là một

chuỗi NS uuid dot uuid tên đầy đủ,

hãy biến nó thành

bài kiểm tra Kobe Bryant tại

gmail.com nhé, vì vậy đây là người dùng giả của chúng ta

phải không các bạn,

vì vậy bây giờ hãy tiếp tục và xem liệu chúng ta có thể

điền trang hồ sơ này với

dữ liệu người dùng thực tế không vì vậy, hãy quay lại

chế độ xem hồ sơ của chúng ta

và ở đây thay vì MJ, chúng ta có thể nói thích

người dùng giả

hoặc xin lỗi, chúng tôi sẽ nói người dùng chấm người dùng giả lập

Dấu chấm

viết tắt giống như vậy

để bạn thấy bây giờ đó là KB cho Kobe Bryant,

điều này thực sự tuyệt vời

bây giờ chúng tôi sẽ đi đến đây và nói người dùng

dấu chấm mô phỏng tên đầy đủ của người dùng

và chúng tôi có thể sao chép và dán

**Main subtitle**

make this email

so the test guy is going to just be  
right you know it's uh

it's the same between the two things you  
can change it back in the mock user

guide doesn't really matter

um you guys will notice now that because

this is weird because it's this is a  
property that's coming from somewhere  
else now it's just not a raw string

right here the accent color guide  
doesn't work so here is where you could

now change the foreground color to dot  
gray I just wanted to go over that  
technicality it is pretty confusing and  
I think it's a bug with uh in Swift that  
Apple needs to fix I'm not sure if they  
ever will it's pretty small and it is  
kind of annoying so hopefully they do  
fix that

uh somebody submitted a complaint to  
them for me please because I don't feel  
like dealing with them anyway

um so that looks really good so now we  
have a user object we are populating our  
profile page with actual user data so

now it's time for the big boy stuff

right we need to set up Firebase so that  
we can have a service that authenticates  
users for us creates them we can log in

with that if you guys get the completed

**Second subtitle**

email này để tạo email này

để người kiểm tra sẽ

chính xác, bạn biết đấy, uh,

hai thứ giống nhau mà bạn

có thể thay đổi lại trong hướng dẫn sử dụng  
mô phỏng

không thực sự quan trọng

ừm, bây giờ các bạn sẽ nhận thấy rằng bởi  
vì

điều này thật kỳ lạ bởi vì đây là một

thuộc tính đến từ một nơi

khác, bây giờ nó không phải là một chuỗi  
thô

ngay tại đây, hướng dẫn màu nhấn

không hoạt động nên ở đây là nơi bây giờ  
bạn có thể

thay đổi màu nền trước thành chấm

xám. Tôi chỉ muốn xem qua

tính kỹ thuật đó, nó khá khó hiểu và

tôi nghĩ đó là một lỗi với uh trong Swift mà

Apple cần sửa. Tôi không chắc liệu họ

có sửa được không. nhỏ và

khá phiền phức, vì vậy hy vọng họ sẽ

khắc phục điều đó

uh ai đó đã gửi khiếu nại cho

họ giúp tôi vì

dù sao thì

tôi cũng không muốn

giao dịch với họ.

trang hồ sơ của chúng tôi với dữ liệu người  
dùng thực tế, vì vậy

bây giờ đã đến lúc dành cho công cụ quan  
trọng,

chúng tôi cần thiết lập Firebase để

chúng tôi có thể có dịch vụ xác thực

người dùng để chúng tôi tạo ra họ, chúng tôi  
có thể đăng nhập

bằng dịch vụ đó nếu các bạn nhận được  
nguồn hoàn chỉnh

**Main subtitle**

source code you have options like for getting a password and all that stuff so

that's what we're going to be doing next we're going to be setting up Firebase and then we're going to create an authentication view model to house all of the functionality for logging a user in signing them up signing them out deleting their account and all of that good stuff and then we also need to do

some presentation logic in the app so that you know if I hit sign out it goes back to my login page and if I

were to close the app and relaunch it you know we wanted to open the Curt to

the correct page as well right I'm going to show you guys how to make it so that if the user signs into the app one time

unless they sign out it will always direct them to the main interface of the app or that profile view which is really

cool unless you're building an app like

a banking app or something where you want the user to have to log in every time typically most apps don't do that so we're going to be showing you how to

do that here as well so next up we're

going to set up Firebase guys let's do it so really quickly before we set up

our own Firebase project I just want to

**Second subtitle**

mã, bạn có các tùy chọn như

lấy mật khẩu và tất cả những thứ đó, vì vậy đó là

những gì chúng tôi sẽ làm tiếp theo,

chúng tôi sẽ thiết lập Firebase

và sau đó chúng tôi sẽ tạo một

mô hình chế độ xem xác thực để chứa tất

cả chức năng ghi nhật ký người dùng

đăng ký họ đăng xuất

xóa tài khoản của họ và tất cả

những nội dung hay đó, sau đó chúng tôi cũng cần thực hiện

một số logic trình bày trong ứng dụng

để bạn biết nếu tôi nhấn đăng xuất thì

nó sẽ quay lại trang đăng nhập của tôi và nếu tôi

đóng ứng dụng và khởi chạy lại,

bạn biết đấy, chúng tôi cũng muốn mở Curt đến

đúng trang. Tôi sẽ chỉ

cho các bạn cách tạo ứng dụng đó để

nếu người dùng đăng nhập vào ứng dụng một lần

trừ khi họ đăng xuất, nó sẽ luôn

hướng họ đến giao diện chính của

ứng dụng hoặc chế độ xem hồ sơ đó, điều này thực sự

thú vị trừ khi bạn đang xây dựng một ứng dụng như

ứng dụng ngân hàng hoặc thứ gì đó mà bạn

muốn người dùng phải đăng nhập mọi

lúc. các ứng dụng không làm điều đó,

vì vậy chúng tôi cũng sẽ chỉ cho bạn cách thực

hiện điều đó ở đây, vì vậy, tiếp theo chúng ta

sẽ thiết lập Firebase. Hãy

thực hiện thật nhanh trước khi chúng ta thiết lập

dự án Firebase của riêng mình. muốn

**Main subtitle**

show you guys what the completed one looks like so you have a better idea of

what we're going to be using this for if you're not familiar with Firebase so we

have this authentication section and this is where all of our user

authentication is going to exist we can see that we signed up with email and

this is our user ID when they signed in

and when they created their account and then our database is where we're going

to actually store all of the users information so you can see here that

this

Kobe Bryant's information and then this is Michael Jordan's information so that we are actually going to be able to store this information in Firebase and retrieve it when a user logs in and

display it in the application this is the foundation of any application that stores uh or creates user profiles and authenticates users so once again you

guys are going to just be able to drag and drop this whole feature into any app

that you create that requires user profiles accounts or user authentication

whatever it may be so let's go over to Firebase if you guys have never done

**Second subtitle**

cho các bạn thấy cái đã hoàn thành trông như thế nào để bạn có ý tưởng tốt hơn về việc

chúng tôi sẽ sử dụng cái này để làm gì nếu bạn không quen thuộc với Firebase nên chúng tôi

có phần xác thực này và

đây là nơi mà tất cả người dùng của chúng tôi

xác thực sẽ tồn tại, chúng tôi có thể thấy rằng chúng tôi đã đăng ký bằng email và

đây là ID người dùng của chúng tôi khi họ đăng nhập

và khi họ tạo tài khoản và

sau đó cơ sở dữ liệu của chúng tôi là nơi chúng tôi sẽ

thực sự lưu trữ tất cả

thông tin người dùng, vì vậy bạn có thể thấy ở đây

thông tin này của Kobe Bryant và sau đó

là thông tin của Michael Jordan để

chúng tôi thực sự có thể

lưu trữ thông tin này trong Firebase và

truy xuất thông tin đó khi người dùng đăng nhập và

hiển thị thông tin đó trong ứng dụng, đây là nền tảng của bất kỳ ứng dụng

lưu trữ uh hoặc tạo hồ sơ người dùng và xác thực người dùng, vì vậy, một lần nữa, các

bạn sẽ có thể kéo

và thả toàn bộ tính năng này vào bất kỳ ứng dụng

nào mà bạn tạo yêu cầu

tài khoản hồ sơ người dùng hoặc xác thực người dùng,

bất kể đó là gì. chuyển sang

Firebase nếu các bạn chưa từng làm

**Main subtitle**

this before go to

console.firebase.google.com you do need to be signed into a Google account to create or use Firebase and create a project so here I'm going to call this Swift UI auth tutorial

let's continue we do not need analytics let's go ahead and create our project and let's just go ahead and let it do its thing here it um

moves pretty quickly and what it's doing right now is literally just creating our Firebase

project and then we're going to set up authentication and our database awesome we're ready to go hit continue and now guys what we're going to do is

go to build and go to Authentication so on this left menu here go to build

and then go to authentication and hit get started

and this is where we're going to sign up or deal with handling our user

authentication so you can see we have a bunch of different providers

I'm going to be adding functionality to this uh so if you guys you know get the source code you will get access to updates

um if you are a gold member and above I believe so you get added to the GitHub

**Second subtitle**

điều này trước đây, hãy truy cập

console.firebase.google.com, bạn cần phải đăng nhập vào tài khoản Google để tạo hoặc sử dụng Firebase và tạo một dự án, vì vậy ở đây tôi sẽ gọi đây là Giao diện người dùng Swift hướng dẫn xác thực,

hãy tiếp tục, chúng tôi không cần phân tích, hãy tiếp tục và tạo dự án của chúng tôi và hãy tiếp tục và để nó thực hiện công việc của mình ở đây, nó

di chuyển khá nhanh

và những gì nó đang làm hiện tại thực sự chỉ là tạo dự án Firebase của chúng tôi

và sau đó chúng tôi' chúng ta sẽ thiết lập xác thực và cơ sở dữ liệu của chúng ta thật tuyệt vời, chúng ta đã sẵn sàng nhấn tiếp tục và bây giờ, những gì chúng ta sẽ làm là

xây dựng và đi đến Xác thực, vì vậy, trên menu bên trái này, hãy đi đến xây dựng

và sau đó đến xác thực và nhấn bắt đầu

và đây là nơi chúng tôi sẽ đăng ký hoặc xử lý việc xử lý xác thực người dùng của chúng tôi

để bạn có thể thấy chúng tôi có một loạt các nhà cung cấp khác nhau.

Tôi sẽ thêm chức năng này uh vì vậy nếu các bạn biết lấy mã nguồn, bạn sẽ có quyền truy cập vào các bản cập nhật

ừm nếu bạn là thành viên vàng trở lên, tôi tin rằng vì vậy bạn được thêm vào

**Main subtitle**

repository of this so you get access to updates as they come where I'm going to be adding stuff like Google sign in Facebook sign in and all that but we are for now just going to be dealing with email and password sign in so go to enable and hit save and let's see we're good to go there and then we're going to go back to build and you're going to go to firestore database and we're going to set this up as well so hit create database and we're going to start in test mode hit next and hit enable the location that it defaults to should be fine and this is creating our databases so the difference between the two things is one of these components the authentication guy handles user authentication it creates a user account the database actually stores the user's information like their the other metadata corresponding to a user like their full name their user identifier and whatever else a user account is going to have in your application this is obviously very basic some users

**Second subtitle**

kho GitHub của cái này để bạn có quyền truy cập vào các bản cập nhật khi chúng đến nơi tôi sẽ thêm những thứ như Đăng nhập Google Đăng nhập Facebook và tất cả những thứ đó nhưng hiện tại chúng tôi sẽ xử lý đăng nhập email và mật khẩu, vì vậy hãy bật và nhấn lưu và hãy xem chúng tôi ổn để đến đó và sau đó chúng tôi sẽ bắt đầu quay lại xây dựng và bạn sẽ truy cập cơ sở dữ liệu firestore và chúng tôi cũng sẽ thiết lập cơ sở dữ liệu này, vì vậy hãy nhấn tạo cơ sở dữ liệu và chúng tôi sẽ bắt đầu ở chế độ thử nghiệm, nhấn tiếp theo và nhấn bật vị trí mà nó mặc định. được rồi và điều này đang tạo cơ sở dữ liệu của chúng tôi, vì vậy sự khác biệt giữa hai điều là một trong những thành phần này, anh chàng xác thực xử lý xác thực người dùng, nó tạo tài khoản người dùng, cơ sở dữ liệu thực sự lưu trữ thông tin của người dùng giống như siêu dữ liệu khác của họ tương ứng với người dùng như thông tin đầy đủ của họ đặt tên cho số nhận dạng người dùng của họ và bất kỳ thứ gì khác mà tài khoản người dùng sẽ có trong ứng dụng của bạn, điều này rõ ràng là rất cơ bản, một số

**Main subtitle**

objects can get very complex like an

Instagram user for example has like hundreds and hundreds of properties associated with it

um you know so this is just a very basic

sort of shell for what a user is going to look like and now our database is set

up as well so now guys

um let's go back to our Swift UI auth

tutorial here and we are going to add Firebase to our app now so let's go and

hit iOS there and we need to get the bundle identifier from our application so let's go back here select your project name all the way up at the top in the directory and grab this bundle identifier you're going to hit the arrow

and copy and paste this

into here and then hit Register App and now we need to download this Google service info Dash plist file

I'm going to go to my downloads and as you guys can see I have it right here

um what I like to do guys is if you see

a dash in a number like this you can't

use this info plist file you have to delete that but if you TR if you have like a Google info p list somewhere else

in here it's not going to allow you to

**Second subtitle**

đối tượng người dùng có thể trở nên rất phức tạp, chẳng hạn như

người dùng Instagram có hàng trăm và hàng trăm thuộc tính được liên kết với nó,

bạn biết điều này chỉ là một loại vỏ rất cơ bản

cho

giao diện của người dùng và bây giờ cơ sở dữ liệu của chúng tôi cũng đã được thiết lập, vì vậy bây giờ các bạn,

hãy quay lại hướng dẫn xác thực giao diện người dùng Swift của chúng tôi

tại đây và chúng tôi sẽ thêm

Firebase vào ứng dụng của mình bây giờ, hãy bắt đầu và

nhấn iOS ở đó và chúng tôi cần lấy số nhận dạng gói từ ứng dụng của mình, vì vậy hãy quay lại đây, chọn tên dự án của bạn ở trên cùng trong thư mục và lấy số nhận dạng gói này mà bạn sẽ nhấn mũi tên, sao

chép và dán cái này

vào đây, sau đó nhấn Đăng ký ứng dụng và bây giờ chúng tôi cần tải xuống tệp plist Dash thông tin dịch vụ Google này. Tôi sẽ chuyển đến phần tải xuống của mình và như các bạn có thể thấy, tôi có nó ngay tại đây,

um những gì tôi thích điều cần làm là nếu bạn thấy

một dấu gạch ngang trong một số như thế này, bạn không thể

sử dụng tệp plist thông tin này, bạn phải xóa tệp đó nhưng nếu bạn TR nếu bạn thích danh sách thông tin p của Google ở một nơi khác

ở đây thì điều đó sẽ không được phép bạn

**Main subtitle**

do that so what I like to do is create

like a new

instance of my finder by hitting command

option n

go to the location of your project so

let's see where is my desktop

programming

Swift UI auth tutorial and just drag

this guy into there

and now we can change the name of it

this is thrown a lot of people off this

file has to say Google service Dash info

plist it can't say anything else so now

that we've done that let's go ahead and

hit next and we need to sorry actually

drag and drop this into our Firebase

project

excuse me for bouncing around a little

bit here guys there we go

so now

just drag and drop it there

make sure you have copy items if needed

checked

and now we have our Google service info

p list so for those of you wondering

what this is for this is just how

our app is going to communicate with the

Firebase server that we just set up

right so it has things like our client

ID or reverse client ID or API key all

**Second subtitle**

làm điều đó vì vậy những gì tôi muốn làm là tạo

giống như một

phiên bản mới của công cụ tìm kiếm của tôi bằng cách nhấn

tùy chọn lệnh n

đi đến vị trí dự án của bạn, vì vậy

hãy xem hướng dẫn xác thực Swift UI lập trình máy tính để bàn của tôi ở đâu

và chỉ cần kéo

anh chàng này vào đó

và bây giờ chúng tôi có thể thay đổi tên của nó, điều

này đã khiến rất nhiều người loại bỏ tệp này.

thực sự xin lỗi, hãy

kéo và thả cái này vào dự án Firebase của chúng tôi,

xin lỗi vì đã nhảy lung tung một

chút ở đây, chúng ta đi đây,

vì vậy bây giờ

chỉ cần kéo và thả nó ở đó,

đảm bảo rằng bạn đã kiểm tra các mục sao chép nếu cần

và bây giờ chúng tôi có thông tin dịch vụ Google của chúng tôi

p vì vậy đối với những bạn đang thắc mắc

đây là gì thì đây chỉ là cách

ứng dụng của chúng tôi sẽ giao tiếp với

máy chủ Firebase mà chúng tôi vừa thiết lập

để nó có những thứ như ID ứng dụng khách của chúng tôi

hoặc ID ứng dụng ngược hoặc khóa API tắt



**Main subtitle**

of the things we need to securely communicate with our Firebase server so you know not just anybody can access that information so guys make sure you keep this information secret obviously I don't care if you see mine because this is just a test project now what we need to do is install the Swift package into our xcode project so that we can actually utilize the Firebase code so basically Firebase has written a bunch of code that we can use we just have to integrate or download their code into our code base and we're going to do that using the Swift package manager so guys go ahead and copy this and let's go back to xcode file and we are going to say add packages go ahead and paste that in there Firebase iOS SDK and select add package and this might take a second to do its thing so we're going to get presented with a checklist here in just a second so let's see um and I'll tell you guys which modules that we want to select for our project here

**Second subtitle**

cả những thứ chúng tôi cần để liên lạc an toàn với máy chủ Firebase của mình để bạn biết rằng không phải ai cũng có thể truy cập thông tin đó, vì vậy mọi người hãy đảm bảo rằng bạn giữ bí mật thông tin này rõ ràng là tôi không quan tâm nếu bạn thấy thông tin của tôi vì đây chỉ là một dự án thử nghiệm. việc cần làm là cài đặt gói Swift vào dự án xcode của chúng tôi để chúng tôi thực sự có thể sử dụng mã Firebase, vì vậy về cơ bản Firebase đã viết một loạt mã mà chúng tôi có thể sử dụng, chúng tôi chỉ cần tích hợp hoặc tải mã của họ vào cơ sở mã của chúng tôi và chúng tôi sẽ làm điều đó bằng cách sử dụng trình quản lý gói Swift, vì vậy mọi người hãy tiếp tục và sao chép tệp này và hãy quay lại tệp xcode và chúng tôi sẽ nói thêm các gói, hãy tiếp tục và dán gói đó vào SDK iOS Firebase và chọn thêm gói và việc này có thể mất một khoảng thời gian thứ hai để thực hiện công việc của nó, vì vậy chúng tôi sẽ trình bày một danh sách kiểm tra ở đây chỉ sau một giây, vì vậy hãy xem ừm và tôi sẽ cho các bạn biết chúng tôi muốn chọn mô-đun nào cho dự án của mình ở đây

**Main subtitle**

so we see this checklist here guys I'm

going to go ahead and expand it just a little bit let's see which ones we need so we don't need analytics or any of

that we do need Firebase auth for sure um we are going to also use Firebase firestore that's how we communicate with our database and Firebase firestore

Swift as well this enables us to actually encode and decode Firebase data that comes back

from uh the Firebase servers so I think that's good if you guys want to work

with images do Firebase storage as well we're not doing anything with profile images here but I'll check it anyway

um and then we just need to let all of these package dependencies install so for those of you who don't know Swift package manager is a much better better alternative to cocoapods because

it enables us to not have to use an XC workspace or download cocoapods from the

terminal and all that this is a much better alternative as I'm sure you guys

already know but there are some people who are new to this and new to the channel

um so you know I got to explain all

those little things like that

**Second subtitle**

để chúng tôi xem danh sách kiểm tra này ở đây các bạn Tôi

sẽ tiếp tục và mở rộng nó một

chút, hãy xem cái nào chúng tôi cần

vì vậy chúng tôi không cần phân tích hoặc bất kỳ cái nào trong số đó

chúng tôi cần xác thực Firebase chắc chắn

chúng tôi cũng sẽ sử dụng Firebase

firestore đó là cách chúng tôi giao tiếp với

cơ sở dữ liệu của chúng tôi và Firebase Firestore

Swift cũng như điều này cho phép chúng tôi thực sự mã hóa và giải mã dữ liệu Firebase quay trở lại

từ các máy chủ Firebase, vì vậy tôi nghĩ điều đó thật tốt nếu các bạn cũng muốn làm việc

với hình ảnh để lưu trữ Firebase.

làm bất cứ điều gì với

hình ảnh hồ sơ ở đây nhưng dù sao thì tôi cũng sẽ kiểm tra nó

và sau đó chúng ta chỉ cần

cài đặt tất cả các phụ thuộc gói này

để những người chưa biết

Swift package manager là một

giải pháp thay thế tốt hơn nhiều cho cocoaPods vì

nó cho phép chúng tôi không phải sử dụng

không gian làm việc XC hoặc tải xuống cocoaPods từ

thiết bị đầu cuối và tất cả điều này là một

giải pháp thay thế tốt hơn nhiều vì tôi chắc rằng các bạn

đã biết nhưng có một số người

mới biết điều này và mới biết

kênh

ừm vì vậy bạn biết đấy, tôi phải giải thích tất cả

những điều nhỏ nhặt như vậy

**Main subtitle**

but anyway

um now that we have Firebase set up guys

we're actually going to be able to now  
start communicating with our Firebase  
servers to register users and log them

in and sign them out and all that stuff

so that's what's coming next we're going  
to be creating and setting up our  
authentication view model so get excited

for that guys

so let's hop up into our authentication

folder and we're going to create two  
groups here one for views  
and one for view models guys just to  
keep this organized

so obviously these two guys  
will go into our view folder

and let's put that there and now we're  
going to go here into our viewmodel

folder and create this auth view model  
so really quickly let's talk about what  
the auth view model is going to be  
responsible for I believe there was an  
error back there

okay yeah that's good to go

um so our authentication view model is

going to be responsible for a couple  
things it's going to be responsible for  
having all the functionality associated  
with authenticating our user right

**Second subtitle**

nhưng dù

sao thì bây giờ chúng tôi đã thiết lập  
Firebase,

chúng tôi thực sự sẽ có thể bắt

đầu giao tiếp với

các máy chủ Firebase của mình để đăng ký  
người dùng và đăng

nhập và đăng nhập cho họ out và tất cả  
những thứ đó,

vì vậy đó là những gì sắp tới, chúng tôi sẽ  
tạo và thiết lập

mô hình chế độ xem xác thực của chúng tôi,  
vì vậy hãy hào hứng

với những người đó, vì vậy

hãy truy cập vào thư mục xác thực của  
chúng tôi

và chúng tôi sẽ tạo hai

nhóm ở đây, một cho các khung nhìn

và một cho các mô hình xem chỉ để

giữ cho nó có tổ chức,

vì vậy rõ ràng hai người này

sẽ vào thư mục xem của chúng ta

và hãy đặt nó ở đó và bây giờ chúng ta

sẽ vào đây trong thư mục viewmodel của  
chúng ta

và tạo mô hình xem auth này

thật nhanh hãy nói về những gì

mô hình chế độ xem xác thực sẽ

chịu trách nhiệm. Tôi tin rằng đã xảy ra

lỗi ở đó,

vâng, tốt thôi,

ừm, vì vậy mô hình chế độ xem xác thực  
của chúng tôi

sẽ chịu trách nhiệm về một số

điều mà nó sẽ chịu trách nhiệm về

việc có tất cả các chức năng liên quan

đến việc xác thực người dùng của chúng tôi  
có quyền

**Main subtitle**

making all those Network calls and stuff

and ultimately being in charge of  
updating our UI or sending notifications

to our UI when we need to update it so

basically what that's going to look like  
is like when I type in test gmail.com

and one in my password  
um my off view model is going to handle  
the form validation but it's also going  
to handle the networking for actually  
signing my user in right and then it's

also going to send updates to my view

so that it can present me with the Right  
View when you know things successfully  
go through so another thing it's going  
to do is for example if I said test like  
one two three four gmail.com  
and try to sign in I get an error  
message it looks like there's no account

associated with this email create an  
account to continue that's because  
there's no account registered with this  
our authentication view model is going

to handle all of the errors as well so  
it's got some pretty like pretty big  
responsibilities behind it so let's go  
ahead and get started with it guys so  
I'm going to create a class here called  
auth viewmodel

and it's going to confirm to conform to  
the observable object protocol

**Second subtitle**

thực hiện tất cả các cuộc gọi và nội dung  
trên Mạng đó

và cuối cùng chịu trách nhiệm

cập nhật giao diện người dùng của chúng  
tôi hoặc gửi thông báo

tới giao diện người dùng của chúng tôi khi  
chúng tôi cần cập nhật

giao diện người dùng đó. trong thử nghiệm  
gmail.com

và một trong mật khẩu của tôi,

ừm, mô hình tắt chế độ xem của tôi sẽ xử lý  
xác thực biểu mẫu nhưng nó cũng sẽ  
xử lý mạng để thực sự

đăng nhập đúng người dùng của tôi và sau  
đó nó

cũng sẽ gửi các bản cập nhật cho chế độ  
xem của tôi

để nó có thể hiển thị cho tôi Chánh

kiến khi bạn biết mọi thứ đã thành công, vì  
vậy, một điều khác mà nó sẽ

làm là ví dụ nếu tôi nói kiểm tra như

một hai ba bốn gmail.com

và thử đăng nhập, tôi nhận được

thông báo lỗi, có vẻ như vậy không có tài  
khoản nào

được liên kết với email này, hãy tạo một  
tài khoản để tiếp tục bởi vì

không có tài khoản nào được đăng ký với  
mô hình chế độ xem xác thực của chúng tôi  
cũng sẽ

xử lý tất cả các lỗi nên

nó có một số

trách nhiệm khá lớn đằng sau nó, vì vậy hãy

tiếp tục và bắt đầu nào các bạn, vì vậy

tôi sẽ tạo một lớp ở đây có tên là

auth viewmodel

và nó sẽ xác nhận việc tuân thủ

giao thức đối tượng quan sát được,

**Main subtitle**

so basically our view is going to be

able to observe changes on our auth view

model

I'm going to create a couple published

properties here as well one is going to be called user session

and let's actually go up to the top here

and import

Firebase

and this is going to be Firebase auth dot user

and it's going to be optional so we're going to go over what this is going to be used or how it's going to be used in more detail in a second but it's

basically going to tell us whether or not we have a user logged in so then

when we open our app it knows whether or

not to Route us to the login flow or the

profile view

um

then we're also going to have a published variable for our current user

and this is going to be our user object

so these two things are different this is a Firebase auth user and this is our user that we created here so we're going

to need both of these things and we'll go over that in more detail in a little bit

**Second subtitle**

vì vậy về cơ bản, chế độ xem của chúng tôi sẽ có

thể quan sát các thay đổi trên mô hình chế độ xem xác thực

của chúng tôi. sẽ tạo một vài thuộc tính đã xuất bản

ở đây cũng như một thuộc tính sẽ được gọi là phiên người dùng

và chúng ta hãy thực sự đi lên trên cùng ở đây

và nhập

Firebase

và đây sẽ là

người dùng dấu chấm xác thực Firebase

và nó sẽ là tùy chọn vì vậy chúng ta

sẽ xem xét những gì nó sẽ được

sử dụng hoặc cách nó sẽ được sử dụng

chi tiết hơn trong một giây nhưng

về cơ bản, nó sẽ cho chúng tôi biết liệu

chúng tôi có người dùng đăng nhập hay không để

khi chúng tôi mở ứng dụng của mình, nó sẽ biết liệu hoặc

không Định tuyến chúng tôi đến luồng đăng nhập hoặc chế

độ xem hồ sơ

,

thì chúng tôi cũng sẽ có một

biến được xuất bản cho người dùng hiện tại của mình

và đây sẽ là đối tượng người dùng của chúng tôi

nên hai điều này khác nhau, đây

là người dùng xác thực Firebase và đây là người dùng của chúng tôi mà chúng tôi đã tạo ở đây vì vậy chúng tôi sẽ

cần cả hai thứ này và chúng tôi sẽ xem xét chi tiết hơn một chút ừm,

**Main subtitle**

um now we're going to have an initializer and we'll go over what this is going to be filled with I just want to fill out all of the functions that we're going to have in here guys so first off we're going to have a function to sign in with email email which is a string password which is a string and it's going to say async throws and if you guys have never seen this async throws that's because Swift you uh Swift released this new async await functionality that allows us to network in a much much better way than using URL sessions and a bunch of nasty completion blocks and you guys might have guessed it I got a tutorial for it the link in the description or the link is in the description to this video if you want to learn more about how async throws work or async await works I'm really excited about it I love using it I've been refactoring all my old projects to use it so this is sort of the future guys um so let's go ahead and continue on with what we need from this auth view model we're going to have another function to create a user with email email string password string full name String and once again we're going to say async

**Second subtitle**

bây giờ chúng tôi sẽ có một trình khởi tạo và chúng tôi sẽ xem qua cái này sẽ được lấp đầy bằng gì Tôi chỉ muốn điền vào tất cả các chức năng mà chúng ta sẽ có ở đây, vì vậy trước tiên chúng ta sẽ có một chức năng để đăng nhập bằng email email là một chuỗi mật khẩu đó là một chuỗi và nó sẽ nói lệnh ném async và nếu các bạn chưa bao giờ thấy lệnh ném async này thì đó là bởi vì Swift, bạn uh Swift đã phát hành chức năng chờ async mới này cho phép chúng tôi kết nối mạng theo cách tốt hơn nhiều so với sử dụng các phiên URL và một loạt các khối hoàn thành khó chịu và các bạn có thể đoán ra. Tôi có một hướng dẫn cho nó, liên kết trong phần mô tả hoặc liên kết có trong phần mô tả của video này nếu bạn muốn tìm hiểu thêm về cách thức hoạt động của async throws hoặc async await hoạt động. Tôi' Tôi thực sự hào hứng với nó. Tôi thích sử dụng nó. Tôi đã tái cấu trúc tất cả các dự án cũ của mình để sử dụng nó, vì vậy đây là một loại tương lai, ừm, vì vậy hãy tiếp tục và tiếp tục với những gì chúng ta cần từ mô hình chế độ xem xác thực này. để có một chức năng khác để tạo người dùng với email chuỗi email chuỗi mật khẩu tên đầy đủ Chuỗi và một lần nữa chúng ta sẽ nói async

**Main subtitle**

throws

just like that

and then we're going to have a function to sign out

and we're going to have a function to delete our account

and then we're also going to have a function to fetch our user data

okay and this is just going to be async we don't need to have a throws here and we'll talk more about that later what I

want us to do now is go ahead and see

how we're going to implement this Osby model and the way we need to implement

it and all of the different places that

we need to implement it as well so you know obviously we're going to need it why is this yelling at me I don't get

it

yeah there we go okay so we're going to

need it inside of our login view right

because when we hit this button we want

it to log our user in so what we would

do is create an instance of our view

model up at the top so we could say like at State object VAR view model

equals auth viewmodel right

and then we could go down here

**Second subtitle**

ném

giống như vậy

và sau đó chúng ta sẽ có một chức năng để đăng xuất

và chúng ta sẽ có chức năng

xóa tài khoản của chúng tôi

và sau đó chúng tôi cũng sẽ có

chức năng tìm nạp dữ liệu người dùng của chúng tôi

và điều này sẽ không đồng bộ,

chúng tôi không cần phải cố gắng ở đây và chúng tôi sẽ nói thêm về điều đó sau đó, những gì tôi

muốn chúng ta làm bây giờ là tiếp tục và xem

cách chúng ta sẽ triển khai

mô hình Osby này và cách chúng ta cần triển khai

nó cũng như tất cả những nơi khác nhau mà

chúng ta cũng cần triển khai để

bạn biết rõ ràng là chúng tôi 'Tôi sẽ cần

nó tại sao điều này lại hét vào mặt tôi Tôi không hiểu,

vâng, chúng tôi sẽ ổn thôi, vì vậy chúng tôi sẽ

cần nó bên trong chế độ xem đăng nhập của mình

vì khi chúng tôi nhấn nút này, chúng tôi muốn

nó ghi nhật ký người dùng của mình Vì vậy, những gì chúng tôi sẽ

làm là tạo một phiên bản của mô hình chế độ xem của chúng tôi

ở trên cùng để chúng tôi có thể nói như

ở Đối tượng trạng thái VAR, mô hình chế độ xem

bằng auth chế độ xem đúng

và sau đó chúng tôi có thể đi xuống đây

**Main subtitle**

and where did that go right here  
 and say  
 viewmodel DOT sign in with email and  
 password  
 and because this is using async await

guys we actually need to wrap this in a

task and say try a weight  
 viewmodel.signin with email and password  
 so that's one way of doing it  
 uh in terms of how we implement the view  
 model like this functionality is is good

here when we click the button we want

our we want to you know call this sign  
 in function that's on our viewmodel but

this is going to get annoying because  
 the if you initialize it as a state

object

because we also need this in our  
 registration view right and what you

don't want to do is create another  
 instance of the view model here and just

have a bunch of different instances of  
 your view model all across the app so

you say okay maybe you could create an  
 observed object

auth view model but then when we create

the registration view we need to  
 initialize it with the view model  
 and that gets annoying as well so the  
 best thing to do guys is create an

**Second subtitle**

và nơi đã làm điều đó ngay tại đây  
 và nói  
 chế độ xem DOT đăng nhập bằng email và  
 mật khẩu  
 và bởi vì điều này đang sử dụng async đang  
 chờ

các bạn, chúng tôi thực sự cần phải thực  
 hiện điều này trong một  
 nhiệm vụ và nói thử trọng số  
 viewmodel.signin bằng email và mật khẩu  
 vì vậy đó là một cách để thực hiện điều đó  
 uh về cách chúng tôi thực hiện  
 mô hình chế độ xem như chức năng này rất  
 tốt

ở đây khi chúng tôi nhấp vào nút mà chúng  
 tôi muốn, chúng  
 tôi muốn bạn biết gọi  
 chức năng đăng nhập này trên chế độ xem  
 của chúng tôi nhưng  
 điều này sẽ gây khó chịu vì  
 nếu bạn khởi tạo nó như một đối tượng  
 trạng thái

bởi vì chúng tôi cũng cần điều này trong  
 chế độ xem đăng ký của chúng tôi và những  
 gì bạn  
 không muốn làm là tạo một  
 phiên bản khác của mô hình chế độ xem tại  
 đây và chỉ  
 có một loạt các phiên bản khác nhau của  
 mô hình chế độ xem của bạn trên tất cả ứng  
 dụng, vì vậy  
 bạn có thể đồng ý tạo một

mô hình xem xác thực đối tượng được quan  
 sát nhưng sau đó khi chúng tôi tạo chế  
 độ xem đăng ký, chúng tôi cần  
 khởi tạo nó với mô hình xem  
 và điều đó cũng gây khó chịu vì vậy  
 điều tốt nhất nên làm là tạo một



**Main subtitle**

environment object here and essentially an environment object gets initialized in one place and can be used in multiple

search or across multiple surfaces in the application which is what we want here because we also need to use this

inside of our profile view as well and we're also going to see that we want to use it in our content view as well and

we'll go we'll go over how exactly but

we're using we need access to stuff

about the user in every single surface of our app right now the authentication

flow the profile view and the main root view the content View

so the best thing to do is to create an environment object which is what we're

going to do we're going to say at State

object VAR view model equals auth view model

and here we are going to set it as an environment object on

our content View

and this way it's only initialized in one place

and guys I do also have tutorials on the differences between State objects and observed objects and environment objects

so make sure you check that out the link

**Second subtitle**

đối tượng môi trường ở đây và về cơ bản là một đối tượng môi trường được khởi tạo ở một nơi và có thể được sử dụng trong nhiều

tìm kiếm hoặc trên nhiều bề mặt trong ứng dụng, đó là điều chúng tôi muốn ở đây vì chúng tôi cũng cần sử dụng điều này

bên trong chế độ xem hồ sơ của mình và chúng tôi cũng sẽ thấy rằng chúng tôi muốn sử dụng nó cũng trong chế độ xem nội dung của chúng tôi và

chúng tôi sẽ tiếp tục, chúng tôi sẽ xem xét chính xác cách thức nhưng chúng tôi

đang sử dụng, chúng tôi cần quyền truy cập vào nội dung

về người dùng trong mọi bề mặt

của ứng dụng của chúng tôi ngay bây giờ luồng xác thực chế

độ xem hồ sơ và giao diện chính gốc xem nội dung Xem

vì vậy điều tốt nhất cần làm là tạo một đối tượng môi trường, đó là những gì chúng ta

sẽ làm, chúng ta sẽ nói ở Đối tượng trạng thái

Mô hình chế độ xem VAR bằng mô hình chế độ xem auth

và ở đây chúng ta sẽ thiết lập nó như một đối tượng môi trường trên Chế độ

xem nội dung của chúng tôi

và theo cách này, nó chỉ được khởi tạo ở một nơi

và các bạn, tôi cũng có hướng dẫn về sự khác biệt giữa các đối tượng Trạng thái và các đối tượng được quan sát và các đối tượng môi trường,

vì vậy hãy đảm bảo bạn kiểm tra xem liên kết

**Main subtitle**

is in the description for all of that as

well it's very important to understand the differences between these things if you're going to build apps using Swift UI one of the main issues I see with new

Swift UI developers is they don't understand how swift UI works as a whole so when it comes to building a complex application they have no idea what they're doing you can build out little screens like this but when it comes to

navigating between a bunch of different views and how everything works at a high

level people get very confused and they

get lost in the saw so it's very important to understand this stuff but anyway

now let's go back to our login View and

we're going to replace this guy with environment objects

and you guys will notice that you can't initialize an environment object right you can only cast it and that's because once again environment objects only get

initialized once

so this is all good and you guys will notice here with the way that we had this set up previously it would ask us

to patch that view model in which is sort of annoying so once again we can

**Second subtitle**

có trong phần mô tả cho tất cả những điều đó

cũng rất quan trọng để hiểu

sự khác biệt giữa những điều này nếu bạn định xây dựng ứng dụng bằng Swift UI, một trong những vấn đề chính mà tôi thấy với

các nhà phát triển Swift UI mới là họ không hiểu Swift UI hoạt động như thế nào Vì

vậy, khi nói đến việc xây dựng một ứng dụng phức tạp, họ không biết

họ đang làm gì, bạn có thể tạo ra

những màn hình nhỏ như thế này nhưng khi nói đến việc

điều hướng giữa một loạt

các chế độ xem khác nhau và cách mọi thứ hoạt động ở

cấp độ cao, mọi người sẽ rất lo lắng. bối rối và họ

bị lạc trong cái cửa, vì vậy điều rất

quan trọng là phải hiểu nội dung này nhưng dù sao đi nữa,

bây giờ chúng ta hãy quay lại Chế độ xem đăng nhập của chúng ta và

chúng ta sẽ thay thế anh chàng này bằng các đối tượng môi trường

và các bạn sẽ nhận thấy rằng bạn không thể khởi tạo một đúng đối tượng môi trường, bạn chỉ có thể sử dụng nó và đó là bởi vì một lần nữa các đối tượng môi trường chỉ được

khởi tạo một lần

nên điều này rất tốt và các bạn sẽ

chú ý ở đây với cách chúng tôi đã

thiết lập điều này trước đó, nó sẽ yêu cầu chúng tôi

và mô hình chế độ xem đó trong điều này hơi khó chịu nên một lần nữa chúng ta có thể

**Main subtitle**

just go here and say at environment  
object VAR oops  
G model auth view model and now it just  
reads it from the environment that we're  
in right so we only initialized it in  
this one place and now both of these  
views can use that view model

accordingly and it's also using the same

instance of that so it's only a single

instance of the view model created I

know I'm harping on this but it's very  
important to understand and as opposed  
to creating multiple instances of the  
view model and if you guys are having  
trouble understanding what I mean there  
like I said go watch the videos  
um did I mention about the difference  
between State objects and observed  
objects and also brush up on your uh

object oriented programming principles

um all that stuff is hard but it's  
necessary if you want to be a good coder

okay so now back here  
when we want to sign our user up we're

going to say task  
try await view model  
DOT sign or create user sorry

with email  
sorry guys email  
password is our password

**Second subtitle**

vào đây và nói  
đối tượng môi trường VAR oops  
G model auth view model và bây giờ nó chỉ  
đọc nó từ môi trường mà chúng ta đang  
ở nên chúng ta chỉ khởi tạo nó ở  
một nơi này và bây giờ cả hai  
chế độ xem này đều có thể sử dụng mô  
hình chế độ xem đó  
tương ứng và nó cũng đang sử dụng cùng  
một  
phiên bản của chế độ xem đó vì vậy nó chỉ  
là một  
phiên bản duy nhất của mô hình chế độ xem  
được tạo. Tôi  
biết tôi đang nói về điều này nhưng điều rất  
quan trọng là phải hiểu và trái ngược  
với việc tạo nhiều phiên bản của  
mô hình khung nhìn và nếu các bạn gặp  
khó khăn trong việc hiểu ý của tôi ở đó  
như tôi đã nói, hãy xem video  
ừm, tôi đã đề cập đến sự khác biệt  
giữa các đối tượng Trạng thái và  
các đối tượng được quan sát và cũng xem  
lại  
các nguyên tắc lập trình hướng đối tượng  
của bạn  
ừm tất cả những thứ đó đều khó nhưng nó  
cần thiết nếu bạn muốn trở thành một lập  
trình viên  
giỏi, vì vậy bây giờ hãy quay lại đây  
khi chúng tôi muốn đăng ký người dùng của  
mình, chúng tôi  
sẽ nói nhiệm vụ hãy  
thử chờ xem  
dấu DOT của mô hình hoặc tạo người dùng  
xin lỗi  
bằng email  
xin lỗi các bạn  
mật khẩu email là mật khẩu

**Main subtitle**

full name is our full name for some

reason with the try away stuff the Swift  
UI autocomplete with functions sucks and

it kind of sticks you like that  
sometimes but not a big deal we got it

done and now let me go and just add some

print statements  
to these functions in our auth view

model  
so that we can make sure that this is

all working correctly  
and let's actually run our app guys so

we haven't done that yet  
so I'm going to select iPhone 14 I have  
the completed app on the iPhone 14 pro  
and let's just make sure all this stuff  
is working

I actually think  
it's going to open up to just our  
content view which is this which we

don't want right  
um

so I'm going to actually go over with  
you guys how we want to set this content

view up now because we've built out this  
login view in our profile view and stuff

but our app is launching the content

view on initialization as we see here if

**Second subtitle**

tên đầy đủ của chúng tôi là tên đầy đủ của  
chúng tôi vì một

lý do nào đó với việc dùng thử công cụ  
tự động hoàn thành giao diện người dùng  
Swift với các chức năng tệ hại và

đôi khi bạn thích điều đó nhưng không phải  
là vấn đề lớn, chúng tôi đã

hoàn thành và bây giờ hãy để tôi đi và chỉ  
cần thêm một số

câu lệnh in

vào các chức năng này trong mô hình chế  
độ xem xác thực của chúng tôi

để chúng tôi có thể đảm bảo rằng tất cả  
điều này

đang hoạt động chính xác

và hãy thực sự chạy ứng dụng của chúng  
tôi, vì

chúng tôi chưa thực hiện điều đó

nên tôi sẽ chọn iPhone 14 I có

ứng dụng hoàn chỉnh trên iPhone 14 pro  
và hãy đảm bảo rằng tất cả những thứ này  
đang hoạt động.

Tôi thực sự nghĩ rằng

nó sẽ chỉ mở ra cho

chế độ xem nội dung của chúng tôi, đây là  
điều mà chúng tôi

không muốn đúng không,

vì vậy tôi sẽ thực sự Hãy cùng

các bạn xem chúng tôi muốn thiết lập chế  
độ xem nội dung này như thế nào

vì chúng tôi đã xây dựng

chế độ xem đăng nhập này trong chế độ  
xem hồ sơ và nội dung của chúng tôi

nhưng ứng dụng của chúng tôi đang khởi  
chạy

chế độ xem nội dung khi khởi tạo như  
chúng tôi thấy ở đây nếu

**Main subtitle**

we go to the app file so basically we

want to just add some presentation logic  
I like to call it to help us figure out  
where we want to go and basically the  
logic behind that is if we have a user  
logged in we take them to the profile  
view if we don't we take them to the  
authentication flow or the login View  
and we are going to use our  
authentication view model to help us do  
that so this is literally just going to  
open up to this Hello World stuff so  
let's go back and go environment object  
of our viewmodel  
which is our auth view model  
and we're going to replace all this  
stuff we're going to say group  
because we're going to have a group of  
views here we're going to say if  
view model  
dot user session  
is not equal to nil  
and else  
so see if you guys can figure out how to  
fill this in right and we'll talk about  
this property in a second  
but basically if there is a user session  
there right if it's not equal to nil  
we're going to send them to the profile  
view

**Second subtitle**

chúng tôi truy cập ứng dụng vì vậy về cơ  
bản, chúng tôi  
chỉ muốn thêm một số logic trình bày mà  
tôi muốn gọi nó để giúp chúng tôi tìm ra  
nơi chúng tôi muốn đến và về cơ bản  
logic đằng sau đó là nếu chúng tôi có người  
dùng  
đăng nhập, chúng tôi sẽ đưa họ đến chế độ  
xem hồ sơ  
nếu chúng tôi không Chúng ta không đưa  
chúng đến  
luồng xác thực hoặc Chế độ xem đăng nhập  
và chúng tôi sẽ sử dụng  
mô hình chế độ xem xác thực của mình để  
giúp chúng tôi làm  
điều đó, vì vậy điều này thực sự sẽ  
mở ra nội dung Hello World này, vì vậy  
hãy quay lại và chuyển sang đối tượng môi  
trường  
của chế độ xem của chúng tôi,  
đây là mô hình chế độ xem xác thực của  
chúng tôi  
và chúng tôi sẽ thay thế tất cả  
những thứ này mà chúng tôi sẽ nói là nhóm  
vì chúng tôi sẽ có một nhóm chế  
độ xem ở đây chúng tôi sẽ nói nếu chế độ  
xem mô hình  
chấm phiên người dùng  
không bằng 0  
và khác,  
vì vậy hãy xem liệu các bạn có thể tìm ra  
cách  
điền đúng không và chúng ta sẽ nói về  
thuộc tính này sau một giây  
nhưng về cơ bản nếu có một phiên người  
dùng  
ở đó ngay nếu nó không bằng 0 thì  
chúng ta' chúng ta sẽ gửi chúng đến chế độ  
xem hồ sơ

**Main subtitle**

and if it is equal to nil we're going to  
send them to the login View  
so now let's run the app again and see  
  
where it takes us so we obviously  
  
haven't signed anyone up yet so this  
user session property should be nil  
  
and it's going to take us right to the  
login view which looks great and if we  
  
hit that sign in button guys we notice  
  
that we're getting our print statement  
back which is coming from our view model  
which is looking really good and we can  
also hop over to our sign up page and if  
we hit that it says create user so this  
  
is looking awesome so far  
um now we're going to actually get into  
  
the implementation of these functions  
and how to use the Firebase code to  
actually you know create our users and  
  
sign them up and also then go over how  
we need to make things work within our  
  
app so that the navigation works  
  
correctly for example after I log in  
  
I want it to take me to the profile view  
I wanted to fetch my user's information  
and display it on my in my profile and  
all that good stuff so that's the next  
  
step guys this is really going to start

**Second subtitle**

và nếu nó bằng không, chúng tôi sẽ  
gửi chúng đến chế độ xem đăng nhập,  
vì vậy bây giờ hãy chạy lại ứng dụng và  
xem  
  
nó sẽ đưa chúng ta đến đâu vì vậy rõ ràng  
là chúng ta  
  
chưa đăng ký bất kỳ ai vì vậy  
thuộc tính phiên người dùng này phải bằng  
không  
  
và nó sẽ đưa chúng ta đến ngay  
chế độ xem đăng nhập trông rất tuyệt và  
nếu chúng ta  
  
nhấn vào nút đăng nhập đó, chúng ta sẽ  
nhận thấy  
  
rằng chúng ta sẽ nhận lại được câu lệnh in  
đến từ mô hình chế độ xem của chúng ta.  
trông rất đẹp và chúng tôi cũng có thể  
chuyển đến trang đăng ký của mình và nếu  
chúng tôi nhấn vào đó, nó sẽ báo tạo người  
dùng, vì vậy trang này  
trông thật tuyệt vời cho đến nay,  
ừm, bây giờ chúng tôi sẽ thực sự bắt tay  
vào  
  
triển khai các chức năng này  
và cách sử dụng mã Firebase để  
thực sự bạn biết tạo người dùng của chúng  
tôi và  
  
đăng ký họ, sau đó xem xét cách  
chúng tôi cần làm cho mọi thứ hoạt động  
trong  
ứng dụng của mình để điều hướng hoạt  
động  
  
chính xác, chẳng hạn như sau khi tôi đăng  
nhập,  
  
tôi muốn nó đưa tôi đến hồ sơ chế độ xem  
Tôi muốn lấy thông tin người dùng của mình  
và hiển thị thông tin đó trên hồ sơ của tôi và  
tất cả những nội dung hay đó, vì vậy đó là  
bước tiếp theo  
  
các bạn, điều này thực sự sẽ bắt đầu

**Main subtitle**

coming together really really soon after that we're going to do all the nice fancy form validation stuff and you guys will have a complete authentication system for your app whenever you want to

drop it in there which is absolutely awesome so let's continue so let's get started with filling out this create user function guys so basically we want we want to fill this

field out here hit sign up have it create a user for us in the back end with Firebase and then have it navigate to our profile view once we create that

user so here's what this is going to look like we are going to start this off by writing a do block and a catch block as well and what we're going to say here is let result equal try await um auth

.auth dot create user with email and password and you guys will see that this utilizes the async throws functionality and gives us back some sort of auth data

result if you guys have done this before with the previous API it was all using completion handlers and as you can see those get really messy which is why we're using async await

here and once again if this is totally

**Second subtitle**

kết hợp với nhau thực sự ngay sau đó chúng ta sẽ làm tất cả những điều thú vị công cụ xác thực biểu mẫu và các bạn sẽ có một hệ thống xác thực hoàn chỉnh cho ứng dụng của mình bất cứ khi nào bạn muốn

thả nó vào đó, điều này hoàn toàn tuyệt vời, vì vậy hãy tiếp tục, vì vậy hãy bắt đầu điền vào chức năng tạo người dùng này, vì vậy về cơ bản chúng tôi muốn chúng tôi muốn điền

trường này ở đây nhấn đăng ký để nó tạo người dùng cho chúng tôi ở phần cuối với Firebase và sau đó để nó điều hướng đến chế độ xem hồ sơ của chúng tôi sau khi chúng tôi tạo

người dùng đó, vì vậy, đây là giao diện chúng tôi sẽ bắt đầu bằng cách viết một khối do và một khối catch và những gì chúng ta sẽ nói ở đây là để kết quả bằng try await um auth

.auth dot tạo người dùng bằng email và mật khẩu và các bạn sẽ thấy rằng điều này sử dụng chức năng ném async và trả lại cho chúng tôi một số loại kết quả dữ liệu xác thực

nếu các bạn đã thực hiện điều này trước đây với API trước đó, tất cả đều sử dụng trình xử lý hoàn thành và như bạn có thể thấy chúng thực sự lộn xộn, đó là lý do tại sao chúng tôi sử dụng async đang chờ

ở đây và một lần nữa nếu điều này hoàn toàn

**Main subtitle**

confusing to you make sure you check out

my async await tutorial and the link is

in the description for that so we're

going to say create user with email and password so basically what this allows

us to do guys is await the result that gives that this gives us back so we

don't need to use completion blocks it's just going to get stored in this result constant that we created there and this allows us to write asynchronous code in

a synchronous way right we can just do it line by line without the use of completion blocks and handlers which is really really nice so now what we're

going to do is set this user session

property and we'll talk about how this is going to come into play in a little bit so we're going to say `self.user`

session

`equals results dot user` and you guys will see that this is a user object that

comes back from this result and that's where we're going to set our user

session to be

now we need to create our actual user data model and once again those are

different right so this is our user data

**Second subtitle**

khó hiểu với bạn, hãy chắc chắn rằng bạn đã xem

hướng dẫn chờ async của tôi và liên kết nằm

trong phần mô tả cho điều đó, vì vậy chúng tôi

sẽ nói tạo người dùng bằng email và mật khẩu, vì vậy về cơ bản điều này cho phép

chúng tôi làm là chờ kết quả

cho rằng điều này mang lại cho chúng tôi vì vậy chúng tôi

không cần sử dụng các khối hoàn thành, nó sẽ được lưu trữ trong hằng số kết quả mà chúng tôi đã tạo ở đó và điều này cho phép chúng tôi viết mã không đồng bộ theo

cách đồng bộ, chúng tôi có thể

thực hiện theo dòng mà không cần sử dụng các khối hoàn thành và trình xử lý, điều này thực sự rất hay, vì vậy bây giờ những gì chúng ta

sẽ làm là thiết lập thuộc tính phiên người dùng này

và chúng ta sẽ nói về cách nó

sẽ hoạt động trong một thời

gian ngắn vì vậy chúng ta' sẽ nói phiên `self.user`

bằng kết quả dot user và các bạn

sẽ thấy rằng đây là một đối tượng người dùng

quay lại từ kết quả này và đó là

nơi chúng ta sẽ đặt phiên người dùng của mình,

bây giờ chúng ta cần tạo thực thể của mình

mô hình dữ liệu người dùng và một lần nữa, chúng

khác nhau đúng không, vì vậy đây là mô



**Main subtitle**

model and this is the Firebase user object so we'll go over a little bit more in detail how they're different for now let's just go ahead and get this created we're going to say let user equal user and the ID is going to be the result

Dot user.uid

so this comes from Firebase so when we create a user in the back end with Firebase we get this uid property back and then we're going to use the full name and email that we get from um our form here to actually create our user object so as you guys can see here that this user object the one that we created has additional properties like the user's full name their email address and all that good stuff whereas the Firebase user object is just like a generic user object that Firebase creates for us on their back end that gives us things like the uid and all of that so the user object that you that we've built in our model folder is where you would put all of the other fields that you want associated with a user like maybe their phone number or their

**Second subtitle**

hình dữ liệu người dùng của chúng tôi và đây là đối tượng người dùng Firebase, vì vậy chúng tôi sẽ tìm hiểu chi tiết hơn một chút về sự khác biệt của chúng

bây giờ, hãy tiếp tục và hiểu điều này được tạo, chúng tôi sẽ nói rằng hãy để người dùng bình đẳng với người dùng và ID sẽ là kết quả

Dot user.uid

vì vậy cái này đến từ Firebase nên khi chúng tôi tạo người dùng ở mặt sau với Firebase, chúng tôi sẽ lấy lại thuộc tính uid này và sau đó chúng tôi Chúng tôi sẽ sử dụng tên đầy đủ và email mà chúng tôi nhận được từ biểu mẫu của chúng tôi ở đây để thực sự tạo đối tượng người dùng của chúng tôi để các bạn có thể thấy ở đây rằng đối tượng người dùng này đối tượng mà chúng tôi đã tạo có các thuộc tính bổ sung như tên đầy đủ của người dùng email của họ địa chỉ và tất cả nội dung hay đó trong khi đối tượng người dùng Firebase giống như đối tượng người dùng chung mà Firebase tạo cho chúng tôi ở mặt sau cung cấp cho chúng tôi những thứ như uid và tất cả những thứ đó để đối tượng người dùng mà chúng tôi đã xây dựng thư mục mô hình của chúng tôi là nơi bạn sẽ đặt tất cả các trường khác mà bạn muốn liên kết với người dùng, chẳng hạn như số điện thoại hoặc

**Main subtitle**

username or you know whatever their date

of birth whatever it may be all of that  
stuff that's not encompassed in the  
Firebase user object this is just what

we get back when we create a user on the

Firebase backend so that's the  
difference between those two things so  
then we're going to encode this user so  
we're going to say guard let encoded

user equal

firestore.encoder and I think we need to

go up here

and import

Firebase firestore Swift so now we can

go here and say encoder

dot encode

user and this is just helping us utilize  
the uh Codable protocol guys so we're

going to say else return

and then we're going to say

try await

firestore

.firestore.collection

users dot document

user.uid or dot ID

dot set data and then we can set that  
encoded user there

so that's looking good and then in our

catch block we can just say for now

print debug

**Second subtitle**

tên người dùng của họ hoặc bạn biết bất kể  
ngày sinh của họ,

bất kể đó có thể là tất cả

những nội dung đó không có trong

Firebase đối tượng người dùng, đây chỉ là  
những gì

chúng tôi nhận được khi chúng tôi tạo người  
dùng trên

phần phụ trợ của Firebase, vì vậy đó là

sự khác biệt giữa hai thứ đó, vì vậy

chúng tôi sẽ mã hóa người dùng này, vì vậy

chúng tôi sẽ nói rằng bảo vệ hãy để người  
dùng được mã hóa bằng

firestore. bộ mã hóa và tôi nghĩ rằng chúng  
ta cần phải

lên đây

và nhập

Firebase firestore Swift để bây giờ chúng ta  
có thể

vào đây và nói người dùng mã

hóa dấu chấm mã hóa

và điều này chỉ giúp chúng ta sử dụng

các giao thức có thể mã hóa uh, vì vậy  
chúng ta

sẽ nói khác trở lại

và sau đó chúng tôi sẽ nói

hãy thử chờ

người dùng firestore .firestore.collection  
chấm tài liệu

user.uid hoặc chấm ID

chấm dữ liệu và sau đó chúng tôi có thể đặt  
người dùng được mã hóa đó ở đó

sao cho trông ổn và sau đó trong khối bắt  
của chúng tôi,

chúng tôi chỉ có thể nói ngay bây giờ

gỡ lỗi in

**Main subtitle**

failed to create user with error

error.localize description

okay

I think we need to add a try to this

let's see yep

stick a try in there

and actually delete that guys

so

let's see if that works out for us

okay so really quickly just to go over

this function one more time guys it's an

asynchronous function that can

potentially throw an error that's what's

going to happen in the catch block if

anything goes wrong so here we try to

create a user using the Firebase code

this is what we get back from the

Firebase package manager we installed

and we're going to await the result of

that and store it in this property once

we get it back

if we get a successful result back we're

going to set our user session property

and then we're going to create our user

object and then encode that object

through the codable protocol and then

upload that data to firestore so then if

anything goes wrong we're going to hit

this catch block here and it will print

out an error description for us so let's

**Second subtitle**

không tạo được người dùng có lỗi

error.localize description

okay

Tôi nghĩ chúng ta cần thêm một bản thử vào cái này,

hãy xem yep, hãy

thử vào đó

và thực sự xóa những người đó

để

xem liệu điều đó có phù hợp với chúng ta

không, hãy thực sự nhanh chóng. để xem lại

chức năng này một lần nữa nhé các bạn, đây là một

chức năng không đồng bộ có khả năng

gây ra lỗi, đó là điều

sẽ xảy ra trong khối catch nếu có

bất kỳ sự cố nào, vì vậy ở đây chúng tôi cố gắng

tạo một người dùng bằng mã Firebase,

đây là những gì chúng tôi nhận được từ

trình quản lý gói Firebase mà chúng tôi đã cài đặt

và chúng tôi sẽ chờ kết quả của

việc đó và lưu trữ nó trong thuộc tính này sau khi

chúng tôi lấy lại

nếu chúng tôi nhận được kết quả thành công, chúng tôi

sẽ đặt thuộc tính phiên người dùng của mình

và sau đó chúng tôi' chúng ta sẽ tạo

đối tượng người dùng và sau đó mã hóa đối tượng đó

thông qua giao thức có thể mã hóa, sau đó

tải dữ liệu đó lên firestore, vì vậy nếu có

bất kỳ sự cố nào xảy ra, chúng ta sẽ nhấn

khối bắt này ở đây và nó sẽ in

ra mô tả lỗi cho chúng ta. hãy

**Main subtitle**

go ahead and run our app again and if

you guys remember we are already calling this function back inside of our registration view so we are good to go

here right so let's go ahead and just run this and see if we can actually

create a user

and now I also want to explain what's going to happen with this user session

property and we'll see what happens once we actually successfully sign a user up

I just want to see if some of our error messages are working and stuff like that

okay so let's go and try to sign up

ooh uh we actually need to configure Firebase guys so we forgot to do that

um

so let's see we I believe need to go to the app file

and here we need to say init

and go up to the top and import Firebase

and we're going to say Firebase app dot configure

and let's run our project again

so this is something that we have to do when our app launches we have to call

this configure function from Firebase that's why it crashed just now and uh

this should all be good to go now so

**Second subtitle**

tiếp tục và chạy lại ứng dụng của chúng ta và nếu

các bạn còn nhớ thì chúng ta đã gọi hàm này trở lại bên trong

chế độ xem đăng ký của chúng ta, vì vậy chúng ta có thể tiếp tục ngay

tại đây, vì vậy hãy tiếp tục và chỉ chạy ứng dụng này và xem liệu chúng ta có thể thực sự

tạo một người dùng hay không

và bây giờ tôi cũng muốn giải thích điều gì sẽ xảy ra với thuộc tính phiên người dùng này

và chúng ta sẽ xem điều gì sẽ xảy ra sau khi chúng tôi thực sự đăng ký thành công một người dùng.

Tôi chỉ muốn xem liệu một số thông báo lỗi của chúng tôi

có hoạt động không và những thứ tương tự như vậy

có ổn không. chúng ta hãy đi và cố gắng đăng ký

ooh uh chúng ta thực sự cần định cấu hình Firebase các bạn nên chúng ta quên làm điều đó

ừm để

xem chúng ta tôi tin rằng cần phải truy cập tệp ứng dụng

và ở đây chúng ta cần nói init

và đi lên trên cùng và nhập Firebase

và chúng tôi sẽ nói

cấu hình dấu chấm ứng dụng Firebase

và hãy chạy lại dự án của chúng tôi, vì vậy đây là điều chúng tôi phải làm

khi ứng dụng của chúng tôi khởi chạy, chúng tôi phải gọi

chức năng cấu hình này từ Firebase,

đó là lý do tại sao nó bị sập ngay bây giờ và uh

cái này bây giờ tất cả sẽ ổn thôi, vì vậy

**Main subtitle**

let's hit sign up and look at that we're  
  
getting an error message here fail to  
create user with error the password must  
be six characters long or more right so  
now let's actually try to create a user  
  
I'll just say test gmail.com  
test password is one two three four five  
  
six I usually just do six lowercase cues  
  
guys it's the easiest thing to remember  
we don't have to do anything with this  
confirmed password field just yet let's  
just go ahead and see if what happens  
when we hit sign up  
ooh okay so one we notice that it takes  
  
us to our profile view right away  
  
um so I want us to go over how that is  
  
happening  
so if we go to our content view we  
  
notice that the view that gets presented  
here is based on this user session  
  
property right so if it's not nil it  
  
shows the profile view so back in the  
auth view model because this is a  
  
published property right and we'll fix  
  
this error that you see in purple here  
in just a second guys because this is a  
published property whenever this user

**Second subtitle**

hãy nhấn đăng ký và xem xét rằng chúng tôi  
đang  
  
nhận được thông báo lỗi ở đây không thể  
tạo người dùng có lỗi, mật khẩu phải  
dài sáu ký tự trở lên, vì vậy  
bây giờ chúng ta hãy thử tạo một người  
dùng.  
Tôi sẽ chỉ nói kiểm tra  
mật khẩu kiểm tra gmail.com là một hai ba  
bốn năm  
sáu Tôi thường chỉ thực hiện sáu dấu hiệu  
viết thường  
các bạn, đó là điều dễ nhớ nhất,  
chúng ta không phải làm bất cứ điều gì với  
trường mật khẩu đã xác nhận này, hãy  
tiếp tục và xem điều gì xảy ra  
khi chúng tôi nhấn đăng ký  
ooh được rồi, vì vậy chúng tôi nhận thấy  
rằng nó đưa  
chúng tôi đến chế độ xem hồ sơ của chúng  
tôi ngay lập tức, vì  
vậy tôi muốn chúng tôi xem điều đó đang  
xảy ra như thế nào  
  
vì vậy nếu chúng tôi chuyển sang chế độ  
xem nội dung của mình, chúng tôi sẽ  
nhận thấy rằng Chế độ xem được trình bày  
ở đây dựa trên quyền thuộc tính phiên  
người dùng này,  
vì vậy nếu không phải là số không, chế độ  
xem  
hồ sơ sẽ hiển thị  
chế độ xem hồ sơ, vì vậy hãy quay lại mô  
hình chế độ xem xác thực vì đây là  
quyền thuộc tính đã xuất bản và chúng tôi  
sẽ khắc phục  
lỗi này mà bạn thấy bằng màu tím tại đây  
chỉ trong một giây vì đây là  
thuộc tính được xuất bản bất cứ khi nào  
phiên người dùng này

**Main subtitle**

session guy gets set it's going to send a notification to my content view to reconfigure itself with this new user session value so at first it was showing the login view because this was nil then once we sign our user up we set that user session property right here so now it has a value so then that gets published to our content view it sends that notification over saying hey some some property changed on The View model right we are listening for changes on The View model something has changed you need to redraw the view or reconfigure the view and now the user session is not nil so it shows the profile view meaning we have a user that that is actually logged in now so let's go back to our auth view model guys and we actually need to go ahead and add something called at main actor here and this is because we need to publish all UI updates back on the main thread so by default all of this asynchronous networking happens on a background thread and we want to make sure that we

**Second subtitle**

được thiết lập, nó sẽ gửi thông báo đến chế độ xem nội dung của tôi để tự cấu hình lại với giá trị phiên người dùng mới này, vì vậy lúc đầu, nó hiển thị chế độ xem đăng nhập vì đây là con số không sau đó, khi chúng tôi đăng ký người dùng của mình, chúng tôi đặt thuộc tính phiên người dùng đó ngay tại đây để bây giờ nó có giá trị, sau đó được xuất bản lên chế độ xem nội dung của chúng tôi, nó sẽ gửi thông báo đó nói rằng này, một số thuộc tính đã thay đổi trên mô hình Chế độ xem, chúng tôi đang lắng nghe thay đổi trên Mô hình Chế độ xem, có gì đó đã thay đổi, bạn cần vẽ lại chế độ xem hoặc định cấu hình lại chế độ xem và bây giờ phiên người dùng không phải là số không nên nó hiển thị chế độ xem hồ sơ, nghĩa là chúng tôi có một người dùng thực sự đã đăng nhập ngay bây giờ, vì vậy hãy quay lại trang của chúng tôi auth view model guys và chúng tôi thực sự cần phải tiếp tục và thêm một cái gì đó được gọi là diễn viên chính ở đây và điều này là do chúng tôi cần xuất bản tất cả các bản cập nhật giao diện người dùng trở lại trên luồng chính để theo mặc định, tất cả mạng không đồng bộ này xảy ra trên một luồng nền và chúng tôi muốn đảm bảo rằng chúng tôi

**Main subtitle**

are publishing our UI changes back on the main thread and that's what this main actor declaration does up there so now let's run our app again and let's uh take a look and see what happens here so it should go back to our login page which is not the functionality we want right because we created an account and what we want to happen is unless we sign out we want it to keep us logged in there so here's what we're going to do we're going to say `self.user session equals auth.auth dot current user` so when our auth viewmodel initializes it's going to check and see if there is a current user so this is functionality we get from Firebase they actually if there's a current user logged in they store that locally on the device they cache that information so if we have someone logged in guess what it's going to say hey this user session has a value and because that has a value based on our presentation logic in the content view it's going to show us the profile view so now that we've done that let's go ahead and see what that looks like

**Second subtitle**

đang xuất bản các thay đổi giao diện người dùng của mình trở lại luồng chính và đó là điều mà phần khai báo diễn viên chính này thực hiện ở đó, vì vậy bây giờ hãy chạy lại ứng dụng của chúng tôi và hãy xem điều gì sẽ xảy ra ở đây để nó quay lại trang đăng nhập của chúng tôi không phải là chức năng mà chúng tôi muốn vì chúng tôi đã tạo một tài khoản và những gì chúng tôi muốn xảy ra là trừ khi chúng tôi đăng xuất, chúng tôi muốn nó giữ cho chúng tôi đăng nhập ở đó, vì vậy đây là những gì chúng tôi sẽ làm chúng tôi sẽ làm giả sử phiên `self.user` bằng `auth.auth` chấm người dùng hiện tại, vì vậy khi chế độ xem `auth` của chúng tôi khởi chạy, nó sẽ kiểm tra và xem liệu có người dùng hiện tại hay không, vì vậy đây là chức năng chúng tôi nhận được từ Firebase, họ thực sự lưu trữ nếu có người dùng hiện tại đăng nhập cục bộ trên thiết bị, họ lưu trữ thông tin đó vào bộ đệm, vì vậy nếu chúng tôi có ai đó đăng nhập, hãy đoán xem nó sẽ nói gì, phiên người dùng này có giá trị và vì phiên đó có giá trị dựa trên logic trình bày của chúng tôi trong chế độ xem nội dung, nó sẽ hiển thị cho chúng tôi hồ sơ xem vì vậy bây giờ chúng ta đã hoàn thành việc đó, hãy tiếp tục và xem nó trông như thế nào

**Main subtitle**

this should open up back to the profile view

and we notice that that's **exactly** what

we would expect right so the next step that I want us to go over guys is how to

actually fetch the user's information because this is still just displaying static information from our mock user

data so

um next up in the auth view model

what I want us to do is go over how we're going to fetch this user right here and then we're going to go over how to sign our user out and how to log our user in so we'll have that full authentication flow set up in this application here so fetching the user is

going to be pretty simple we're going to say guard let uid equal

`auth.auth.currentUser.uid` else return so this is just going to get the current user's ID and let me actually go ahead

and pull up

um

our Firebase app really quickly

yeah see this is the step we had forgot to do

and let's go to authentication I just want to show you guys that that test at gmail.com is actually registered here's the user ID and then if we go to our

**Second subtitle**

sẽ mở trở lại chế độ xem hồ sơ

và chúng tôi nhận thấy rằng đó chính xác là những gì

chúng ta mong đợi, vì vậy bước tiếp theo mà tôi muốn chúng ta thực hiện qua các bạn là cách

thực sự tìm nạp thông tin của người dùng vì điều này vẫn chỉ hiển thị

thông tin tĩnh từ dữ liệu người dùng giả của chúng tôi,

vì vậy,

tiếp theo trong mô hình chế độ xem xác thực,

điều tôi muốn chúng tôi làm là tìm hiểu cách chúng tôi sẽ tìm nạp người dùng này ngay tại đây và sau đó chúng ta sẽ tìm hiểu cách đăng xuất người dùng và cách đăng nhập người dùng để chúng ta sẽ

thiết lập luồng xác thực đầy đủ trong

ứng dụng này tại đây để việc tìm nạp người dùng

sẽ khá đơn giản, chúng ta sẽ

nói guard hãy để uid bằng

`auth.auth.currentUser.uid` khác quay lại

vì vậy điều này sẽ lấy

ID của người dùng hiện tại và để tôi thực sự tiếp tục

và mở

ứng dụng Firebase của chúng tôi lên thật nhanh vâng,

hãy xem đây là bước chúng ta đã làm quên làm

và hãy chuyển sang phần xác thực. Tôi chỉ muốn cho các bạn thấy rằng bài kiểm tra tại gmail.com thực sự đã được đăng ký, đây là ID người dùng và sau đó nếu chúng ta truy cập



**Main subtitle**

firestore database we can see that here

we have a user's collection and this is

where all of the user's metadata is

stored so their email their full name  
their ID and once again you can store as  
many properties you want about your user

it's all relative to what you need for

your individual application so this is  
the information that we need to fetch  
now guys we need to reach out to  
Firebase go and find our user ID and  
then get all of this information and map

that into our user object so that we can

display it on this screen here so that's

what the purpose of this fetch user  
function is and


um

I want to explain this line of code here  
as well on line 31 this is how we got

that information uploaded to Firebase we  
said

firestore.firestore.collectionusers and  
that takes us to this spot right here

where we create this user's collection

then we  create this document with the  
user's ID so you guys notice that that's  
the user ID it matches up directly with  
what we see here here and in our  
authentication section as well and then

**Second subtitle**

cơ sở dữ liệu firestore của mình, chúng ta  
có thể thấy rằng ở đây

chúng ta có bộ sưu tập của người dùng và  
cái này là

nơi lưu trữ tất cả siêu dữ liệu của người  
dùng

để email của họ có tên đầy đủ

ID của họ và một lần nữa, bạn có thể lưu trữ  
bao nhiêu thuộc tính bạn muốn về người  
dùng của mình,

tất cả đều liên quan đến những gì bạn cần  
cho

ứng dụng cá nhân của mình, vì vậy đây là  
thông tin mà chúng tôi cần để tìm nạp ngay  
bây giờ, các bạn, chúng tôi cần liên hệ với  
Firebase và tìm ID người dùng của mình,  
sau đó lấy tất cả thông tin này và ánh xạ  
thông tin

đó vào đối tượng người dùng của chúng tôi  
để chúng tôi có thể

hiển thị nó trên màn hình này tại đây, vì vậy  
đó là

mục đích của việc tìm nạp người dùng này  
chức năng là và

ừm,

tôi muốn giải thích dòng mã này ở đây

cũng như trên dòng 31, đây là cách chúng  
tôi

tải thông tin đó lên Firebase, chúng tôi đã  
nói

firestore.firestore.collectionusers và

điều đó đưa chúng tôi đến vị trí này ngay tại  
đây,

nơi chúng tôi tạo bộ sưu tập của người  
dùng này

sau đó chúng tôi tạo tài liệu này với

ID của người dùng để các bạn lưu ý rằng

ID người dùng đó khớp trực tiếp với

những gì chúng tôi thấy ở đây và trong

phần xác thực của chúng tôi, sau đó

**Main subtitle**

we set the user's data and this is our

user object we encoded it right here and

then uploaded that information to

Firebase so we've successfully saved our

users information in a back end so

that's awesome now we want to retrieve

that information I just wanted to make

it clear that that's what we're doing

here I've done this so many times guys

that this is just second nature to me so

I'm really trying to break it down as

with as much detail as possible for you

so let's go ahead and see what we need

to do now we're going to say

`firestore.firestore.collection`

`users` Dot document

`uid` and then we're going to say dot get

document

and you guys will notice this returns a

snapshot so we're going to say let

snapshot equal

this guy and we also need to say uh

await

and I believe we need to say try as well

so let's add a try with a question mark

and I personally don't care to handle

the error here

um for fetching the user I just try to

go and get it you can handle the error

**Second subtitle**

chúng tôi đặt dữ liệu của người dùng và đây là

đối tượng người dùng mà chúng tôi đã mã hóa nó ngay tại đây và

sau đó tải thông tin đó lên

Firebase, vì vậy chúng tôi đã lưu thành công thông tin người dùng của mình vào phần phụ trợ, vì

vậy thật tuyệt vời khi bây giờ chúng tôi muốn truy xuất

thông tin đó. Tôi chỉ muốn

làm rõ rằng đó là những gì chúng tôi đang làm

ở đây tôi' Tôi đã làm điều này rất nhiều lần và đây chỉ là bản chất thứ hai đối với tôi, vì vậy

tôi thực sự đang cố gắng chia nhỏ nó càng chi tiết càng tốt cho các bạn, vì vậy hãy tiếp tục và xem chúng ta cần làm gì bây giờ. để nói

người dùng `firestore.firestore.collection`

Chăm tài liệu

`uid` và sau đó chúng ta sẽ nói dấu chấm lấy tài liệu

và các bạn sẽ nhận thấy điều này trả về một ảnh chụp nhanh, vì vậy chúng ta sẽ nói hãy để

ảnh chụp nhanh bằng

anh chàng này và chúng ta cũng cần nói uh đang chờ

và tôi tin rằng chúng ta cũng cần phải nói thử,

vì vậy hãy thêm từ thử với một dấu chấm hỏi

và cá nhân tôi không quan tâm đến việc xử lý

lỗi ở đây

ừm để tìm nạp người dùng. Tôi chỉ cố gắng truy cập và nhận lỗi, bạn có thể xử lý lỗi

**Main subtitle**

if you want and implement this do catch

style block there but typically things

don't go wrong here so I typically don't bother with it but now what we need to

do is write a guard statement

because this is an optional try we're not wrapping it within a catch so that's the difference here

and then because we conform to the codable protocol here we can say let

user equal

snapshot dot data and you guys will see

this data as option user.self

and we can actually go here and say self.current

user equals that

so now we are setting this current user

property here

with the information that we get back from let's see why is that yelling at me but the error is not handled let me see what I have in my code really quickly

oh we need to add a try there as well

and actually yep that should work cool so now we have to talk about when

we want to fetch this user data so it's

going to be in a couple places guys

first off it's going to be in our init

function up here so we're going to say

**Second subtitle**

nếu bạn muốn và thực hiện điều này, hãy bắt

kiểu chặn ở đó nhưng thông thường mọi thứ

sẽ không sai ở đây nên tôi thường không bận tâm đến nó nhưng bây giờ điều chúng ta cần

làm là viết một tuyên bố bảo vệ

vì đây là tùy chọn, chúng tôi sẽ thử

không gói nó trong một lần bắt, vì vậy đó là sự khác biệt ở đây

và sau đó bởi vì chúng tôi tuân thủ

giao thức có thể mã hóa ở đây, chúng tôi có thể nói hãy để

dữ liệu dấu chấm chụp nhanh bằng nhau của người dùng và các bạn sẽ thấy

dữ liệu này dưới dạng tùy chọn user.self

và chúng tôi thực sự có thể đến đây và nói self.current

user bằng với nó

nên bây giờ chúng ta đang thiết lập thuộc tính user hiện tại

ở đây

với thông tin mà chúng ta lấy lại được.

Hãy xem tại sao điều đó lại mắng tôi

nhưng lỗi không được xử lý. Hãy

nhANH chóng xem tôi có gì trong mã của mình.

chúng tôi cũng cần thêm một thử nghiệm ở đó

và thực sự vâng, nó sẽ hoạt động

tốt, vì vậy bây giờ chúng tôi phải nói về thời điểm

chúng tôi muốn tìm nạp dữ liệu người dùng này vì vậy nó

sẽ ở một vài nơi

trước tiên mọi người sẽ ở trong init của chúng tôi

chức năng ở đây, vì vậy chúng tôi sẽ nói

**Main subtitle**

task

fetch user or we would say try nope just

await

fetch user

so basically we're going to go and try  
to fetch that user data right away and

let's just go down here and add a print  
statement for that so we're going to say  
print debug current user is

and we would say self dot current user

so let's go ahead and run the app now to

see if we're actually successfully  
getting this data back from Firebase and

if it's successfully decoding into our  
user object and I'll just briefly

explain what I mean by that in a second

here so you guys see that we get back  
our print statement here current user is

optional and we notice that it comes

back or it's printing out as our actual

user object it says Swift UI off

tutorial dot user so that's not just

coming back as like raw data we're  
actually successfully mapping the data  
we get back from Firebase into our user  
object or user model that we have right

here

**Second subtitle**

tác vụ

tim nạp người dùng hoặc chúng tôi sẽ nói  
thử không, chỉ

chờ

tim nạp người dùng

nên về cơ bản, chúng tôi sẽ cố gắng

tim nạp dữ liệu người dùng đó ngay lập tức  
và

chúng ta hãy đi xuống đây và thêm bản in  
tuyên bố cho điều đó, vì vậy chúng tôi sẽ nói  
người dùng hiện tại gỡ lỗi in

và chúng tôi sẽ nói người dùng hiện tại tự  
chấm,

vì vậy, hãy tiếp tục và chạy ứng dụng ngay  
bây giờ để

xem liệu chúng tôi có thực sự

lấy lại thành công dữ liệu này từ Firebase  
hay không và

liệu nó có thành công không giải mã thành  
đối tượng người dùng của chúng tôi và tôi  
sẽ chỉ

giải thích ngắn gọn ý của tôi về điều đó  
trong một giây

ở đây để các bạn thấy rằng chúng tôi lấy lại  
câu lệnh in của mình ở đây người dùng hiện  
tại là

tùy chọn và chúng tôi nhận thấy rằng nó  
quay

lại hoặc nó được in ra dưới dạng của chúng  
tôi

đối tượng người dùng thực tế nó nói Swift  
UI tắt

hướng dẫn người dùng dấu chấm để không  
chỉ

quay lại giống như dữ liệu thô mà chúng tôi  
thực sự đang ánh xạ thành công dữ liệu

chúng tôi nhận được từ Firebase vào

đối tượng người dùng hoặc mô hình người  
dùng mà chúng tôi có ở

đây

**Main subtitle**

so basically that's because we conform to this codable protocol if you guys are confused about that highly recommend just going on YouTube and searching up a quick uh codable protocol tutorial basically it allows us to take incoming raw data or raw Json data and map it into a data object like we see here with our user that's what's known as decoding and then we saw what encoding was when we did this when we encoded our user object so this is the opposite process of that right it takes a user object and encodes it into Json data so that we can upload it to Firebase you can't just upload a user object like this to Firebase it has to be in a raw data format like a key value system or a dictionary system so we're successfully getting that information back which is awesome we're calling that on fetch so if there is a currently logged in user it will go ahead and fetch this stuff if there's not it'll hit this guard statement and it'll stop right there so we won't waste an API call which is also awesome and now I want us to talk really quickly

**Second subtitle**

nên về cơ bản đó là vì chúng tôi tuân thủ giao thức có thể mã hóa này nếu các bạn bối rối về điều đó, chúng tôi khuyên bạn chỉ nên truy cập YouTube và tìm kiếm một hướng dẫn về giao thức có thể mã hóa nhanh, về cơ bản, nó cho phép chúng tôi lấy dữ liệu thô hoặc dữ liệu Json thô và ánh xạ nó vào một đối tượng dữ liệu như chúng tôi hãy xem ở đây với người dùng của chúng tôi, đó là cái được gọi là giải mã và sau đó chúng tôi đã xem mã hóa là gì khi chúng tôi thực hiện việc này khi chúng tôi mã hóa đối tượng người dùng của mình, vì vậy đây là quy trình ngược lại với quyền đó, nó lấy một đối tượng người dùng và mã hóa nó thành dữ liệu Json để chúng tôi có thể tải nó lên Firebase, bạn không thể chỉ tải một đối tượng người dùng như thế này lên Firebase, nó phải ở định dạng dữ liệu thô như hệ thống giá trị khóa hoặc hệ thống từ điển để chúng tôi lấy lại thành công thông tin đó, điều đó thật tuyệt vời, chúng tôi đang gọi nó khi tìm nạp, vì vậy nếu có một người dùng hiện đang đăng nhập, nó sẽ tiếp tục và tìm nạp nội dung này nếu không, nó sẽ chạm vào câu lệnh bảo vệ này và nó sẽ dừng ngay tại đó, vì vậy chúng tôi sẽ không lãng phí lệnh gọi API, đó là cũng rất tuyệt và bây giờ tôi muốn chúng ta nói thật nhanh

**Main subtitle**

about where else we need to put this so

really quickly guys you're going to want to put it at the end of your create user function okay and let me explain why here let me just take a look at my code to make sure everything is looking good yep okay so after we upload the data to Firebase right it won't go into the next line until that process is done and then we're just going to say await fetch user just like that and why is that so the reason for that is because once we create our user we're going to set this user session property right so that's going to help us navigate to the profile screen but we actually need to fetch the data that we just uploaded to Firebase so that it can be properly displayed on the screen and that has to happen when we create our user so we'll create another account in a little bit I want us to go over now so we can see that working but what I want us to do is now go over how to actually inject our user data into this profile screen so let's go back to our profile view

**Second subtitle**

về nơi nào khác mà chúng ta cần đặt cái này thật

nhanh các bạn sẽ muốn đặt nó ở cuối chức năng tạo người dùng của mình

được không và để tôi giải thích tại sao ở đây để tôi chỉ hãy xem mã của tôi để đảm bảo mọi thứ đều ổn, vâng, được rồi, vì vậy sau khi chúng tôi tải dữ liệu lên Firebase ngay, nó sẽ không chuyển sang dòng tiếp theo cho đến khi quá trình đó hoàn tất và sau đó chúng tôi sẽ nói đang chờ

tìm nạp người dùng chỉ như vậy và tại sao lại như vậy lý do là bởi vì khi chúng tôi tạo người dùng của mình, chúng tôi sẽ đặt đúng thuộc tính phiên người dùng này để điều đó sẽ giúp chúng tôi điều hướng đến màn hình hồ sơ nhưng chúng tôi thực sự cần tìm nạp dữ liệu đó chúng tôi vừa tải lên Firebase để nó có thể hiển thị chính xác trên màn hình và điều đó phải xảy ra khi chúng tôi tạo người dùng của mình, vì vậy chúng tôi sẽ tạo một tài khoản khác trong giây lát. Tôi muốn chúng tôi xem qua ngay bây giờ để chúng tôi có thể thấy tài khoản đó đang hoạt động nhưng những gì tôi muốn chúng tôi làm bây giờ là tìm hiểu cách thực sự đưa dữ liệu người dùng của chúng tôi vào màn hình hồ sơ này, vì vậy hãy quay lại chế độ xem hồ sơ của chúng tôi

**Main subtitle**

and here you guys can see we're just using this mock user right here so what we need to do is add another instance of our environment object and what we're going to do is wrap this entire thing in an if let statement so let me just cut this with command X and say if let user equals viewmodel dot current user and then paste it all in there and here we just replace user.mock user with user so that's going to be our user initials user.full name user.email and that looks good and really quickly guys I think I want to change the full name here from test let's change it to Michael Jordan you guys might have done something more descriptive when we signed up the user but you know I just want to make sure the initials show up correctly and all that stuff and it's not just tests so I just really quickly directly modified the data in Firebase you want to try to avoid that if possible but sometimes if you just need to test some things like we're doing here it's okay but now let's go ahead and run this

**Second subtitle**

và ở đây các bạn có thể thấy chúng tôi chỉ đang sử dụng người dùng giả này ngay tại đây, vì vậy những gì chúng tôi cần làm việc cần làm là thêm một thể hiện khác của đối tượng môi trường của chúng ta và những gì chúng ta sẽ làm là bao bọc toàn bộ điều này trong một câu lệnh if let, vậy hãy để tôi cắt phần này bằng lệnh X và nói if let user equals viewmodel dot user hiện tại và sau đó dán nó tất cả trong đó và ở đây chúng tôi chỉ thay thế người dùng user.mock bằng người dùng để người dùng của chúng tôi viết tắt tên người dùng.tên đầy đủ user.email và điều đó có vẻ tốt và thực sự nhanh chóng. Tôi nghĩ tôi muốn thay đổi tên đầy đủ ở đây từ thử nghiệm. thay đổi nó thành Michael Jordan, các bạn có thể đã làm điều gì đó mô tả hơn khi chúng tôi đăng ký người dùng nhưng bạn biết đấy, tôi chỉ muốn đảm bảo rằng những chữ cái đầu tiên hiển thị chính xác và tất cả những thứ đó và đó không chỉ là các bài kiểm tra nên tôi đã thực sự nhanh chóng sửa đổi trực tiếp dữ liệu trong Firebase, bạn muốn cố gắng tránh điều đó nếu có thể nhưng đôi khi nếu bạn chỉ cần kiểm tra một số thứ như chúng tôi đang làm ở đây thì không sao nhưng bây giờ hãy tiếp tục và chạy lại phần này

**Main subtitle**

again

and we should notice that once we get

this current user guy it'll populate our entire profile screen with the user's

data that's coming back from Firebase and you guys notice that that's exactly what's happening there right the user's name in Firebase is Michael Jordan that's the test the email and then these are his initials so that's looking really really good next up what I want us to do is go over how to sign a user

out

so we're going to go back to our auth

view model now and implement this sign out function and this should trigger two

events right it should make it so that it takes us back to our login screen within the app so that's like what should happen client side and on the back end it should also execute that sign out functionality so it actually signs us out with the Firebase servers as well so there's two things we want to happen there one thing on client side which is the actual navigation we wanted

to take us back to our login screen and then we also wanted to sign us out on

the back end so this is actually really simple guys we're just going to say do try

auth.auth DOT sign out

**Second subtitle**

và chúng tôi sẽ lưu ý rằng khi chúng tôi có được

người dùng hiện tại anh bạn, nó sẽ điền vào toàn bộ màn hình hồ sơ của chúng ta với dữ liệu của người dùng

đến từ Firebase

và các bạn lưu ý rằng đó chính xác là những gì đang xảy ra ở đó, tên người dùng trong Firebase là Michael Jordan, đó là email kiểm tra và sau đó đây là tên viết tắt của anh ấy, vậy trông thực sự rất tốt, tiếp theo, điều tôi muốn chúng ta làm là tìm hiểu cách đăng xuất người dùng,

vì vậy chúng ta sẽ quay lại mô hình xem xác thực của mình

ngay bây giờ và triển khai

chức năng đăng xuất này và điều này sẽ kích hoạt hai

sự kiện đúng như vậy làm cho nó

đưa chúng ta trở lại màn hình đăng nhập trong ứng dụng để nó giống như những gì sẽ xảy ra ở phía máy khách và ở mặt sau, nó cũng sẽ thực thi

chức năng đăng xuất đó để nó cũng thực sự đăng xuất chúng ta bằng máy chủ Firebase, vì vậy, có hai điều chúng tôi muốn xảy ra ở đó, một điều ở phía khách hàng, đó là điều hướng thực tế mà chúng tôi muốn

đưa chúng tôi trở lại màn hình đăng nhập và sau đó chúng tôi cũng muốn đăng xuất chúng tôi ở

mặt sau, vì vậy điều này thực sự rất đơn giản, chúng tôi chỉ cần nói hãy thử

đăng xuất auth.auth DOT



**Main subtitle**

and we're going to say self.user session equals nil self.current user equals nil and here's our catch block failed to sign out with error dot localized description I've never had an error happen on sign out but you know it is recommended to put it there but let me just talk briefly about what these three lines of code are doing so this line here signs out user on back end this uh wipes out user session and takes us back to login screen so if you guys remember our presentation Logic for our content view is all based on this user session property so if we wipe it out or set it to nil then it's going to take us back to our login screen so basically it's going to say hey something updated on The auth View model my user session is now Nils therefore I will show the login View and then we also want to wipe out our current user object which is our data model right because we don't want to hold on to that in any way because then you might see a potential error happen where you log out as one user log back

**Second subtitle**

và chúng tôi sẽ nói rằng phiên self.user bằng 0. self. user hiện tại bằng 0 và đây là khối catch của chúng tôi không đăng xuất được với mô tả được bản địa hóa về dấu chấm lỗi mà tôi chưa từng có đã xảy ra lỗi khi đăng xuất nhưng bạn biết rằng nên đặt nó ở đó nhưng hãy để tôi nói ngắn gọn về những gì ba dòng mã này đang làm để dòng này ở đây đăng xuất người dùng ở mặt sau, uh, phiên người dùng này sẽ bị xóa và đưa chúng ta quay lại màn hình đăng nhập, vì vậy nếu các bạn nhớ Logic trình bày của chúng tôi cho chế độ xem nội dung của chúng tôi đều dựa trên thuộc tính phiên người dùng này, vì vậy nếu chúng tôi xóa nó hoặc đặt nó thành không thì nó sẽ đưa chúng tôi trở lại màn hình đăng nhập nên về cơ bản, nó sẽ hoạt động nói này, một cái gì đó được cập nhật trên Mô hình Chế độ xem xác thực, phiên người dùng của tôi hiện là Nils, do đó tôi sẽ hiển thị Chế độ xem đăng nhập và sau đó chúng tôi cũng muốn xóa đối tượng người dùng hiện tại, đó là mô hình dữ liệu của chúng tôi vì chúng tôi không muốn giữ theo bất kỳ cách nào bởi vì sau đó bạn có thể thấy một lỗi tiềm ẩn xảy ra khi bạn đăng xuất khi một người dùng đăng nhập lại với

**Main subtitle**

in as another and you might see the old data still presented there which you don't want so you want to wipe this guy

out as well

so let me say wipes out

current user data model

so that's what these three lines of code are doing now we just need to go ahead and implement this

um as in our profile view so we'll go to

profile view

we have print sign out

so here we're just going to say

viewmodel DOT sign out

and if we run our app

we should notice that when we click on

that button it signs us out

so we're still logged in that's great it

fetches our user information that's

great we hit sign out perfect it logs us

out and takes us back to our login

screen now to test and see if we are

really logged out then what we're going

to want to do is launch the app again

and verify that it takes us to our login

screen and that's exactly what it's

doing so this is working beautifully

guys next up we're going to go over how

to log our user in

**Second subtitle**

tư cách người khác và bạn có thể thấy dữ liệu cũ vẫn được hiển thị ở đó mà bạn không muốn, vì vậy bạn muốn xóa sạch anh chàng này

đồng thời, hãy để tôi nói xóa sạch

mô hình dữ liệu người dùng hiện tại,

vì vậy đó là những gì ba dòng mã này

đang làm bây giờ, chúng tôi chỉ cần tiếp tục

và triển khai điều này

ừm như trong chế độ xem hồ sơ của chúng tôi, vì vậy chúng tôi sẽ chuyển sang chế

độ xem hồ sơ,

chúng tôi có dấu hiệu in out

nên ở đây chúng ta sẽ nói

viewmodel DOT đăng xuất

và nếu chúng ta chạy ứng dụng của mình,

chúng ta sẽ nhận thấy rằng khi chúng ta nhấp vào

nút đó, nó sẽ đăng xuất chúng ta

vì vậy chúng ta vẫn đăng nhập, điều đó thật tuyệt, nó

đã tải thông tin người dùng của chúng ta, điều đó thật

tuyệt chúng tôi nhấn đăng xuất hoàn hảo, nó sẽ đăng xuất chúng tôi

và đưa chúng tôi trở lại

màn hình đăng nhập ngay bây giờ để kiểm tra và xem liệu chúng tôi có

thực sự đăng xuất hay không, điều chúng tôi sẽ

muốn làm là khởi chạy lại ứng dụng

và xác minh rằng nó sẽ đưa chúng tôi đến màn hình đăng nhập của chúng tôi

và đó chính xác là những gì nó đang

làm, vì vậy màn hình này đang hoạt động rất tốt,

tiếp theo chúng ta sẽ tìm hiểu cách

đăng nhập người dùng của mình,

**Main subtitle**

so we are going to head back to our auth

view model

and we are going to fill in this sign in  
function so this one's pretty simple

we're gonna once again wrap this up in a

do catch block

let me fix that really quick so we're

going to say do

catch and we are going to say let result  
equal try await

auth.auth DOT sign in with email oops

email and password so we're going to say  
pass in that email password and it's  
important to note guys that we are  
getting those from our input parameters

so we pass these parameters in when we

call this function back in the login  
view because that's where that  
information is coming from the user has  
to fill in this form we pass those

values in when we call this function

so that looks good then we're going to  
say self.user session equals result dot  
user and then we're going to say await

fetch user so I'm actually going to  
comment this out really quickly  
uh just so you guys can see what happens  
if we don't call that function and then

here guys we're going to say print debug

**Second subtitle**

vì vậy chúng ta sẽ quay lại mô hình xem xác  
thực của mình

và chúng ta sẽ điền thông tin này

chức năng đăng nhập vì vậy chức năng này  
khá đơn giản,

một lần nữa chúng tôi sẽ kết thúc việc này  
trong

khối do catch. Hãy

để tôi khắc phục điều đó thật nhanh vì vậy  
chúng tôi

sẽ nói do

catch và chúng tôi sẽ nói let result

equal try await

auth. auth DOT đăng nhập bằng email Rất  
tiếc

email và mật khẩu vì vậy chúng tôi sẽ nói  
vượt qua mật khẩu email đó và điều

quan trọng cần lưu ý là chúng tôi đang  
lấy những thông số đó từ các tham số đầu  
vào của mình

vì vậy chúng tôi chuyển các tham số này  
vào khi chúng tôi

gọi lại chức năng này chế

độ xem đăng nhập vì đó là nơi

thông tin đến từ người dùng

phải điền vào biểu mẫu này, chúng tôi  
chuyển

các giá trị đó vào khi chúng tôi gọi chức  
năng này

để có vẻ tốt, sau đó chúng tôi sẽ

nói phiên self.user bằng kết quả

người dùng dấu chấm và sau đó chúng tôi  
sẽ nói đang chờ

tìm nạp người dùng vì vậy tôi thực sự sẽ

bình luận điều này thật nhanh

uh để các bạn có thể thấy điều gì sẽ xảy ra  
nếu chúng tôi không gọi chức năng đó và  
sau đó

ở đây chúng tôi sẽ nói là in gỡ lỗi

**Main subtitle**

failed to log in with error  
 dot localize description cool  
 so we've already called this function  
 back inside of our login view so let's  
 go ahead and run this and see if we can  
 log into our app  
 so because we're setting the user  
 session property it is going to take us  
 to our profile  
 but  
 since we're not fetching the user I'm  
 pretty sure that profile is going to be  
 blank so  
 that's what's going to happen here so  
 let's log in  
 with the account that we created test  
 gmail.com  
 hit sign in  
 and yep so it logs Us in right the  
 screen changes but because there's no  
 user  
 right this whole profile view is wrapped  
 inside of this if let statement right so  
 the user has to be there in order for  
 this view to render so in order for that  
 to  
 happen we do have to call this fetch  
 user function so that it goes and gets  
 that logged in user from Firebase and  
 guys it's very important that you wait  
 to call that function until after you

**Second subtitle**

không thể đăng nhập với lỗi  
 mô tả bản địa hóa dấu chấm rất tuyệt  
 vì vậy chúng tôi đã gọi lại chức năng này  
 bên trong chế độ xem đăng nhập của mình,  
 vì vậy hãy  
 tiếp tục và chạy chức năng này và xem liệu  
 chúng tôi có thể  
 đăng nhập vào ứng dụng của mình  
 không vì chúng tôi đang đặt thuộc  
 tính phiên người dùng nó sẽ đưa chúng tôi  
 đến hồ sơ của chúng tôi  
 nhưng  
 vì chúng tôi không tìm nạp người dùng nên  
 tôi  
 khá chắc chắn rằng hồ sơ đó sẽ  
 trống nên  
 đó là điều sẽ xảy ra ở đây, vì vậy  
 hãy đăng nhập  
 bằng tài khoản mà chúng tôi đã tạo gmail  
 thử nghiệm  
 . com  
 nhấn đăng nhập  
 và vâng để nó đăng nhập Chúng tôi ở bên  
 phải  
 màn hình thay đổi nhưng vì không có  
 quyền người dùng nên toàn bộ chế độ xem  
 hồ sơ này được bao bọc  
 bên trong câu lệnh if let này vì vậy  
 người dùng phải ở đó để chế  
 độ xem này hiển thị như vậy trong để điều  
 đó  
 xảy ra, chúng tôi phải gọi  
 chức năng tìm nạp người dùng này để nó đi  
 và nhận  
 người dùng đã đăng nhập đó từ Firebase và  
 các bạn, điều rất quan trọng là bạn phải đợi  
 để gọi chức năng đó cho đến sau khi bạn

**Main subtitle**

call the sign in function because if you  
 you have to sign in first as the you  
 know the user that you're trying to log  
 in with so that  
 this uid guy exists and we can fetch the  
 correct user information and display it  
 in our profile view so that's why we  
 have to have that there in the login  
 function but if we were to run the app  
 again you'll notice that on the on the  
 initialization of our off view model  
 it's going to call fetch user and we're  
 going to be good to go  
 so yep Michael Jordan so what I want us  
 to do now guys is let's just go ahead  
 and create another account so let's say  
 test2 gmail.com for make one for Kobe  
 Bryant  
 and hit sign up and we should notice  
 that once we sign up it goes and takes  
 us to the profile infectious Kobe's  
 information from the back end so that's  
 looking absolutely amazing that's the  
 bulk of the functionality here guys what  
 we're going to do to finish this video  
 up is ADD our form validation stuff  
 that's just going to make it so you know  
 the button is only enabled under certain  
 conditions and stuff and that's going to  
 look really good so what I mean by that

**Second subtitle**

gọi chức năng đăng nhập bởi vì nếu  
 bạn phải đăng nhập trước vì bạn  
 biết người dùng mà bạn đang cố đăng  
 nhập để  
 anh chàng uid này tồn tại và chúng tôi có  
 thể tìm nạp  
 thông tin người dùng chính xác và hiển thị  
 thông tin đó  
 trong chế độ xem hồ sơ của chúng tôi, vì  
 vậy đó là lý do tại sao chúng tôi  
 phải có thông tin đó ở đó trong  
 chức năng đăng nhập nhưng nếu chúng tôi  
 chạy lại ứng dụng,  
 bạn sẽ nhận thấy rằng khi  
 khởi tạo mô hình tắt chế độ xem của chúng  
 tôi,  
 nó sẽ gọi người dùng tìm nạp và chúng tôi  
 sẽ ổn thôi,  
 vâng Michael Jordan vậy thì sao Tôi muốn  
 chúng ta  
 làm ngay bây giờ, mọi người hãy tiếp tục  
 và tạo một tài khoản khác, vì vậy hãy nói  
 test2 gmail.com để tạo một tài khoản cho  
 Kobe  
 Bryant  
 và nhấn đăng ký và chúng ta nên lưu ý  
 rằng sau khi chúng ta đăng ký, nó sẽ đưa  
 chúng ta đến hồ sơ thông tin lây nhiễm của  
 Kobe  
 từ phía sau, vì vậy điều đó  
 trông cực kỳ tuyệt vời, đó là  
 phần lớn chức năng ở đây, những gì  
 chúng ta sẽ làm để hoàn thành video này  
 là THÊM công cụ xác thực biểu mẫu của  
 chúng tôi  
 sẽ tạo ra nó để bạn biết  
 nút là chỉ được kích hoạt trong một số  
 điều kiện và nội dung nhất định và điều đó  
 sẽ  
 trông thực sự tốt, vì vậy ý tôi là ví

**Main subtitle**

for example

is like if there's no password filled in here

the sign in button is not enabled right and it actually has to be greater than six characters and all of that stuff you guys can even implement stuff where it

has to have a capital letter and blah blah blah but we're not going to worry about that now I'm just going to show you the basic functionality of it you guys can add to it as you see fit so

that's what we're going to do next so what we're going to do guys is in our auth viewmodel file we're going to

create a protocol and we're going to call it authentication form protocol and we're going to have one property in here it's going to be called form is valid it's going to be a Boolean and it's going to require us to get that value so what we're going to do is

everywhere we have a form we are going to conform to or implement this authentication form protocol and determine the logic that will tell us whether or not the form is valid so let's just go ahead and take a look at what that's going to look like so in your login view scroll down to the bottom and let's create an extension here so we're going to say login View

**Second subtitle**

dụ như nếu không có mật khẩu điền vào đây thì

nút đăng nhập không được bật đúng và nó thực sự phải lớn hơn sáu ký tự và tất cả những thứ đó các bạn thậm chí có thể triển khai những thứ mà nó

phải có chữ in hoa và blah blah blah nhưng chúng ta sẽ không lo lắng về điều đó bây giờ tôi sẽ chỉ cho các bạn chức năng cơ bản của nó mà các bạn có thể thêm vào với nó khi bạn thấy phù hợp, vì vậy

đó là những gì chúng ta sẽ làm tiếp theo vì vậy những gì chúng ta sẽ làm là trong tệp auth viewmodel của chúng ta, chúng ta sẽ

tạo một giao thức và chúng ta sẽ gọi nó là giao thức biểu mẫu xác thực và chúng tôi sẽ có một thuộc tính ở đây, nó sẽ được gọi là biểu mẫu hợp lệ, nó sẽ là Boolean và nó sẽ yêu cầu chúng tôi lấy giá trị đó, vì vậy những gì chúng tôi sẽ làm là ở

mọi nơi chúng tôi có biểu mẫu. sẽ tuân thủ hoặc triển khai giao thức biểu mẫu xác thực này và xác định logic sẽ cho chúng tôi biết biểu mẫu có hợp lệ hay không, vì vậy, hãy tiếp tục và xem

giao diện sẽ như thế nào trong chế độ xem đăng nhập của bạn, hãy cuộn xuống dưới cùng và hãy tạo một tiện ích mở rộng ở đây để chúng tôi sẽ nói Chế độ xem đăng nhập

**Main subtitle**

and we're going to conform to this authentication form protocol and because we have that property there form is valid it's going to require that we implemented right so if you hit fix it's going to say Hey you because we have to follow the rules of this protocol so we have to have this form as valid property here so we are going to want to you know add some simple logic to detect whether or not our form is valid so let me just go ahead and show you guys what it is so we can talk about it once we're done we are going to say return exclamation point email dot is empty and email dot contains an at symbol and password is not empty and password dot count is greater than five so Firebase has some password rules the password has to be at least six characters you guys could make this password logic whatever you want but essentially we now have this form is valid property that's going to be determined based on all of these conditions right we have to have an email in there we have to have a password in there the email has to contain an at symbol and the password

**Second subtitle**

và chúng tôi sẽ tuân thủ giao thức biểu mẫu xác thực này và vì chúng tôi có thuộc tính đó nên biểu mẫu hợp lệ nên chúng tôi sẽ yêu cầu triển khai đúng như vậy nếu bạn nhấn sửa nó sẽ nói Này bạn vì chúng tôi phải tuân theo các quy tắc của giao thức này nên chúng tôi phải có biểu mẫu này làm thuộc tính hợp lệ ở đây vì vậy chúng tôi sẽ muốn bạn biết thêm một số logic đơn giản để phát hiện xem biểu mẫu của chúng tôi có phải là hợp lệ, vì vậy hãy để tôi tiếp tục và cho các bạn thấy nó là gì để chúng ta có thể nói về nó sau khi hoàn thành, chúng ta sẽ nói dấu chấm than trả về dấu chấm email trống và dấu chấm email chứa ký hiệu at và mật khẩu không trống và số dấu chấm mật khẩu lớn hơn năm, vì vậy Firebase có một số quy tắc mật khẩu, mật khẩu phải có ít nhất sáu ký tự, các bạn có thể đặt logic mật khẩu này theo bất kỳ cách nào bạn muốn nhưng về cơ bản, chúng tôi hiện có biểu mẫu này là thuộc tính hợp lệ sẽ được xác định dựa trên tất cả trong số các điều kiện này, chúng tôi phải có một email trong đó, chúng tôi phải có mật khẩu trong đó, email phải chứa ký hiệu at và

**Main subtitle**

count has to be greater than five so now

the question becomes okay how are we going to use this form as valid property

so basically it's all just going to be implemented on this sign in button here guys so

right here we are going to say on the sign in button this guy right here and

once again that's where it's nice to have these comments it makes it easier to find things we're going to say dot disabled

form is valid and we want to negate that

and then we are going to give the button an opacity

based on whether or not the form is valid so this is what it's going to look

like we're going to say dot opacity and if the form is valid yes it's going to be 1.0 no it's going to be 0.5 so that's going to give it that faded look if the opacity is I mean sorry if the form isn't valid so let's

go ahead and just run this really quickly guys to take a look at what that looks like and then we are going to implement that

in our registration view as well so you guys can see that the button is faded out

and let's try a couple different test

**Second subtitle**

số lượng mật khẩu phải lớn hơn năm, vì vậy bây giờ

câu hỏi trở nên ổn, chúng tôi

sẽ sử dụng như thế nào biểu mẫu này là thuộc tính hợp lệ,

vì vậy về cơ bản, tất cả sẽ được

triển khai trên nút đăng nhập này, vì vậy

ngay tại đây, chúng tôi sẽ nói về

nút đăng nhập, anh chàng này ngay tại đây và

một lần nữa, thật tuyệt khi

có những nhận xét này giúp dễ dàng

tim thấy những thứ mà chúng tôi sẽ nói rằng

biểu mẫu vô hiệu hóa dấu chấm là hợp lệ và chúng tôi muốn phủ nhận điều đó

và sau đó chúng tôi sẽ cung cấp cho nút một độ mờ

dựa trên việc biểu mẫu có

hợp lệ hay không, vì vậy đây là những gì nó sẽ làm có vẻ

như chúng ta sẽ nói độ mờ của dấu chấm

và nếu biểu mẫu hợp lệ thì

có, nó sẽ là 1,0, không, nó sẽ

là 0,5, vì vậy điều đó sẽ làm cho nó

trông mờ nhạt nếu độ mờ là ý tôi,

xin lỗi nếu biểu mẫu không phải là hợp lệ, vì vậy hãy

tiếp tục và chạy thật

nhanh để xem nó

trông như thế nào

và sau đó chúng tôi cũng sẽ triển khai điều đó

trong chế độ xem đăng ký của mình để các bạn có thể thấy rằng nút bị mờ

đi

và hãy thử một một vài trường hợp thử nghiệm khác nhau



**Main subtitle**

cases here so let's see if I just put  
test  
in there and test  
and like make my password a bunch of  
characters because the email field  
doesn't have this at symbol it's not  
going to work but you notice as soon as  
I put that at symbol in there it's going  
to be good to go  
um you know then you could say like  
gmail.com or something and you guys  
could implement some further validation  
if you want  
um and then that's good to go and  
obviously if we try to sign in with that  
account it's we're going to get an error  
message  
um  
so let me just do this and you see if I  
delete the password right it fades the  
button out  
and it doesn't fade it back in or make  
it come to life with that color until  
the password goes over six characters so  
the form validation is working nicely  
there  
um let me just go ahead and add  
something called a Mark here  
in the code  
so that we know exactly what this is  
and we can actually copy and paste this  
guys and go to our registration View and  
do the same thing

**Second subtitle**

ở đây, vì vậy hãy xem liệu tôi có đặt  
thử nghiệm  
vào đó và thử nghiệm  
và thích đặt mật khẩu của mình thành một  
loạt  
ký tự vì trường email  
không có ký hiệu này không, nó sẽ không  
hoạt động nhưng bạn sẽ chú ý ngay khi  
tôi đặt rằng tại biểu tượng trong đó sẽ  
rất tốt nếu bạn truy cập  
ừm, bạn biết đấy, sau đó bạn có thể nói như  
gmail.com hoặc một cái gì đó và các bạn  
có thể thực hiện một số xác nhận bổ sung  
nếu bạn muốn  
và sau đó thật tuyệt khi tiếp tục và  
rõ ràng là nếu chúng tôi cố gắng ký với  
tài khoản đó, chúng tôi sẽ nhận được một  
thông báo lỗi,  
ừm,  
vậy hãy để tôi làm điều này và bạn xem nếu  
tôi  
xóa đúng mật khẩu,  
nút sẽ mờ dần  
và không mờ trở lại hoặc  
hoạt động trở lại với màu đó cho đến khi  
mật khẩu vượt quá sáu ký tự để  
xác thực biểu mẫu hoạt động tốt  
ở đó,  
hãy để tôi tiếp tục và thêm  
một thứ gọi là Đánh dấu vào đây  
trong mã  
để chúng tôi biết chính xác đây là gì  
và chúng tôi thực sự có thể sao chép và dán  
mã này  
các bạn và đi tới phần đăng ký của chúng  
tôi Xem và  
làm điều tương tự

**Main subtitle**

so email blah blah password yep and we

want to also make sure

full name

oops this needs to be extension

registration view sorry

so now full name Doc is empty

and we want to negate that

and lastly we are going to want to make

sure that the password is equal to

confirm password so let's go here with

for our password validation and say and

confirm password equals password

right so this is going to make sure the

passwords match

before we validate this form and then we

can literally just go back here

copy and paste these two lines of code

and put it back on this button

let's run our code and check this out

and then we are going to do some UI

implementation as well to show like a

check mark to make sure or an x mark to

make sure those passwords match and

that'll be the final feature of this

YouTube tutorial so let's see we'll have

test gmail.com

full name is test

ah man I hate it when it does this

sometimes it like Auto fills that with a

strong password and it's really annoying

on the simulator so let's just see tests

you guys see it's still not enabled

**Second subtitle**

để gửi email blah blah mật khẩu yep và  
chúng tôi

cũng muốn đảm bảo họ và

tên Rất tiếc,

đây cần phải là tiện ích mở rộng

. cuối cùng, chúng tôi sẽ muốn

đảm bảo rằng mật khẩu bằng với

mật khẩu xác nhận, vì vậy hãy đến đây để

xác thực mật khẩu của chúng tôi và nói và

xác nhận mật khẩu bằng mật khẩu,

vì vậy điều này sẽ đảm bảo

mật khẩu khớp

trước khi chúng tôi xác thực biểu mẫu này  
và sau đó chúng ta

có thể thực sự chỉ cần quay lại đây

sao chép và dán hai dòng mã này

và đặt lại vào nút này.

Hãy chạy mã của chúng ta và kiểm tra điều  
này

và sau đó chúng ta sẽ thực hiện một số triển  
khai giao diện người dùng

để hiển thị như một

dấu kiểm cho đảm bảo hoặc dấu x để

đảm bảo rằng các mật khẩu đó khớp và

đó sẽ là tính năng cuối cùng của

hướng dẫn YouTube này, vì vậy hãy xem  
chúng tôi sẽ có

tên đầy đủ là gmail.com thử nghiệm

ah, tôi ghét nó khi

đôi khi nó làm điều này giống như Tự động  
điền

mật khẩu mạnh và nó thực sự khó chịu

trên trình giả lập, vì vậy hãy xem các bài  
kiểm tra mà

các bạn thấy nó vẫn chưa được bật

**Main subtitle**

still not enabled

but as soon as those two things equal each other then the button becomes enabled so that is looking really really good guys so that's our form validation

uh next up I do want to give you guys a nice little bonus feature where we are going to show some sort of UI indicator that will let the user know that the passwords either match or don't match so let's go ahead and get started with that but really quickly before we do that

guys I do want to talk briefly about this protocol so back in the auth view

model we created this protocol and we use this so that both of these two things can implement that protocol and notice that we require some sort of logic to tell us whether or not the form is valid so obviously that's not required but it is considered like good programming and best practices to do

that because now anytime you have

another form you will see that all of the existing forms like this login view

and registration view conform to this protocol so that when if you ever create another one or if another person works

on your code whatever it may be they know that a form will have to conform to

**Second subtitle**

vẫn chưa được bật

nhưng ngay sau khi hai thứ đó bằng nhau thì nút sẽ được bật nên trông thực sự các bạn thực sự tốt, vậy đó là phần xác thực biểu mẫu của chúng tôi, uh tiếp theo tôi muốn cung cấp cho các bạn một tính năng phần thưởng nhỏ thú vị, nơi chúng tôi sẽ hiển thị một số loại chỉ báo giao diện người dùng sẽ cho người dùng biết rằng mật khẩu khớp hoặc không khớp. Vì vậy, hãy tiếp tục và bắt đầu với điều đó nhưng thực sự nhanh chóng trước khi chúng ta thực hiện điều đó.

Tôi thực sự muốn nói ngắn gọn về giao thức này, vì vậy, trong mô hình chế độ xem xác thực,

chúng tôi đã tạo giao thức này và chúng tôi sử dụng giao thức này để cả hai điều này có thể thực hiện giao thức đó và lưu ý rằng chúng tôi yêu cầu một số loại logic để cho chúng tôi biết biểu mẫu có hợp lệ hay không nên rõ ràng là không bắt buộc nhưng nó được coi là lập trình tốt và các phương pháp hay nhất để làm

điều đó bởi vì bây giờ bất cứ khi nào bạn có một

biểu mẫu khác, bạn sẽ thấy điều đó tất cả các biểu mẫu hiện có như Chế độ xem đăng nhập

và Chế độ xem đăng ký này đều tuân theo giao thức này để khi bạn tạo một biểu mẫu khác hoặc nếu một người khác làm việc

trên mã của bạn, họ có thể biết rằng một biểu mẫu sẽ phải tuân theo

**Main subtitle**

this protocol and implement this form as

valid property and use the logic like

this to enable or disable your button so

that's why we use a protocol there and

Implement that across our two different

class

so now let's go back to our registration

View and implement this fund confirm

password feature so we are going to uh

take this input view here for the

confirm password field and wrap it into

Z stack so cut that out really quickly

we're going to say the alignment is dot

trailing

and we're going to put that guy right

there

just like before

and we are going to add a little check

mark or x mark at the end of the field

right here based on whether or not the

passwords are equal to each other so we

want to say if

password dot is empty

and

confirm password

dot is empty so we want to make sure

both of them have what's it called text

before we start showing it and then

we're also going to check if password

equals confirm password

**Second subtitle**

giao thức này và triển khai biểu mẫu này  
làm thuộc

tính hợp lệ và sử dụng logic như

thế này để bật hoặc tắt nút của bạn, đó

là lý do tại sao chúng tôi sử dụng một giao  
thức ở đó và

Triển khai giao thức đó trên hai lớp khác  
nhau của chúng tôi,

vì vậy bây giờ hãy quay lại phần đăng ký  
của chúng tôi

Xem và triển khai

tính năng mật khẩu xác nhận quỹ này Vì  
vậy, chúng tôi sẽ

lấy chế độ xem đầu vào này ở đây cho

trường mật khẩu xác nhận và đưa nó vào

ngăn xếp Z để cắt nó ra thật nhanh, chúng  
tôi sẽ

nói rằng căn chỉnh là dấu chấm

và chúng tôi sẽ đặt anh chàng đó ngay

tại đó

giống như trước đây

và chúng tôi sẽ thêm một dấu kiểm nhỏ

hoặc dấu x ở cuối trường

ngay tại đây dựa trên việc

các mật khẩu có bằng nhau hay không, vì  
vậy chúng tôi

muốn nói nếu

dấu chấm mật khẩu trống

và

xác nhận

dấu chấm mật khẩu trống nên chúng tôi  
muốn đảm bảo

cả hai đều có cái được gọi là văn bản

trước khi chúng tôi bắt đầu hiển thị văn bản  
và sau đó

chúng tôi cũng sẽ kiểm tra xem mật khẩu

có bằng xác nhận mật khẩu

**Main subtitle**

and say else there

so here

we are going to say image system name

checkmark dot Circle

dot fill

dot image scale is dot large dot font

weight is Dot bolds dot foreground color

is color

dot system green oops

okay and then we can just copy and paste

this and in this elk sorry else block

we're going to say

xmark.circle.fill and make the color

system red

okay so basically guys the logic here is

saying that hey make sure both of these

two Fields have text that's what this is

doing because as you're filling out the

form you don't want that to show up

right you only want it to show up once

the user has typed stuff into both of

these fields

and then it's going to say if the two

things are equal show a check mark right

here if they're not show an x mark and

the tint will be red so let's go ahead

and run this now

and see if that works and that's just a

nice little bonus feature for you guys

there to really improve the user

**Second subtitle**

và nói khác ở đó

vì vậy ở đây

chúng tôi sẽ nói dấu kiểm tên hệ thống hình ảnh

dấu chấm Hình tròn dấu chấm

lấp đầy hình ảnh dấu chấm Tỷ

lệ hình ảnh là dấu chấm lớn.

Trọng lượng phông chữ là Dấu chấm đậm.  
Màu nền trước

là màu của

hệ thống dấu chấm màu xanh lá cây.

Rất tiếc, sau đó chúng ta chỉ cần sao chép và dán

cái này và trong khối nai sừng tấm này, xin lỗi,

chúng ta sẽ nói

xmark. circle.fill và làm cho

hệ thống màu có màu đỏ

là được, vì vậy về cơ bản, logic ở đây

nói rằng hãy đảm bảo rằng cả

hai Trường này đều có văn bản, đó là những gì nó đang

làm bởi vì khi bạn điền vào

biểu mẫu, bạn không muốn điều đó hiển thị

đúng, bạn chỉ muốn nó hiển thị khi

người dùng đã nhập nội dung vào cả hai

trường này

và sau đó sẽ cho biết liệu hai

nội dung có bằng nhau hay không, hãy hiển thị dấu kiểm ngay

tại đây nếu chúng không hiển thị dấu x và

dấu kiểm tông màu sẽ có màu đỏ, vì vậy hãy tiếp tục

và chạy ứng dụng này ngay bây giờ

và xem tính năng đó có hoạt động không và đó chỉ là một

tính năng bổ sung nhỏ thú vị dành cho các bạn

ở đó để thực sự cải thiện

**Main subtitle**

interface and user experience of this

app

so we're going to go sign up

test gmail.com

test

so you guys will notice that if I start

typing into this password field nothing

shows up here right because we're

requiring that both of the fields have

text so that's by Design

and then you see here that now the x

button starts showing up right because

they're not equal

and as soon as they are equal that green

check mark shows up to say hey your

passwords match if we change it it goes

back to X and it invalidates our form

and if we they match then this is green

and everything is good to go and that is

pretty much going to wrap it up for the

YouTube version of this app guys we have

a full authentication set up here where

we can log users in sign them up fetch

their information from Firebase and all

that awesome stuff this can be dragged

and dropped into literally any single

project you have that requires user

authentication or user profiles the

**Second subtitle**

giao diện người dùng và trải nghiệm người dùng của

ứng dụng này,

vì vậy chúng ta sẽ đăng ký

kiểm tra gmail.com

kiểm tra

để các bạn sẽ nhận thấy rằng nếu tôi bắt đầu

nhập vào trường mật khẩu này thì không có gì

hiển thị ở đây vì chúng tôi

yêu cầu cả hai trường phải có

văn bản, do đó, đó là do Thiết kế

và sau đó bạn thấy ở đây bây giờ là

nút x bắt đầu hiển thị đúng vì

chúng không bằng nhau

và ngay khi chúng bằng nhau, dấu kiểm màu xanh lá cây đó sẽ

hiển thị để nói rằng: này,

mật khẩu của bạn khớp nếu chúng tôi thay đổi, nó sẽ quay

lại X và nó sẽ làm mất hiệu lực biểu mẫu của chúng tôi

và nếu chúng khớp nhau thì điều này có màu xanh lá cây

và mọi thứ đều hoạt động tốt và điều đó

gần như sẽ hoàn tất cho

phiên bản YouTube của ứng dụng này, chúng tôi có

một xác thực đầy đủ được thiết lập tại đây, nơi

chúng tôi có thể đăng nhập người dùng đăng ký, lấy

thông tin của họ từ Firebase và tất cả

thứ tuyệt vời đó có thể được kéo

và thả vào bất kỳ

dự án đơn lẻ nào mà bạn có yêu cầu

xác thực người dùng hoặc hồ sơ người dùng,

**Main subtitle**

completed version of these this app is going to have a couple extra features that are nice to have so for example it's going to have a custom error handling it'll also have this forgot password feature so that users can reset their password via email and in the future I'm going to be adding social sign-ins like Google and Facebook and all of that stuff and we also see here that we have like a custom loading indicator there as well so once again guys if you want that source code it is available on my website at stefanocode.com sign up for any one of these membership plans to get access to the YouTube courses so that's everything you see here I'll be adding this one right now and signing up like I said gets you access to all of this stuff and make sure you check out all of the other courses I have in the description to this video so thanks so much for watching guys we'll see you in the next one peace out

**Second subtitle**

phiên bản hoàn chỉnh của những ứng dụng này sẽ có một vài tính năng bổ sung rất hay, chẳng hạn như nó sẽ có xử lý lỗi tùy chỉnh, nó cũng sẽ có tính năng quên mật khẩu này để người dùng có thể đặt lại mật khẩu của họ qua email và trong tương lai tôi sẽ thêm các đăng nhập xã hội như Google và Facebook và tất cả những thứ đó và chúng tôi cũng thấy ở đây, chúng tôi cũng có một chỉ báo tải tùy chỉnh ở đó, vì vậy một lần nữa, nếu bạn muốn mã nguồn đó, nó có sẵn trên trang web của tôi tại stefanocode.com, hãy đăng ký bất kỳ gói thành viên nào trong số này để có quyền truy cập vào các khóa học YouTube, vì vậy đó là mọi thứ bạn thấy ở đây, tôi sẽ thêm cái này ngay bây giờ và đăng ký như tôi đã nói sẽ giúp bạn truy cập vào tất cả những thứ này và đảm bảo rằng bạn xem tất cả các khóa học khác mà tôi có trong phần mô tả của video này, cảm ơn rất nhiều vì đã theo dõi các bạn, chúng tôi sẽ gặp lại các bạn trong phần tiếp theo.