Now that we've talked all about routes and the static keyword in detail, let us finish off the work that we started with the routes.

And I'd like to pose that to you as a challenge.

Can you create static IDs for all of the remaining screens and replace the strings in our main.dart file similar to how we've done it with the welcome screen? And as part 2 of the challenge, I'd like you to take a closer look at the welcome screen.

Notice that there are two buttons. At the moment the buttons do nothing.

But needless to say that the top button should send users to the log in screen and the other one should send users to the registration screen.

As part 2 of this challenge, I'd like you to implement this functionality. Modify the code in the welcome\_screen.dart file so that when a user presses a button they can navigate to the corresponding screen.

I'll give you a few seconds to pause the video to complete the challenge.

Ready?

Here's the solution.

So for time's sake, I'm going to simply use copy and paste and I'm going to create this one which is going to be registration screen.

And then I've got one called log in screen.

And finally I've got my chat screen.

So now that I've created all my static const in my stateful widgets, I can now come back here and modify all of these hardcoded strings to tap into the actual static property. So that now if I make any typos, then I would get an error and Android Studio would tell me that it doesn't know what I'm talking about and it would prompt me to fix my errors. So finally we have our chat screen screen.id and now we're done.

We have our four routes defined.

The key is a string and those strings come from our static const that are associated with each of these screens.

And our values are going to push a particular screen onto our navigation stack.

So now if we take a closer look at the welcome\_screen.dart file, you can see that we've got two buttons here.

We've got one that says log in and another that says register. And when each of these buttons are pressed we want to go to the respective screens.

Now we can use our named routes and static constants to implement the navigation functionality for our two buttons on the welcome screen. All we need to do is remove our comments and instead tap into the navigator and use our pushNamed method to push the current context to the routeName.

And remember previously we'd been typing hard coded strings in here which again creating more sources of potential errors.

Well in this case, we're going to use our static properties.

So when this button, the login button, is pressed we want to go to the log in screen.

And for this file to know about the log in screen, we of course have to import the log in screen. And down here we can now say LogInScreen.id to tap in to that static constant property.

And now when our log in button gets pressed, then we should be pushing the log in screen onto the stack.

And when our registration button gets pressed, then we should be pushing our registration screen onto the screen.

So let's import registration screen as well so that all code no longer has any errors, and finally let's close that off with a semicolon.

So if we run our app now when I click on the log in button, it'll take me to a log in screen with the log in button. And if I click on the register button, it'll take me to the registration screen.

So that's pretty cool.

We're using our static constants in order to make our code a little bit safer in this case and also do it in the most efficient way possible.

Now just a word of warning here that when you created your named routes, if you followed the style that we saw on the Flutter documentation, say we had a log in route, a registration root and a home root, well that works fine.

So whenever we're using this forward slash if you come from web development you'll know that to be our different route.

So go to home or go to log in or go to registration.

And this kind of comes from that URL format right? where we have our home page slash registration etc..

But when you're doing this in Flutter, if you have your routes start with that forward slash you must define a route that has nothing after the forward slash.

So for example, you can't have something that looks like this.

You can have a /welcome /login /registration and maybe here this one would be a /chat.

This will crash your app.

And if you tried to set the,for example initial route to the welcome screen, then you can see this is the error that you'll get. You'll get something about a global key was used multiple times inside one child's list.

And this is perhaps not the best error message, even though Flutters really good at giving you precise and clear error messages.

In this case the reason is because if you're going to use that forward slash format, one of these routes has to be forward slash nothing.

So that way you can go to /log in or /chat or registration and it will work just fine.

So make sure that even if you're using static properties or whatever it may be if you're declaring it inside here, if you're going to use forward slashes make sure that one of those routes has nothing after the forward slash.

But in this case I think it's actually more clear to do it the way that we did, which is to have a ID for each screen such as welcome screen or the registration screen.

And we don't have to follow that forward slash convention.

