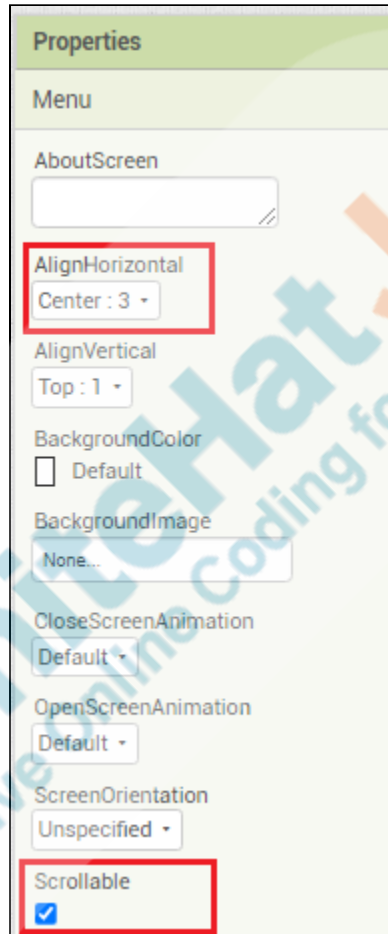


## 2. Design the screen: Then, we will design the **Menu** screen:

### 2.1. Set the **Properties** of the **Menu** screen, as shown below:



**Properties**

Menu

AboutScreen

AlignHorizontal  
Center : 3

AlignVertical  
Top : 1

BackgroundColor  
☐ Default

BackgroundImage  
None...

CloseScreenAnimation  
Default

OpenScreenAnimation  
Default

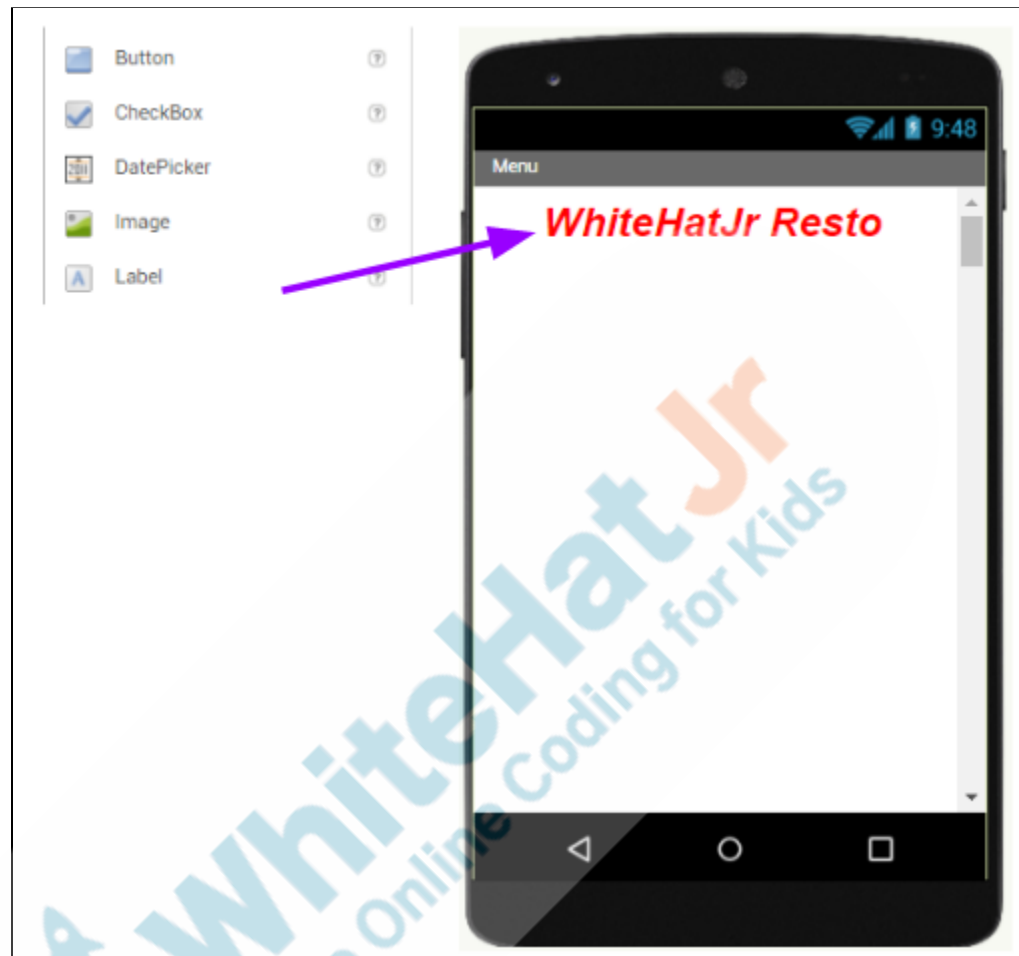
ScreenOrientation  
Unspecified

Scrollable  
☒

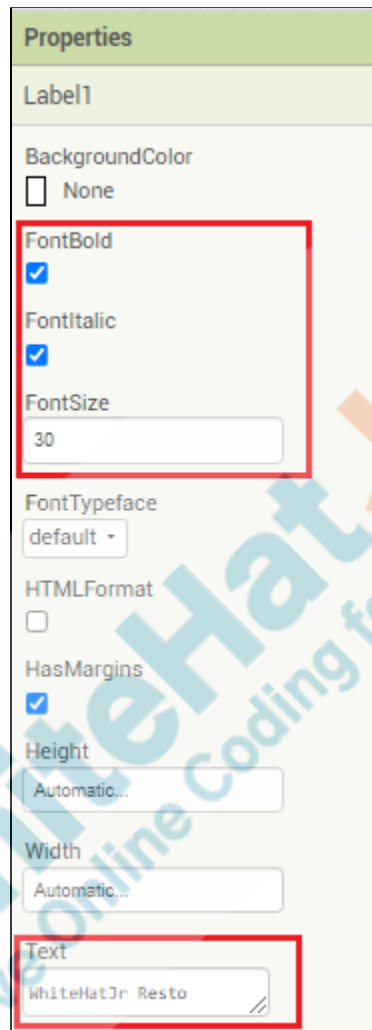
### 2.2. **Add a Label:** Add a **label** to hold the restaurant name:

(*Tips & Tricks: You can copy-paste the **label** having the name of the restaurant from **screen1***)

**Note:** Ask students to keep any restaurant name as per their choice.



- Set the **Properties** of the Label, as shown below:



**Properties**

Label1

BackgroundColor  
☐ None

FontBold  
☒

FontItalic  
☒

FontSize  
30

FontTypeface  
default ▾

HTMLFormat  
☐

HasMargins  
☒

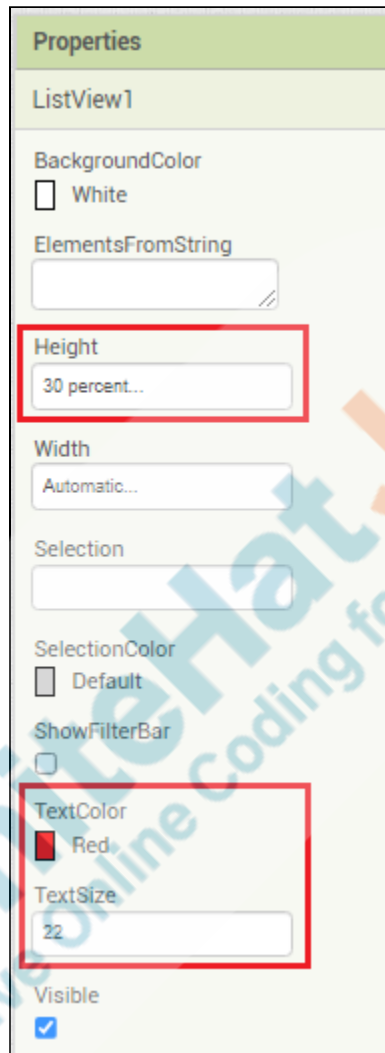
Height  
Automatic...

Width  
Automatic...

Text  
WhiteHatJr Resto

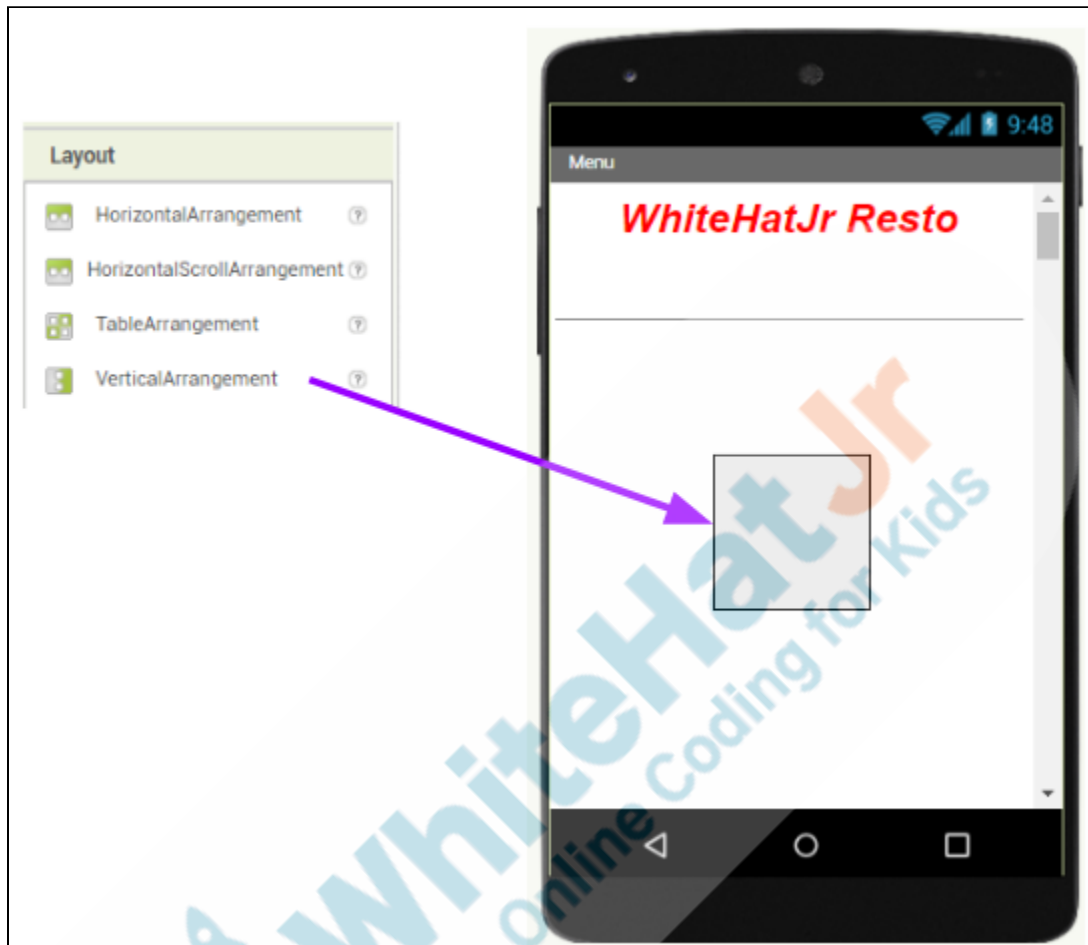
2.3. **Add a ListView:** Then, we will add a **ListView** to display all the menu items available in the restaurant, as shown below:

- Set the **Properties** of **ListView1**, as shown below:

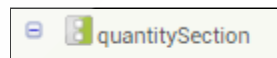


The **ListView** will hold the list of all menu items (Dish name) and their prices added by the admin in the database (**for now admin will add these items details directly in the database using Firebase console but later in upcoming classes we will learn to add items using the Admin App**)

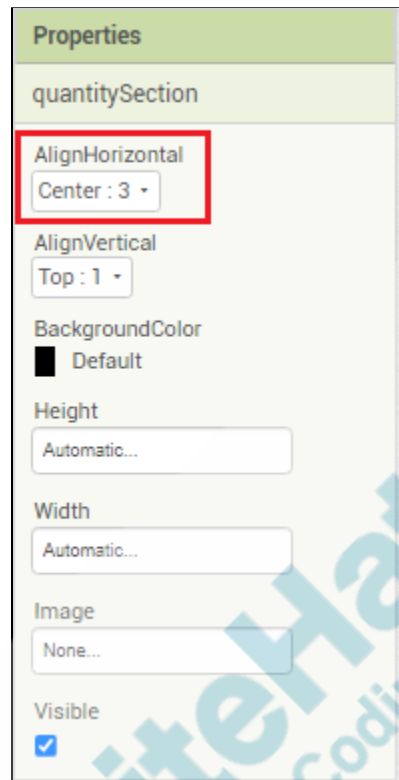
- 2.4. **Add a VerticalArrangement:** Then, we will add a **VerticalArrangement** which will hold the information about the food item. It will hold information such as the name of the selected dish, adding the quantity, subtracting the quantity, getting the price of the dish, confirming the dish, and going back to the Menu.



Change the name of the component to **quantitySelection**.



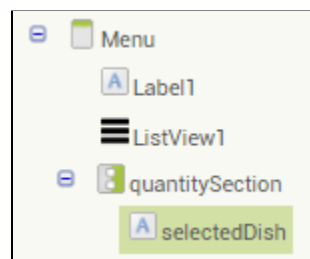
- Set the **Properties** of **quantitySelection**, as shown below:



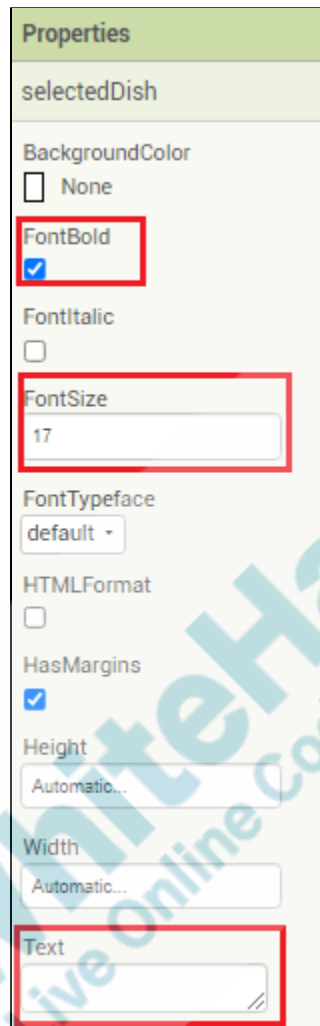
2.5. **Add a label:** Then, add a **label** component inside **quantitySelection**:



Change the **ID** or the name of the component to **selectedDish**:



- Set the **Properties** of **selectedDish**, as shown below:



**Properties**

selectedDish

BackgroundColor  
☐ None

FontBold  
☒

FontItalic  
☐

FontSize  
17

FontTypeface  
default ▾

HTMLFormat  
☐

HasMargins  
☒

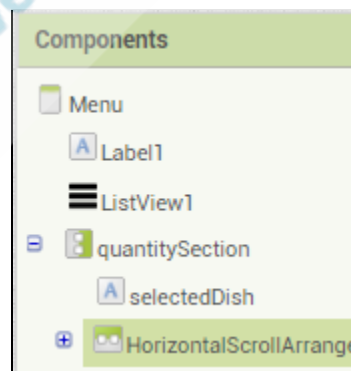
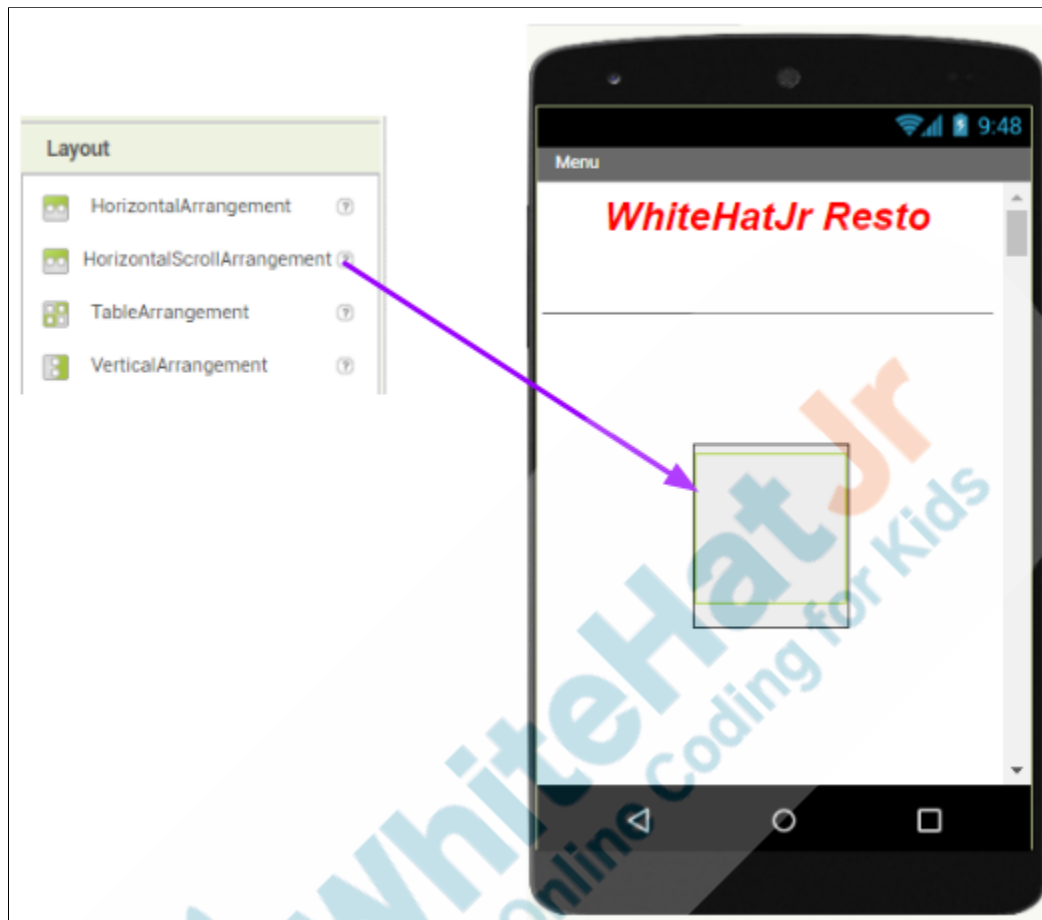
Height  
Automatic...

Width  
Automatic...

Text

2.6. **Add a HorizontalScrollArrangement layout:** Then, add a **HorizontalScrollArrangement** layout inside the **quantitySelection**.



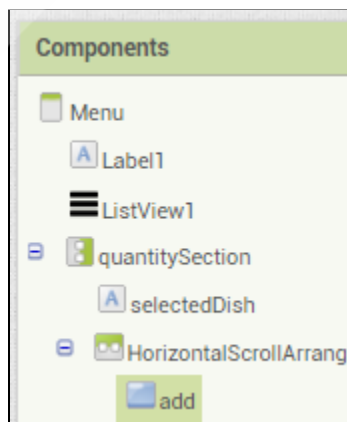


2.7. **Add 2 buttons:** Then inside '**HorizontalScrollArrangement1**', add 2 buttons.

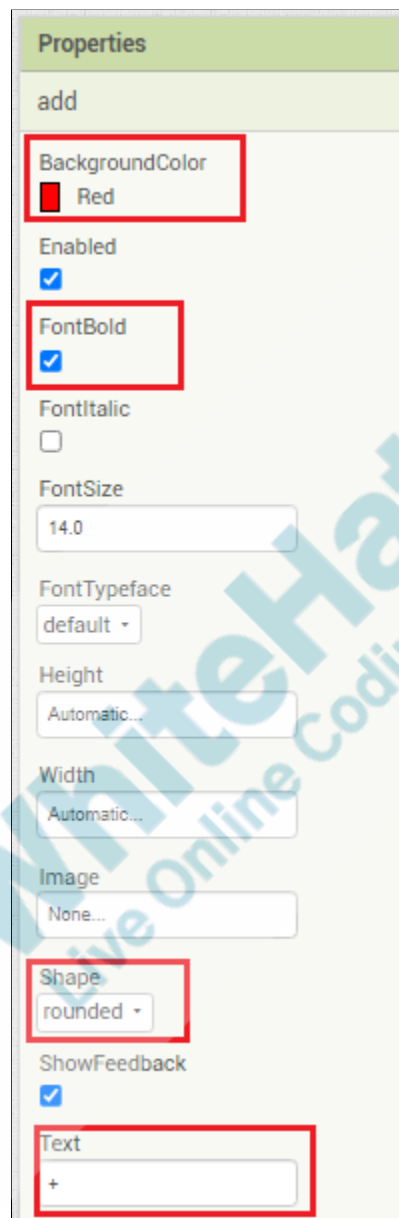
- One button to **increase** the quantity of a food item
- Another button to **decrease** the quantity of a food item



Change the name of the **button** component to **add**:



- Set the **Properties** of **add**, as shown below:



**Properties**

add

BackgroundColor  
☒ Red

Enabled  
☒

FontBold  
☒

FontItalic  
☐

FontSize  
14.0

FontTypeface  
default ▾

Height  
Automatic...

Width  
Automatic...

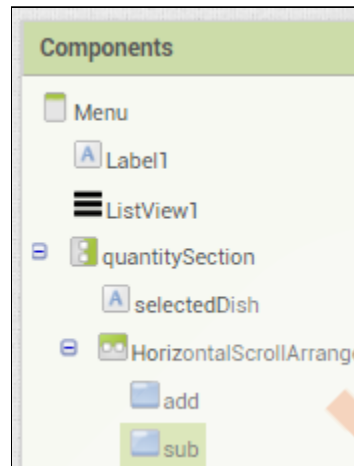
Image  
None...

Shape  
rounded ▾

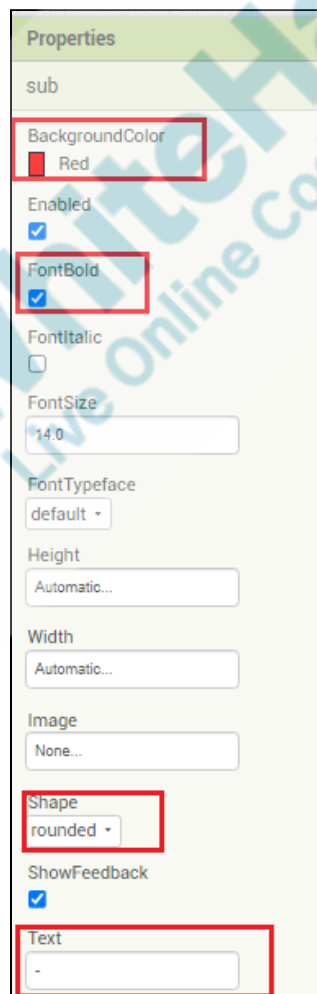
ShowFeedback  
☒

Text  
+

Change the name of the other button component to **sub**.



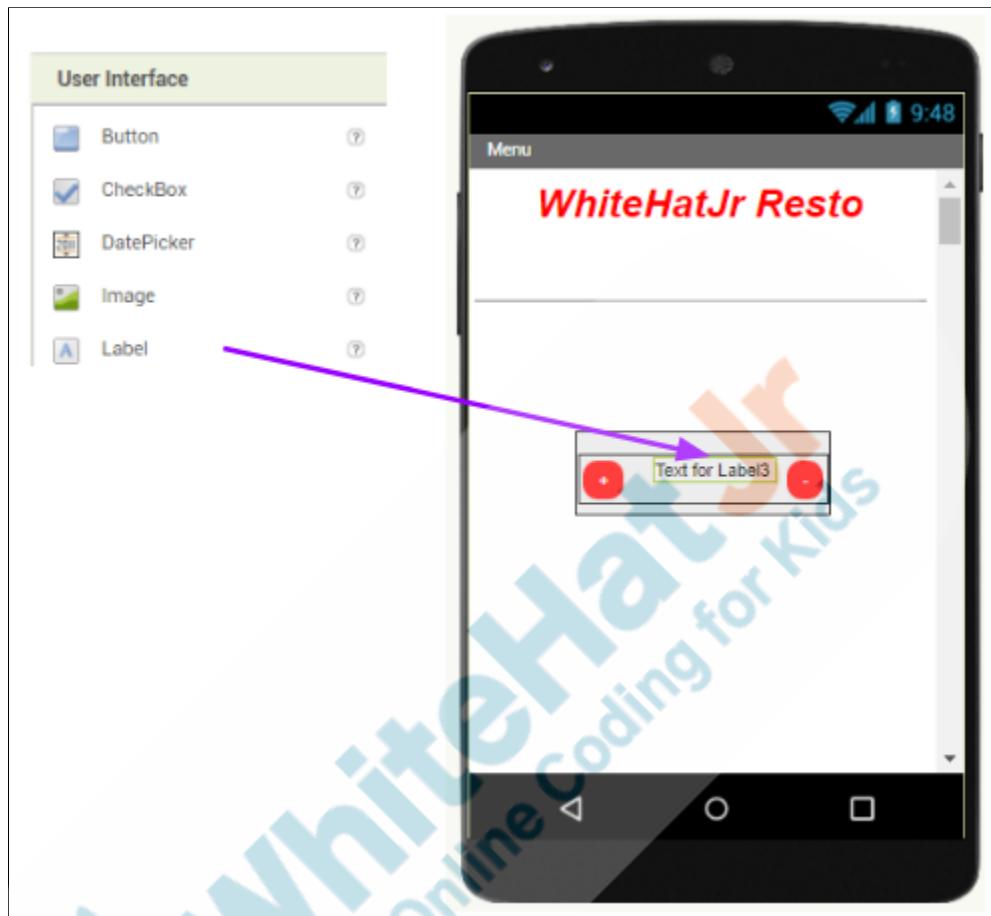
- Set the **Properties** of **sub**, as shown below:



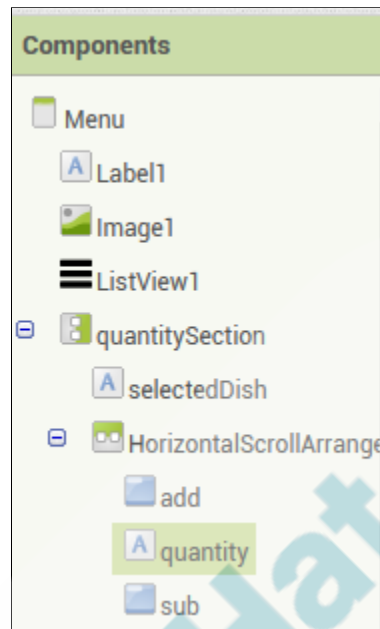
**Output:**



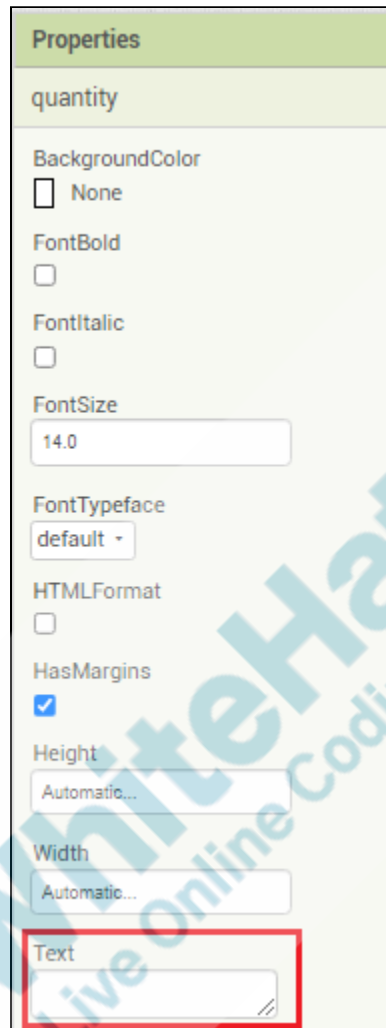
- 2.8. **Add a Label component to hold the quantity of the dish:** Now, add a **label** inside the **‘HorizontalScrollArrangement1’** between **add** and **sub** components.



Change the name of the component to **quantity**:



- Set the **Properties** of **quantity** and we are setting the text property of the quantity label as empty because the value of this component depends on the user's selection. The user will add the quantity using the **add** and **sub** button:



**Properties**

quantity

BackgroundColor  
☐ None

FontBold  
☐

FontItalic  
☐

FontSize  
14.0

FontTypeface  
default ▾

HTMLFormat  
☐

HasMargins  
☒

Height  
Automatic...

Width  
Automatic...

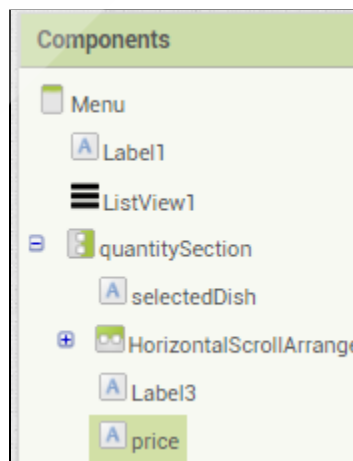
Text

- 2.9. **Add a Label inside the “quantitySelection”:** Then, we will add a **label** inside the ‘quantitySelection’ component below the ‘HorizontalScrollArrangement’. This component will be used to hold the total cost of the food items selected.

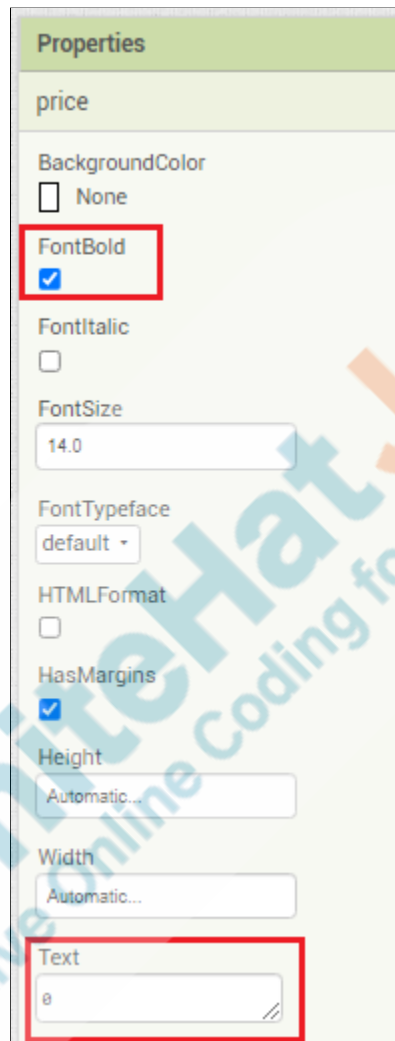




Change the **ID** or the name of the component to **price**.



- Set the **Properties** of **price**, as shown below:



**Properties**

price

BackgroundColor  
☐ None

**FontBold**  
☒

FontItalic  
☐

FontSize  
14.0

FontTypeface  
default ▾

HTMLFormat  
☐

HasMargins  
☒

Height  
Automatic...

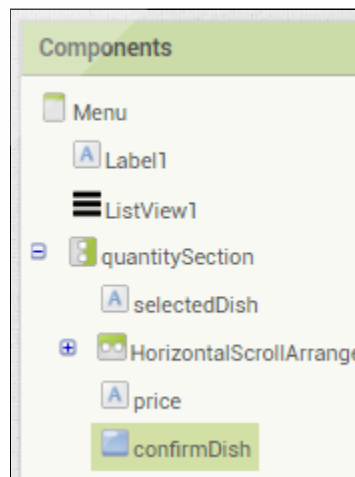
Width  
Automatic...

**Text**  
e

- 2.10. **Add a button named “Confirm The Dish” inside “quantitySelection”:**  
Then, we will add a button inside the ‘quantitySelection’ component below the ‘price’ label.



Change the name of the component to '**confirmDish**':



- Set the **Properties** of 'confirmDish', as shown below:



The image shows a 'Properties' panel for an object named 'confirmDish'. The panel has a green header. Below the header, the object name 'confirmDish' is displayed. The following properties are listed:

- BackgroundColor**: A red color swatch is selected, and the word 'Red' is next to it. This property is highlighted with a red box.
- Enabled**: An unchecked checkbox.
- FontBold**: An unchecked checkbox.
- FontItalic**: An unchecked checkbox.
- FontSize**: A text input field containing '14.0'.
- FontTypeface**: A dropdown menu showing 'default'.
- Height**: A text input field containing 'Automatic...'.
- Width**: A text input field containing 'Automatic...'.
- Image**: A text input field containing 'None...'.
- Shape**: A dropdown menu showing 'rounded'. This property is highlighted with a red box.
- ShowFeedback**: A checked checkbox.
- Text**: A text input field containing 'Confirm This Dish'. This property is highlighted with a red box.

Below the main panel, there is a separate section for 'TextAlignment' and 'TextColor':

- TextAlignment**: A dropdown menu showing 'center : 1'.
- TextColor**: A white color swatch is selected, and the word 'White' is next to it. This property is highlighted with a red box.

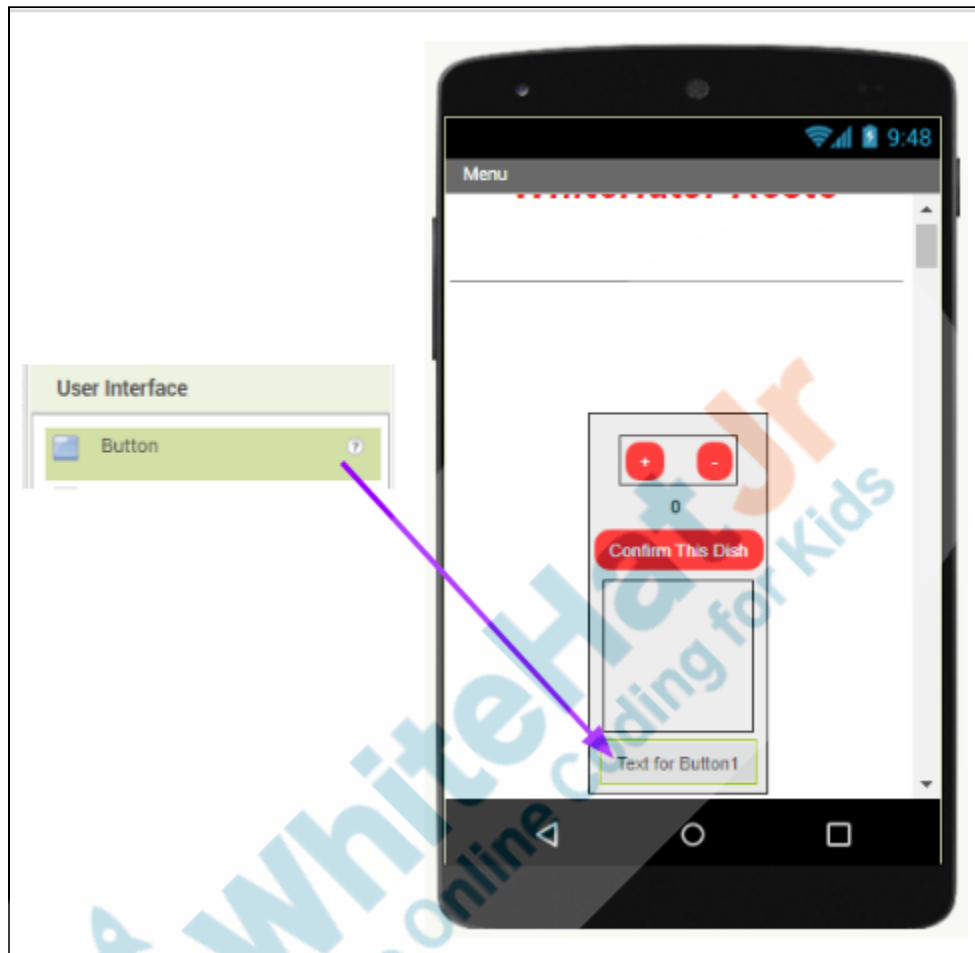
**Output:**



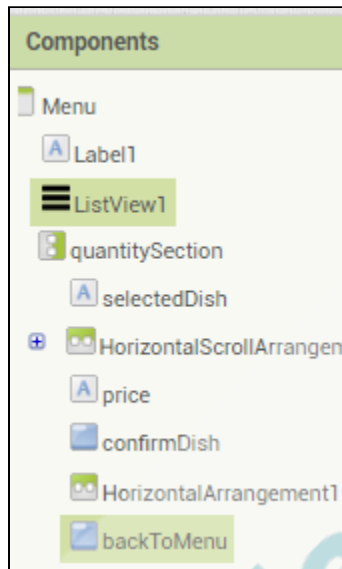
- 2.11. **Add space below the 'Confirm This Dish' button:** Then, we will add a **HorizontalArrangement** inside 'quantitySelection' below the 'Confirm This Dish' button. This **HorizontalArrangement** will be used to add space between the 'Confirm This Dish' button and the next button (**Don't want this dish back to menu**) that we will add.



- 2.12. **Add a button named “Don't want this dish back to menu”:** Then, we will add a button that will allow the user to go back to the **Menu**. Place this button after ‘**HorizontalArrangement1**’.



**Don't want this dish back to menu:** Change the name of the component to '**backToMenu**'.




- Set the **Properties** of 'backToMenu', as shown below:



**Properties**

backToMenu

BackgroundColor  
 Orange

Enabled  
☒

FontBold  
☐

FontItalic  
☐

FontSize  
14.0

FontTypeface  
default ▾

Height  
Automatic...

Width  
Automatic...


Image  
None...

Shape  
rounded ▾

ShowFeedback  
☒

Text  
Don't want this dish back to m

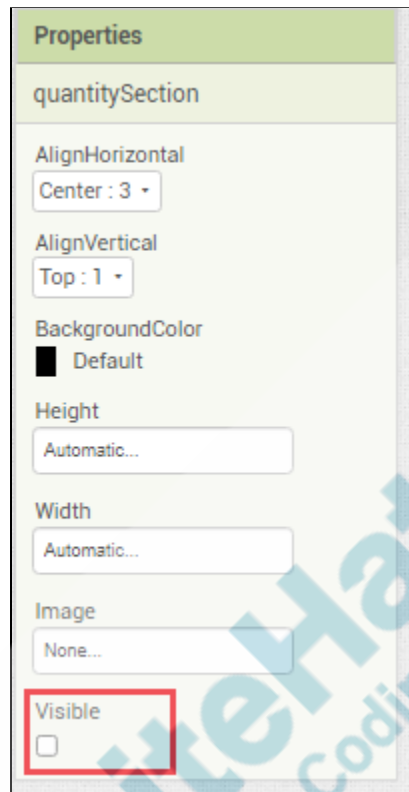
TextAlignment  
center : 1 ▾

TextColor  
 White

**Output:**



- 2.13. **Set the visibility of 'quantitySelection' to false:** Then, we will set the visibility of 'quantitySelection' to **false**, simply by unchecking the **Visible** check box. Because we want this section to be visible only when a user selects a particular menu item.



- 2.14. **Add a Label for the 'Cart Summary':** Now, add a **label** below the 'quantitySelection' component. This label will be used to hold the text '**Cart Summary**'.



- Set the **Properties** of 'Label2', as shown below:



**Properties**

Label2

BackgroundColor  
☐ None

FontBold  
☒

FontItalic  
☐

FontSize  
14.0

FontTypeface  
default ▾

HTMLFormat  
☐

HasMargins  
☒

Height  
Automatic...

Width  
Automatic...

Text  
Cart Summary

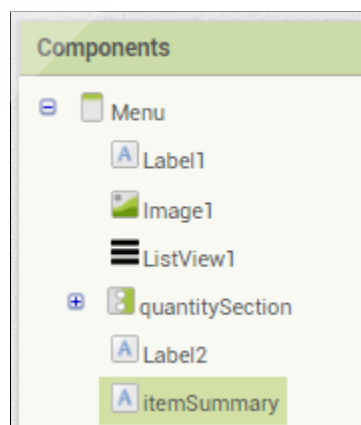
TextAlignment  
left : 0 ▾

TextColor  
☒ Red

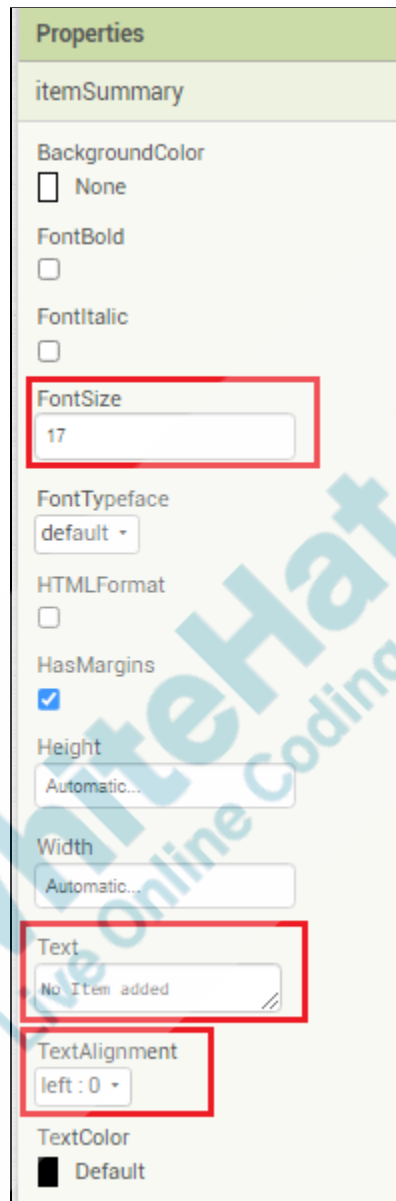
- 2.15. **Add a Label to display the food items added to the cart:** Then, we will add a label below 'label2'. This label will be used to hold the food items added to the cart by the user.



Change the **ID** or the name of the component to '**itemSummary**':



- Set the **Properties** of '**itemSummary**':



**Properties**

itemSummary

BackgroundColor  
☐ None

FontBold  
☐

FontItalic  
☐

FontSize  
17

FontTypeface  
default ▾

HTMLFormat  
☐

HasMargins  
☒

Height  
Automatic...

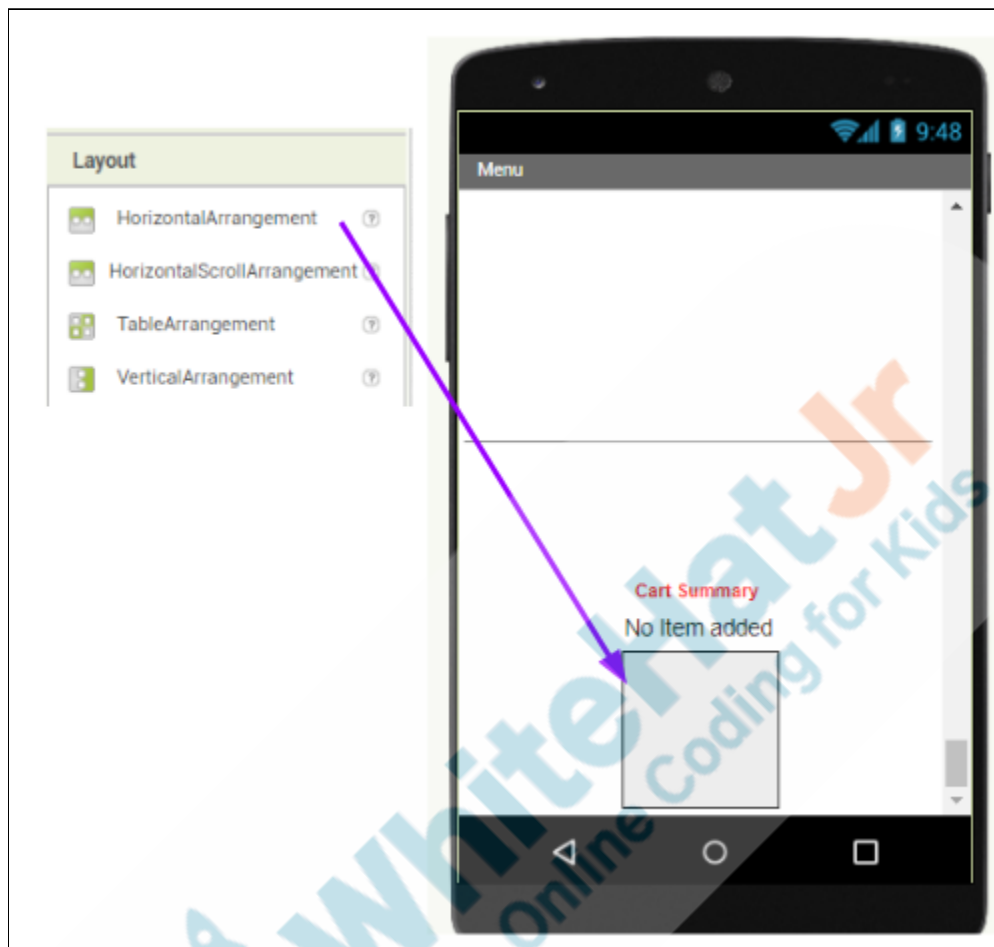
Width  
Automatic...

Text  
No Item added

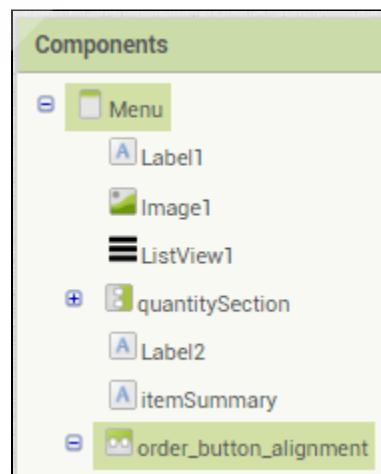
TextAlignment  
left : 0 ▾

TextColor  
☒ Default

- 2.16. **Add a 'HorizontalArrangement' to create space for 2 buttons:** Then we will add a 'HorizontalArrangement' below the 'itemSummary' label component. This section is used to hold 2 buttons:
- One button will be used to **place orders**.
  - Another button will be used to **clear the cart**.



Change the name of the component to 'order\_button\_alignment'.

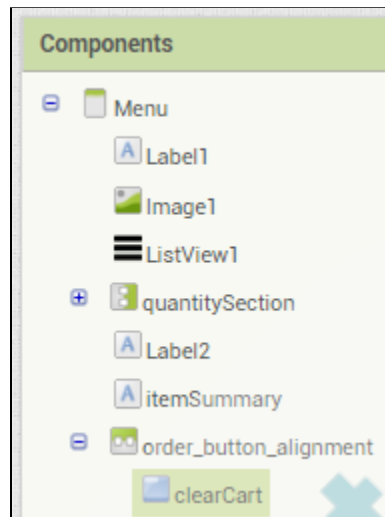




2.17. **Add 2 buttons - “Clear Cart” and “Place Order”**: Then inside the ‘order\_button\_alignment’ component, we will add 2 buttons.



**Clear Cart:** Change the name of the button component to ‘clearCart’:



- Set the **Properties** of 'clearCart', as shown below:



**Properties**

clearCart

BackgroundColor  
 Red

Enabled  
☒

FontBold  
☐

FontItalic  
☐

FontSize  
14.0

FontTypeface  
default ▾

Height  
Automatic...

Width  
Automatic...

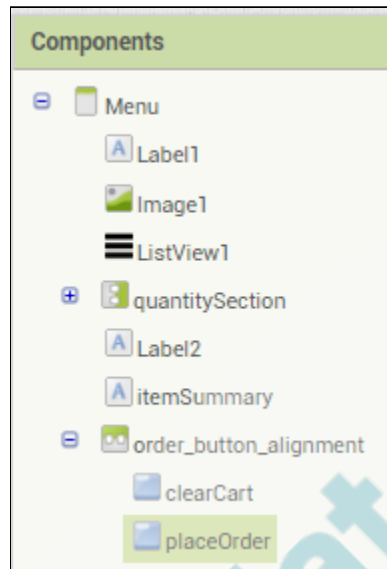
Image  
None...

Shape  
default ▾

ShowFeedback  
☒

Text  
Clear Cart

**Place Order:** Change the name of the button component to 'placeOrder':



- Set the **Properties** of 'placeOrder', as shown below:



**Properties**

placeOrder

BackgroundColor  
 Custom...

Enabled  
☐

FontBold  
☐

FontItalic  
☐

FontSize  
14.0

FontTypeface  
default ▾

Height  
Automatic...

Width  
Automatic...

Image  
None...

Shape  
default ▾

ShowFeedback

TextAlignment  
center : 1 ▾

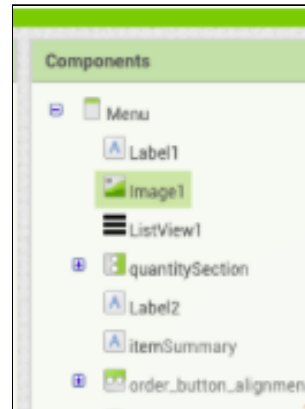
TextColor  
☐ White

And we will keep this button **disabled**. Hence, in the **Properties** section, keep the **Enable** checkbox **unchecked**. The reason for doing this will be explained to you in the next class.

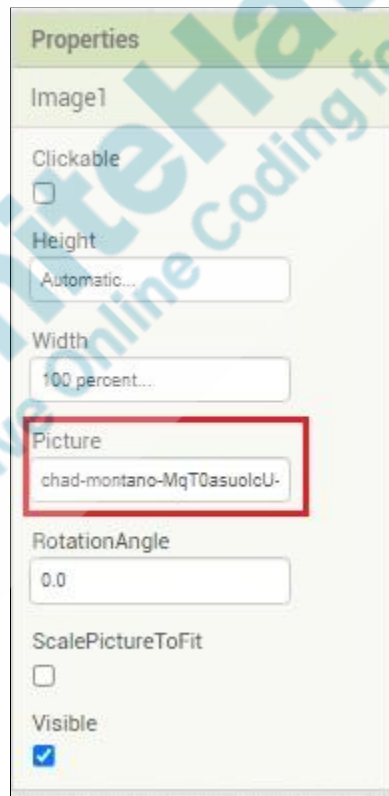
- 2.18. **Display an image for the restaurant:** Now, we will add an image component to display an image that can represent the restaurant visually.



Add this image right above the **Listview1** so that when a user opens the app, the user is able to see the image.



- Set the **Properties** of **Image1**, as shown below:



You can either put any image you want OR use the image from [Student-Reference 2](#).

- 2.19. **Add a Firebase component to the “Menu” screen:** Now, we will add a **Firestore component** to the **Menu screen**. We are doing this because the food items that the user will select will get stored in this Firestore database.


Moreover, further, it will also store the name of the customer/user who will be placing the order.



Set the **Properties** of **FirebaseDB1**:

Log into your **Firebase** account → Go to **Realtime firebase database** → Copy the **database URL link** → Paste the **Firebase database URL link** in the **FirebaseURL** property.





Finally, we have **connected** the **database** to the app.  
Great work!

2.20. **Add the TinyDB component to the “Menu” screen:** Now, we will add the **TinyDB** component to the **Menu** screen.  
We are doing this to exchange the values of variables within the screens of the MIT app.

