Now we can see that our design is really coming together. And it's looking already pretty nice.

But this just a few things that look a little bit different for our info sections here, compared to what　we initially wanted.

And these are really subtle things like maybe a little bit a rounding of the corners, or a little bit of shadow under the white part, to make it feel like it's popping out of the page a little bit. So we can either try and customize what we have to make it look like this, or we can simply use a pre-built widget class called card. And you might have seen this in a lot of the apps that you use from day to day, especially the ones from Google.

This is a very very subtly shadowed rectangle and it has a small curve to all four edges, and it has a decent amount of padding so the contents centered on the card.

And we can create this really really simply by using the card widget.

Let's swap out our containers for a card widget instead.

And the first thing that you'll notice is that we have some red lines. And that's because the card widget doesn't actually have a padding property.

So let's go ahead and delete the padding that we've set for both of those.

But other than the padding, everything else is perfectly valid. Cards have a margin that you can set.

It has a child that can contain items inside here.

And if I hit save right now, you can see that our info sections, automatically start looking like cards.

The only problem is that our cards don't have any padding.

So what can we do when we have a widget that doesn't actually have a property that we want to change because padding doesn't exist for card?

But, if you look through the Flutter documentation, you might come across something called the padding

class. And this is a widget that can be added as a parent to any other widget and it'll give it some padding.

And this is how you might use it.

So you could have a padding widget as the parent to a card widget.

And now you end up with some padding around it.

Now the way that you use padding is quite interesting.

Let's go ahead and add our padding. And we're going to do it around our row because we're going to pad out our row inside our card.

So I'm going to select or put my cursor on the row widget, and I'm going to hold down OPTION or ALT and hit ENTER, to pull up the intention actions menu. And I'm going to select wrap with new widget and that new widget is of course going to be padding. And padding of course has a padding property, and we're going to set it to EdgeInsets.all with 25 points of padding.

Now you can see that we have all this massive space around our row.

So the padding goes around it's child.

It's almost like we're adding a soft blanket around which ever widget it is, that we've wrapped our padding.

In our case, that's our row.

And so the row now exists inside a 25 point padding.

Now, what would you think would have happened had we put our padding around our card instead? Well in that case, we can remove our widget from our row, and we can add it to our card.

And you can see that adding padding around widgets is such a common action that you can actually select it in this menu already. And you can select that, and it comes pre-built with a certain amount of padding.

So let's change that to 25. And you can see that when the padding goes around the card, it doesn't actually achieve what it is that we wanted, which is to get a slightly larger card with a bit of space between the edges of the card and the actual content.

But in this case what's actually happened, is the padding has gone around our card and it almost looks like we've added margins around our card.

It's important to remember that when you're adding padding, the child of the padding widget is going to get a blanket essentially wrapped around it and separated from everything else.

I'm going to go ahead and delete that padding.

And I want to show you that not only can we use the card widget to make our lives easier, we can even get rid of our row widget by using something that's used very commonly with cards in Flutter.

And it's used so commonly that they even mention it in the documentation. Whenever you're looking at documentation on the Flutter website, there's usually a section called see also.

And this very helpfully points you towards other widgets that are used really commonly with this widget.

And here, it points to something called a list tile.

And this displays icons and text in a card.

Well that sounds pretty much like what we need right?

So let's click on it and see what it does.

This is a single fixed-height row that typically contains some text as well as a leading or trailing icon.

So that sounds like a much more convenient way of implementing all of this. Instead of having our row, we're going to use this list tile instead.

So I'm going to cut the row out and I'm going to past it at the bottom for reference. And inside my card, as a child, I'm going to add a list tile. And you can see that list tiles have a number of properties.

The leading area which is on the left of that row, the title which is a bit of text that we could put, it could also have a subtitle.

And finally a trailing icon. Let's add our list tile, and let's give it a leading widget which is going to be our icon.

So I'm going to take all of my icon and I'm going to paste it inside here.

So let's get rid of that duplicate comma.

And now we can add a title to our list title.

And that title is of course going to be our text.

So from the bottom part of our text to the top,

I'm going to paste that inside here.

And now we have a list tile and a piece of text, and we can delete the rest of this.

So that's a lot shorter.

If you hit save, and you take a look at the results, you can see that it's already got a bit of space between the leading icon and the title.

And we've got a little bit of padding around our list tile, and it looks pretty great already.

So as a challenge, try and reformat the other list tile so that it does exactly the same thing, and have a look through the documentation for the list tile, to see what other things you can customize about it. The only other thing that we need to change here is this card. And this card currently has a row in it which we're going to remove.

So we're going to find the row comment and cut out this row, past it at the bottom in case we need it.

And then we're going to add a list tile as the child, and it's going to have a leading widget which is going to be our icon, so icon.

Let's put that in here, and delete the extra comma and then it's also going to have a title which is going to be our text widget here.

And let's paste that in there. Now we can delete all of this, and we can hit save.

Cards, by default, come with a white color. So we can delete this color property without it changing anything about our card.

And you can see that we now have this vastly simplified code because we're able to use pre-built widgets that conform to the material design principles. And all we have to do is to figure out which widget to use and how we want to set its properties.

And through this, we're able to customize the appearance of these widgets to our liking. All that we have to do now, is to get our column to be centered.

So do you still remember how to do that? See if you can challenge yourself to find that out from the column documentation. See if you can figure out how to center our content for our column.

Let's head over to our column class. And you might already remember this, from our previous lessons on the column, but one of the most important things that you can set is the cross axis alignment and the main axis alignment. The main axis determines how the children should be placed along the main axis, which in the case of a column, is of course along this axis.

So that is the one that we want to make centered.

So let's find out where we created our column, and then right underneath it, we're going to add our main axis alignment, and we're going to change it to center.

And now, all of our items will get pushed right into the middle, and it starts looking a lot more like a little business card. So now, the final thing that we need to add, is a little line between this section and this section to separate it. And to do that, we're going to add a sized box because this is a really easy way of adding something with a fixed height inside our column. And I'm gonna give it a height of just 20 points. And then, I'm gonna give it a child. And the child is gonna be a horizontal line.

So in web design, you have something called the horizontal rule which creates a single horizontal line.

But in Flutter, we have something called a divider. And the divider class simply puts a one pixel thick horizontal line, onto our screen.

Let's add that as the child. And let's change its color to maybe a nice light teal color.

And you can see that we now have this line going from edge to edge.

Now what if we want it to be just a little bit shorter?

Well, we can actually change the size of our sized box because at the moment, its width is as wide as the screen.

What if we change the width, instead to 150 pixels. Now our line gets just a little bit shorter, and it's just enough to separate our top group and our bottom group. So in this module, we explored a number of widgets including layout widgets such as containers, columns and rows, but also customizable widgets such as a circle avatar or the card widget or list tile.

And we looked at how we can use the documentation to see what properties of these widgets we can set and customize them to our liking.

Finally we also learn about how to use custom fonts and icons to make our design come to life.

So now it's truly your time to update this app to make it your own.

So personalize it for yourself.

Change it to have your details or your job description or whatever it is that you want to change about this. And once you're done, be sure to share it with the community and everybody else who's taking the course along with you, using the hashtBrew. And I look forward to seeing all of your amazing creations.

So that's all for me for this module.

I'll see you on the next one.