All right. So most of our input page is now complete and all of the components now work on this page as expected.

Now there's just a couple of styling things to tidy up.

For example at the bottom here, we've got this calculate button and ideally we'd like to maybe format that text to make it look a little bit better.

So let's go ahead and add a style to this.

And of course we're going to create it as a constant in here.

So I'm going to create a new const called kLargeButtonText style or you could call it anything you like.

And it's going to be a text style widget.

And I'm going to set the font size as probably 25 and the font weight as bold.

And now I'm going to apply this constant to my text here.

So k It was LargeButtonText.

All right let's check that out.

So that looks a lot better.

And now all I need is to center my text into that container so that it's right in the middle here rather than all the way to the left.

Now there's just one last thing which is our calculate text widget is just a little bit too close to the bottom.

It would be nice, especially on an iPhone where we know that this is an interactive area,it would be nice to bring that word up a little bit so that the user target area is a little bit higher so they don't accidentally quit the app.

So in order to do that, we can add a little bit of padding to our container. And we're going to add some padding that only adds it to the bottom.

And I'm going to put 20 pixels of padding here.

So now when I hit save, you can see it pushes it right up so that it's actually properly within the safe area.

And that's pretty much our input page complete now.

So we're ready to move on to the next page in our application which is the result page.

So that's what we're going to be doing mostly in this lesson and we're going to be building it out so that it can look like the one that we have in our demo here, where we have a card and we have the results and we have another button to head back so that we can recalculate our result as necessary.

We've already created a result page that looks really really simple. And we can get to it to look at it by clicking on the calculate button.

So we don't actually really need this text in the middle.

That was just to make sure that we can tell that this is the second screen.

But what we do need are all of the components that's going to make up this page.

So let's go ahead and build it out.

Firstly it's going to be a top-to-bottom vertical layout.

So I'm going to add my column as usual, and my column is going to be spaced out evenly.

So I'm going to go for a space evenly for my main axis alignment.

And if you forgot about what that does, just hit CONTROL + J or CONTROL + Q On Windows and it tells you exactly what it is that this particular selection will do.

Now we're ready to add our children to our column. And the first child is going to be an expanded widget which is going to contain just a container. And our container is going to have a child that is a little bit of text.

The text is just going to say the words 'Your Result'.

So if we hit save we can see that it just shows up as a little bit of body text.

So we're going to need to style it up a little bit.

Now while I'm going along creating the entire user interface for the second page, all that I'm doing is just following along with the design that I'm seeing on this page. Although I've simplified it a little bit so that we don't spend ages adding loads of little components, I still kept most of the design consistent with what we see here.

Now if you want to simply create this design all by yourself then feel free to just use this screen as your guide to create the design from scratch by yourself.

But if you want to follow along with me, then I'm gonna quickly style it out so that we have the same code and we stay consistent with each other.

So now that I've added my text, I want to add a little bit of style. And when I'm adding a style, it's easier to simply just add it to my constants because I might want to use it a little bit later on.

So let's add a little bit of style for that title.

So I'm going to add a const kTitleTextStyle and this is going to be a text style widget. And the font size I'm going to set to quite large, maybe a 50 and then the font weight is going to be bold. All right.

So now I can use this kTitleTextStyle in my result page as long as I import the constants file.

So let's go ahead and add that in.

All right.

So now that I've added this we can check it out and it should look a lot bigger like this. Let's go ahead and add our next widget inside our column.

So after the expanded widget, I'm going to add another expanded widget but this one is going to be a lot larger.

I want it to take up more space in this column so that it takes up maybe five times as much as this one or this one.

And in order to do that, I'm going to use the flex property.

So by default all expanded widgets get a flex of 1,so they're laid out evenly.

But in this case I'm going to give this one a flex of 5, so that it takes up more space. And the child that's going to be in this expanded widget is going to be a reusable card widget which we can only get access to if we import the reusable card widget.

So we're only adding in the files that we actually need.

So let's create our reusable card and it has color as a required property.

So I'm going to put in the constant that's the active card color.

Let's check that out and make sure that it's taking up as much space as we want it to.

So at the moment this is taking up 1 times the available space on the screen,and this is going to take up five times the height of this one.

Now I'm ready to add a card child to my reusable card.

And this is going to contain another column.

So it's gonna be a column inside a column.

And in this column, we have a main axis that is going to be again spaced evenly.

And we're also going to add a cross alignment that is going to center everything in the horizontal axis.

And then we're ready to add our children which is going to be some text widgets.

So the first text widget is going to be the result of the BMI, so we'll say overweight, normal or underweight.

And then the second text widget is going to be the actual reading rounded to one decimal place.

And finally we have an interpretation for their weight so that the user can actually understand what this BMI means.

Let's go ahead and add our first text widget which is simply going to be a result.

So I'm gonna start out with just the string of maybe normal over here. And then we're gonna give it a style which is going to be relatively large and I'm going to call it resultTextStyle.

So this is going to be a green color as I've picked it out from this particular design.

So the color is going to be a custom hex code and it's going to be 0xFF, so fully opaque and then 24D876. And you can see that it shows up as a nice bright limy me sort of green.

Now I've got my color down, I'm going to add some font size. Let's keep it about 22, should do.

And then the font weight is going to be again bold.

Forgot to add the k here so let's just change that to kResultTextStyle.

All right.

So now we can go ahead and apply this to our results page right here.

So it's going to be the kResultTextStyle.

Let's check that out.

All right.

So it's looking pretty good. But our card is now shrunken to a very small size.

So if we want to stretch it out, all we have to do is add a cross axis alignment property and make it stretch across the screen.

So now our card will take up as much width as it possibly can.

And in the center we've got that normal interpretation.

So this is the first text widget and the second one is going to be the actual BMI reading,so it's going to be another text widget inside the column. And let's start this one out with a , I don't know,18.3 or whatever number comes to your mind.

And it's going to be as a string rather than as a number because that's what we need inside the text widget. And then we're going to style this one up as well,and it's gonna go at the bottom here.

So this is going to be the kBMITextStyle and this is going to be quite large.

So we're going to change the font size to maybe about 100 just to make it super large and also change the font weight to bold as well, just so that it has the biggest emphasis on the screen. So that's the kBMITextStyle. And you can see it's absolutely massive.

You can't miss that when you head over to the screen.

Now all we need is the final text widget which is going to be the interpretation.

And in this case I'm just gonna write a little bit of 'Your BMI result is quite low.

You should eat more'.

And we're of course going to style this up as well.

So let's add a kBodyTextStyle and this is going to be a slightly smaller style so the font size is gonna be about 22 and we're not gonna change anything else about it. So let's go ahead and add the KBodyTextStyle and if we hit save right now, you can see that by default all text is aligned to the left.

So if we wanted to change the text alignment we have to do that inside the text widget.

So we'll check the text align and we'll will make it centered. And this way it goes right in the middle so that it looks more consistent with our design. Now the final thing that we need is that button at the bottom here telling the user that they can recalculate their BMI result if they think that they inputted something wrong.

This is going to be exactly the same as the button that we had earlier which is the calculate button.

So instead of creating it from scratch again, let's go ahead and select the widget that we want to extract and go into our Flutter outline and it's also highlighted over here and you can see it the last one in our tree in our column.

So let's right click on it and extract our widget.

I'm simply going to call it the bottom button.

You can call anything else you like.

And we're going to click refactor.

So now it takes our bottom button out and we can delete that new keyword if it pops up for you.

If it doesn't, then it means the Flutter team updated the tooling and it's no longer using the earlier version of Dart.

So the latest version Dart there is no new.

And then we're going to delete the the constructor for our bottom button and we're going to set it up with a custom onTap and some custom text because these are the parts that will differ between the first page and the second.

So let's go ahead and add some properties in here.

So we've got a final function that's going to be called onTap,and we're also going to have a final string that's going to be the buttonTitle.So now we can create our custom constructor which is going to be bottomButton and this is going to take two inputs and both of them should be required because it's a button.

It probably needs some text and it probably needs to handle a tap.

So this.onTap and also add required this.buttonTitle.

Now we can go into our gesture detector and I'm going to simply cut out all of this because I'm going to need it a little bit later on. And instead I'm going to use the onTap that gets passed in to the bottomButton. And I'm also going to use the text that's passed in from the buttonTitle.Now this is completely refactored.

We can go into here and add those properties in.

So we've got our button title which is going to be the word CALCULATE in all caps, and then we're going to have onTap which is where I'm going to paste in everything that I had from before.

So it's that callback which pushes the next screen, which is the result page, onto the navigation stack.

And if you find it helpful, you can add some commas so that Dart will reformat it properly for you.

So now I'm going to take all of this out of my input page. And let's create a new Dart file and we'll call it bottom\_button and now I'm going to be have to paste in that entire class.

So let's add that into here.

And again import our material package so that everything works and also import our constants so that we know what these colors and sizes and styles are.

And while I'm at it, I might as well separate out my round icon button as well just in case I need it somewhere else too.

Doesn't make sense to keep it inside the input page and hold on to it there.

So let's create our round\_icon\_button and here I'm going to cut this out and paste it into this new file.

Now we can again import our material package and that's all we need.

Now I can go into my import page and import the bottom\_button.dart so that it knows what the bottom button is, and also import my round\_icon\_button.dart file so that it knows what the round buttons are.

So we have no more errors in this page and this entire page is now dedicated to the user interface and the functionality of the inputs screen in our app.

If you want you can actually quite easily add a folder to organize your lib because as you can imagine,as you create more custom components and more screens it can get a little bit messy. So you can go into file, new and we can select new directory. And we can call this maybe components, and we can also create a new one that is going to hold on to our screens so we'll call this one screens.

And now we can move our input page and our result page into screens and make sure that you keep the search for references checked so that everything gets moved along correctly.

And we're also going to move our components,so the bottom button, the icon content, the reusable card and round icon button into our components folder just so that everything looks a lot neater.

We've got components,we've got our screens but we've also got some errors. And the errors are because it no longer can see constants in the same folder right?

Whereas previously, the bottom\_button.dart and the constants, they were all inside the same folder.

Now it's actually at a different level.

So all you have to do is delete that and retype it.

So start out with constants and we can search for it and we'll be able to find it inside our package bmi\_calculator/constants and now all your errors will go away.

So let's go ahead and copy this and replace it everywhere where there's an error for finding the constants file and even in our screens we've got also the same issue up here.

We've got our icon content that is now moved inside the components folder,the reusable card that's been moved so we have to restructure all of these and also do the same for all of the other places where we're using components.

So now we no longer have any errors with our imports and all our code will work just fine.

So the final one is our results page.

So let's go ahead and add the location of our constants and also our reusable card.

All right.

So now that we've refactored and tidied everything up, it's now time to actually add in that final component into our column here.

So after that last expanded widget, we're going to add that bottom button.

So again we have to import that bottom button that's inside our components folder, and we'll be able to use it down here to add a bottom button widget.

And we're going to add a button title which is going to say RE-CALCULATE.

So a little bit different so it indicates that we go back. And then we're going to have a onTap which is going to have a callback.

Now when the user taps on this bottom button, what we want to happen is to pop off this second screen,this result screen, and go back to this previous screen so that they can add new inputs and recalculate.

So as we saw earlier when we learned about navigation and route, it's really really simple to destroy and pop off the screen that's at the very top.

All we need to do is write Navigator.pop and pass in the current context.

So now if we hit save and we calculate our BMI, we click on recalculate,we can go back to the input screen. The very last thing that we're going to do is simply to just tweak this top text widget just so that it comes down a little bit closer to the card and have a little bit of margin from the left.

So let's go into that container widget and let's go ahead and add some padding, maybe around 15 of padding so that it comes away a little bit from the top and the left. And then I'm also going to add some alignment just so that it goes down a little bit maybe down to the bottom left.

So now let's hit save.

And now it looks a lot more coherent. It's closer to the card and further away from the top and it looks like it's all grouped nicely together.

Now that we have pretty much all of our design done both for the input page as well as for the result page, there's only one thing that's missing and that's some actual functionality. The actual ability to calculate the BMI.

So for all of that and more, I see on the next lesson.

