PowerApps Popup Message Box

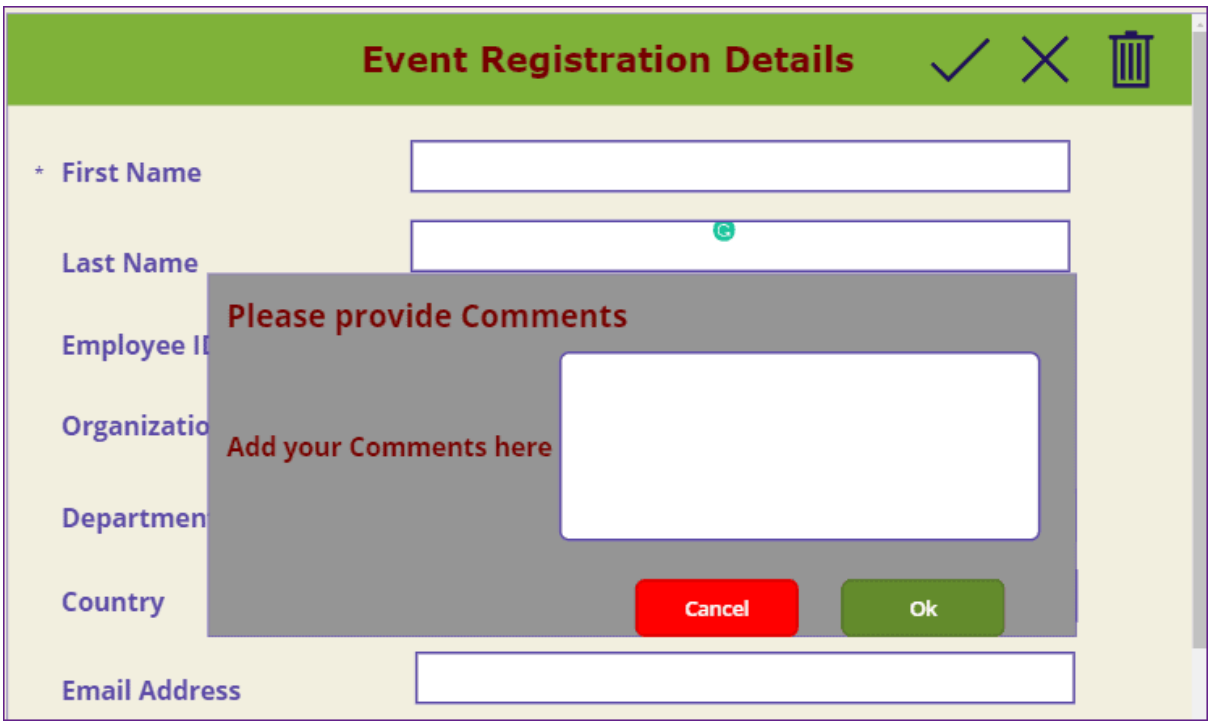
PowerApps Popup or dialog box is a type of box where you can show some message to the user. Let's take an example.

Suppose in your organization, you have created a Powerapps app for all the employees. After creating the app, you need to store some information in a particular location.

At the same time, When you entered a new entry and mistakenly you pressed the **Delete** button instead of the **Submit** button. Then you missed the entire data from there and you need to recreate or re-enter the data again which is time-wasting for all.
So to resolve this type issue, We are creating a dialog or Pop up box in Powerapps. So that a user can confirm before going to do any action.

There is no need to use any Powerapps function to create the dialog box. We can create Powerapps Popup screen by using some Powerapps input controls like Labels, Button, Text input, etc. and all controls will depend upon its **Visible** property.

The below screenshot represents a **Powerapps Popup or dialog box**. When the user will Approve, Reject, or Delete the item, at that time this dialog box will appear.



To **create a Powerapps dialog box**, first of all, We need to do this below thing as:

Insert Powerapps Controls, icons and make them as a Group.

Powerapps Group and Ungroup controls

On the PowerApps screen or Form, add these below Powerapps icon and controls as:

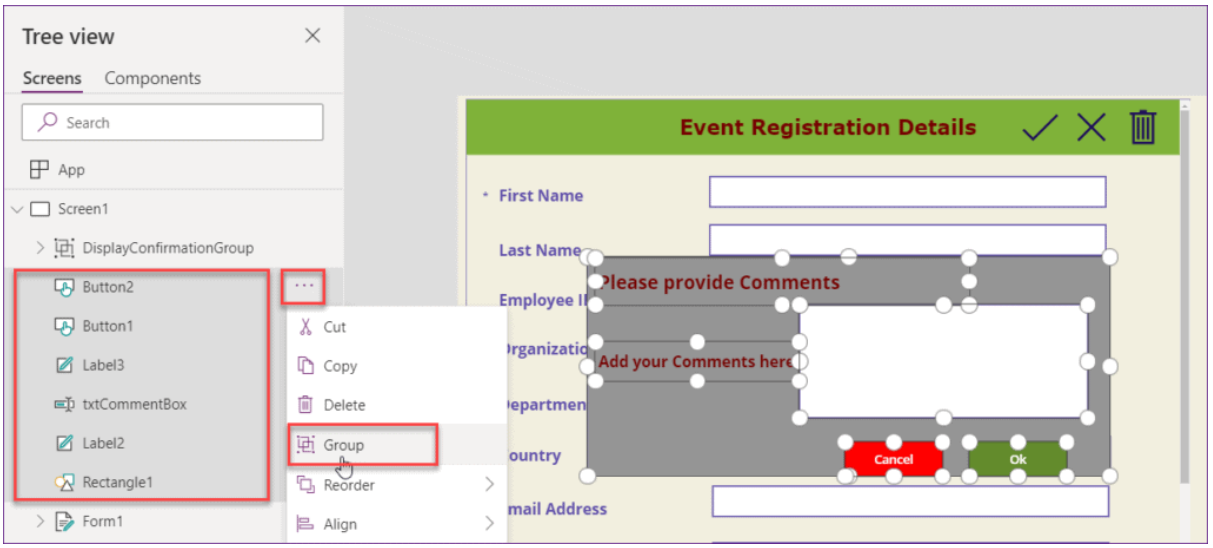
- Insert a **Rectangle** (Insert -> Icons -> Rectangle)
- Add two **Labels** (Insert -> Label)
- Add One **Text Input** control (Insert -> Text -> Text Input)
- Insert two **Buttons** (Insert -> Button)

Now, you need to make a Group by taking all these input controls. The purpose of making a Powerapps Group is, suppose you want to set the

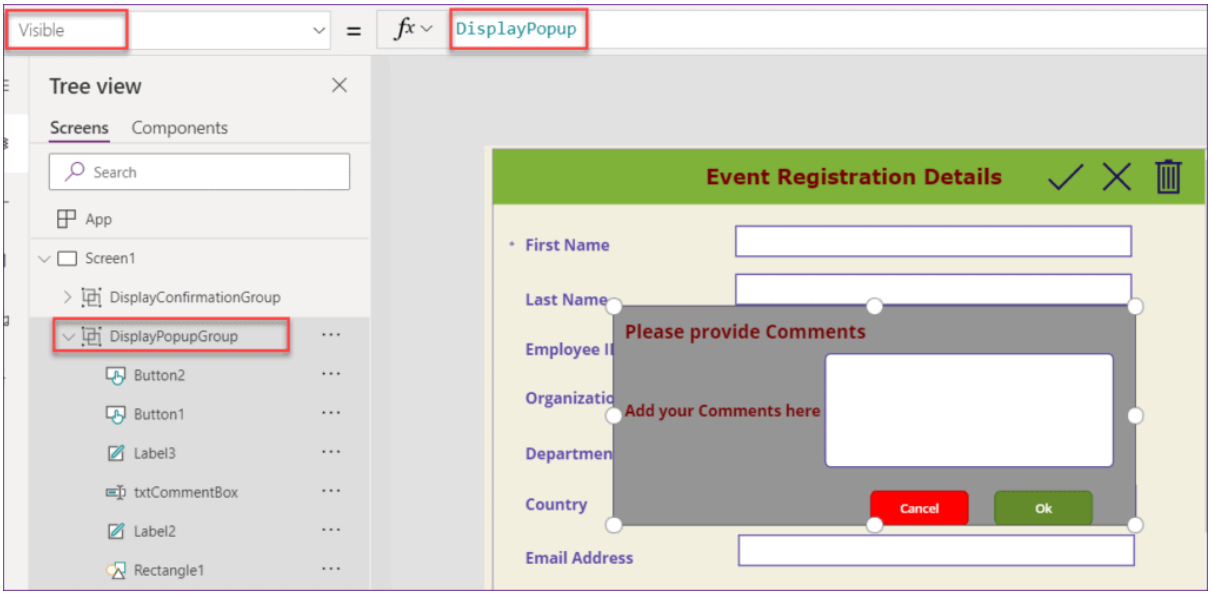
Visible property of all these input controls at a time. Then, in that case, you do not need to set the property individually.

You can make a **Group** control and set its **Visible** property at a time as like below:

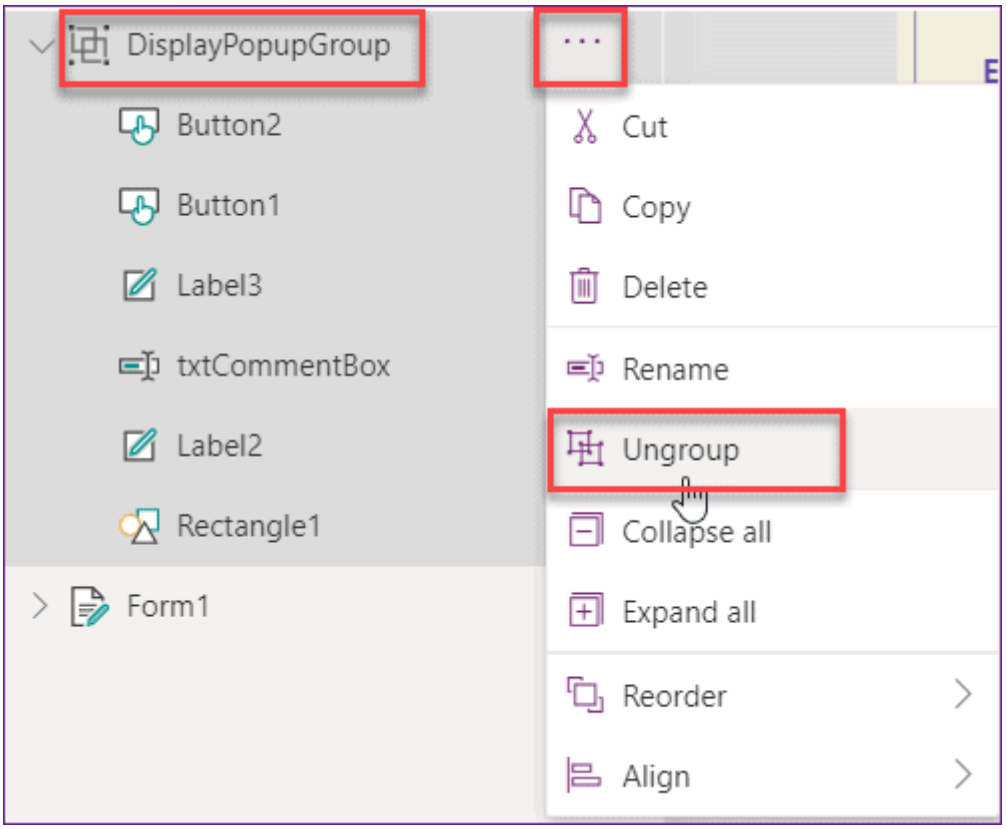
Select (Ctrl + Click) all the input controls those you want to make a Group. Click on 3 dot ellipses (**…**) and then **Group** as shown below.



Enter a Group name as “**DisplayPopupGroup**“. Select the group and set its **Visible** property as “**DisplayPopup”** (it’s a variable) as below. In this way, you can make the Powerapps input Group control and set its property altogether.



Similarly, If you want to ungroup the controls, then just select the group -> Click on 3 dot ellipses (…) -> **Ungroup** as shown below.



PowerApps dialog component

Now I will show you how I made this above Powerapps dialog comment box when the user Approve or Reject the specific item.

- I have inserted the Powerapps input controls and grouped them as above.
- I have three Icons in my Powerapps Edit form as:

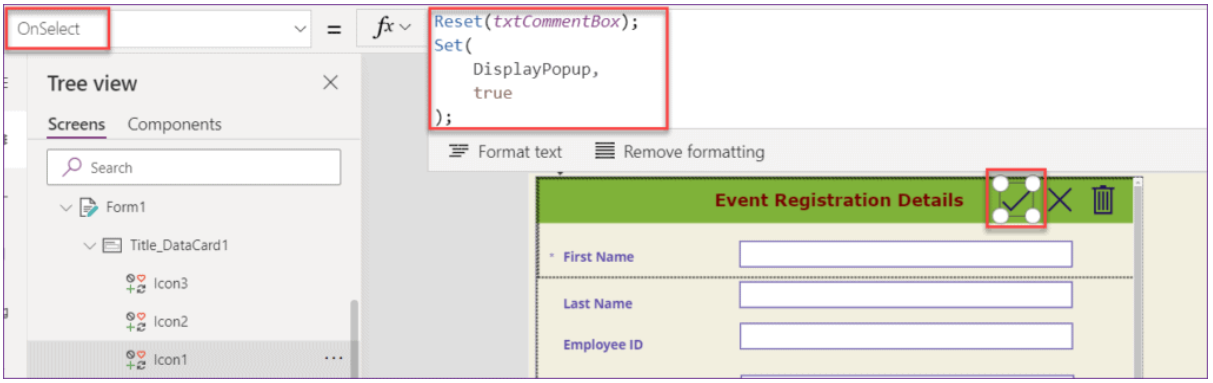
1. **Approve icon**: When a user will click on this icon, a Comment dialog box will appear where he/she can enter the approval comments.
2. **Reject or cancel icon**: When a user will click on this icon, a Comment dialog box will appear where he/she can enter the rejection comments.
3. **Trash or Delete icon**: When a user will click on this icon, a confirmation dialog box will appear where it will ask the user to confirm the item deletion.

- Click on the **Approve** icon and set its **OnSelect** property as:

```
OnSelect = Reset(txtCommentBox);
Set(
    DisplayPopup,
    true
);
```

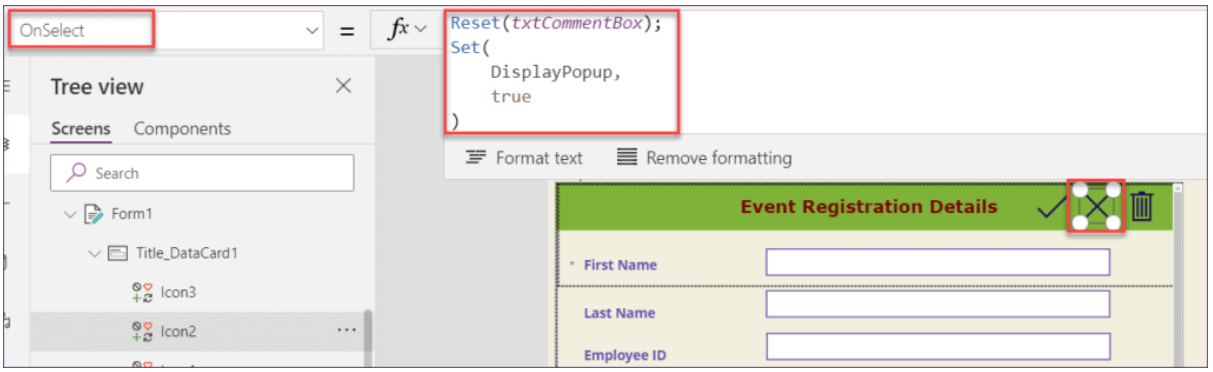
Where,

- **Reset** = It is the property that will help you to reset the text box. This means it will reset and clear the previous text value.
- **txtCommentBox** = Text input control name
- **DisplayPopup** = Variable name
- **true** = This is the Boolean Value



- Similarly, click on the **Reject** or **Cancel** icon and set its **OnSelect** property as:

```
OnSelect = Reset(txtCommentBox);
Set(
    DisplayPopup,
    true
)
```

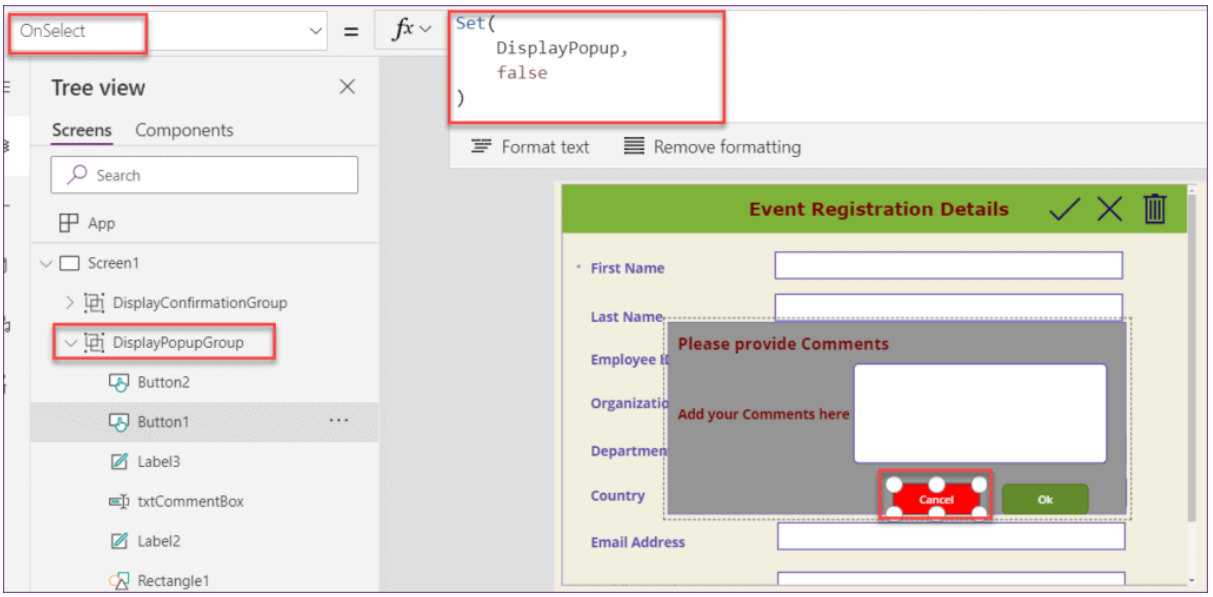


- Now go to the **Cancel** button from the Group (DisplayPopupGroup) and set its **OnSelect** property as:

```
OnSelect = Set(
    DisplayPopup,
    false
)
```

NOTE:

If you can not see the Popup comment box on your Powerapps screen, then just preview the app and click on the Approve icon from the edit form. then close it. Now you can see the dialogue box.



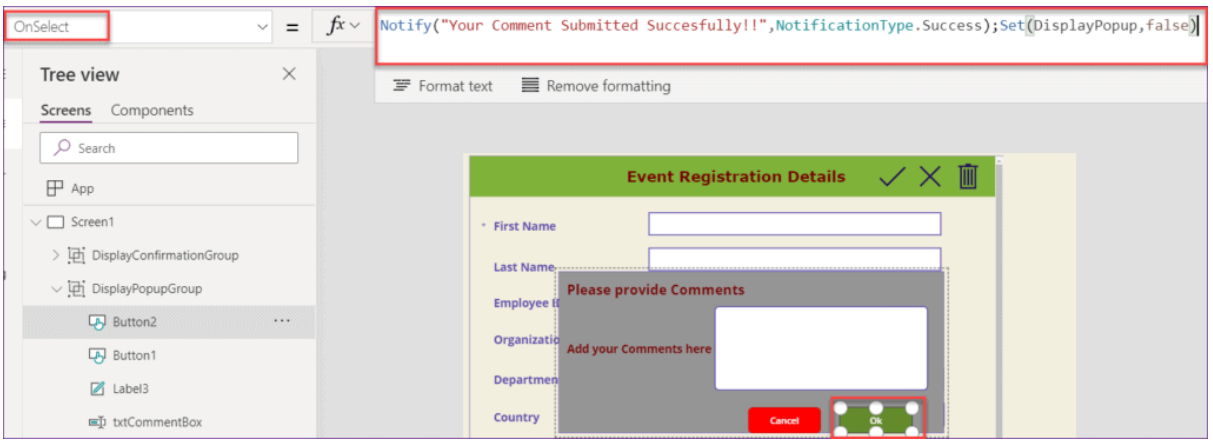
This above code specifies, When a user will click on the **Cancel** button, then the Popup dialog box will disappear.

- Next, click on the **Ok** button and set its **OnSelect** property as:

```
OnSelect = Notify("Your Comment Submitted
Succesfully!!",NotificationType.Success);Set(DisplayPopup,false)
```

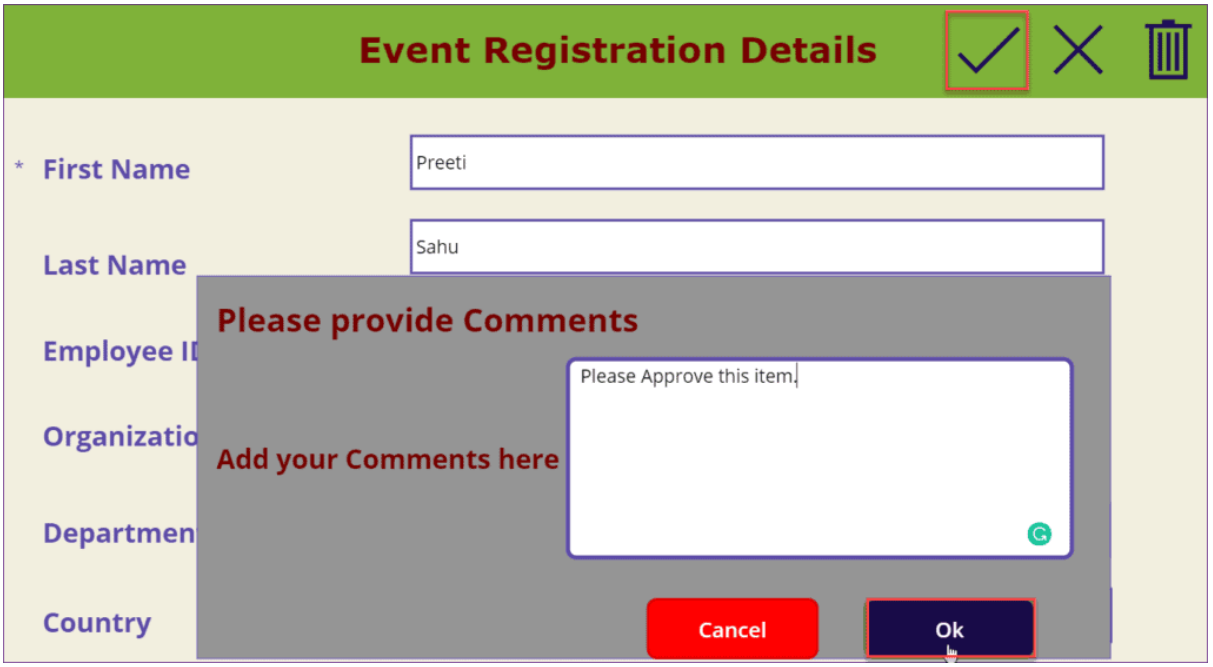
Where,

- **Notify** = PowerApps Notify function helps to display a notification message to the user
- **NotificationType.Success** = This notification argument specifies the message is a success notification message.
- **DisplayPopup** = Variable name that you have created in the Cancel icon's OnSelect property

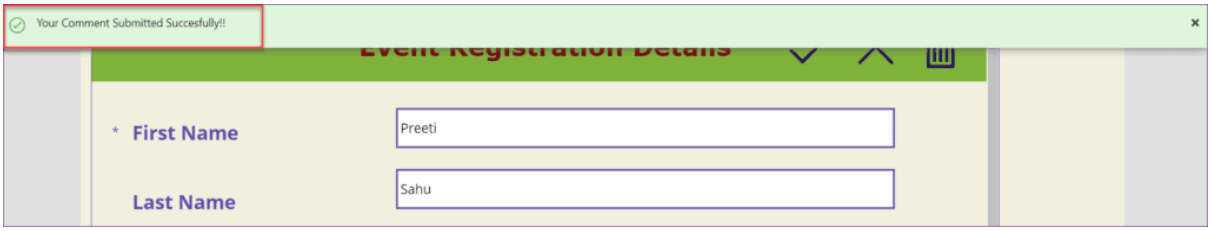


This above code specifies, When a user will click on the **Ok** button, then the Popup dialog box will disappear and a successful notification will appear on the Powerapps screen.

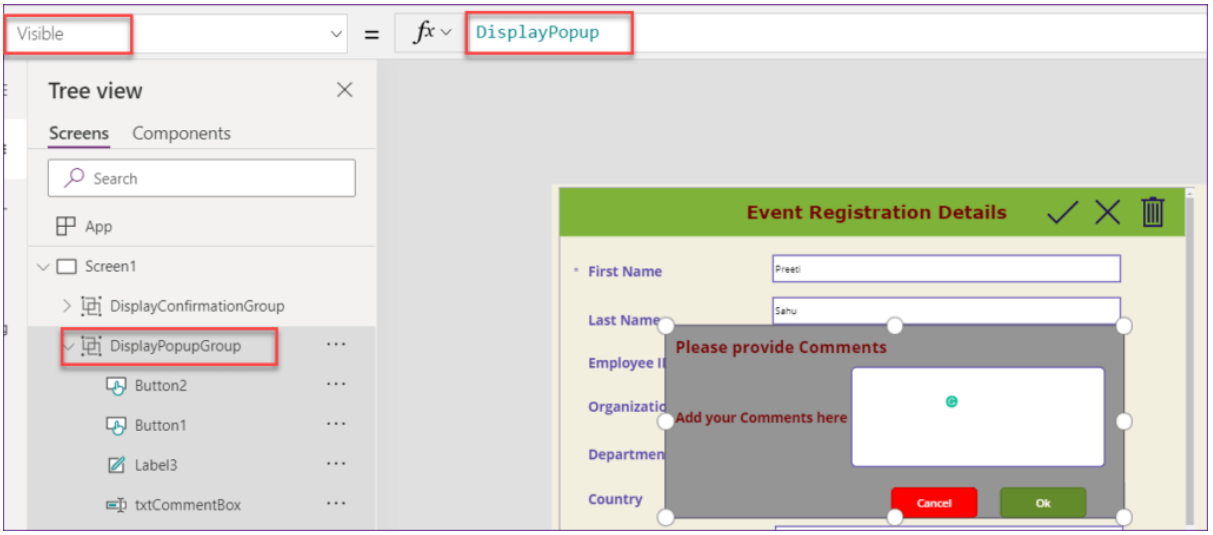
- Preview (F5) the app. Enter the comment in the comment section and click on **Ok**.



You can see the successful notification message as like below screenshot:



At last, you need to set the variable (**DisplayPopup**) on the Group's **Visible** property as:



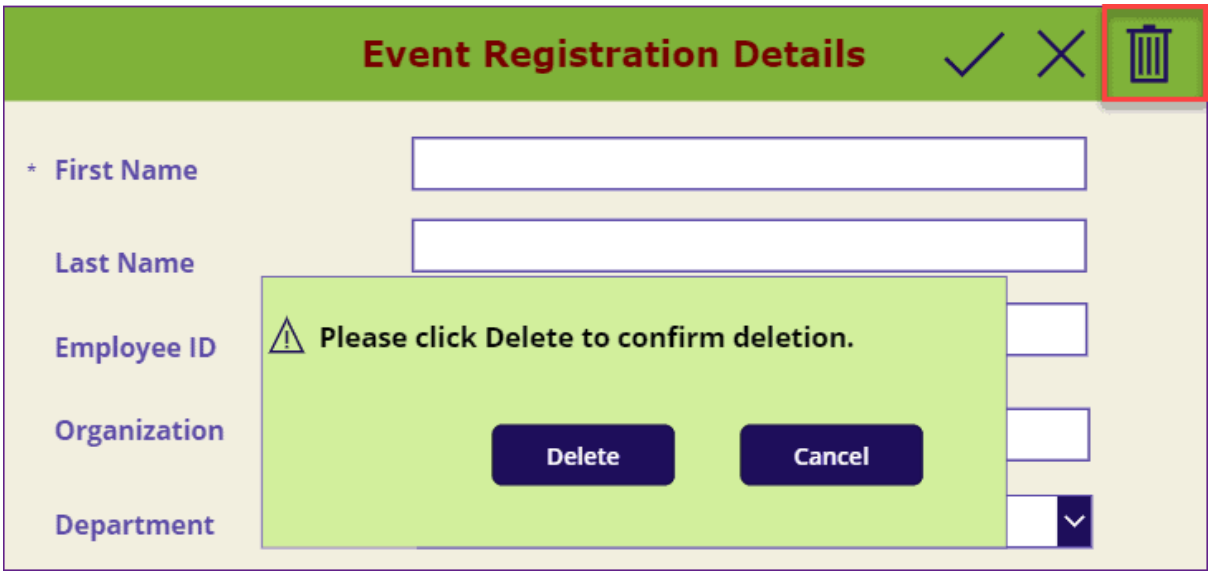
This Comment dialog box will appear when a user click on approve or reject icon from the Powerapps edit form.

PowerApps Confirmation Popup

Here, I will show you how you can create and use the Confirmation Popup box in Powerapps.

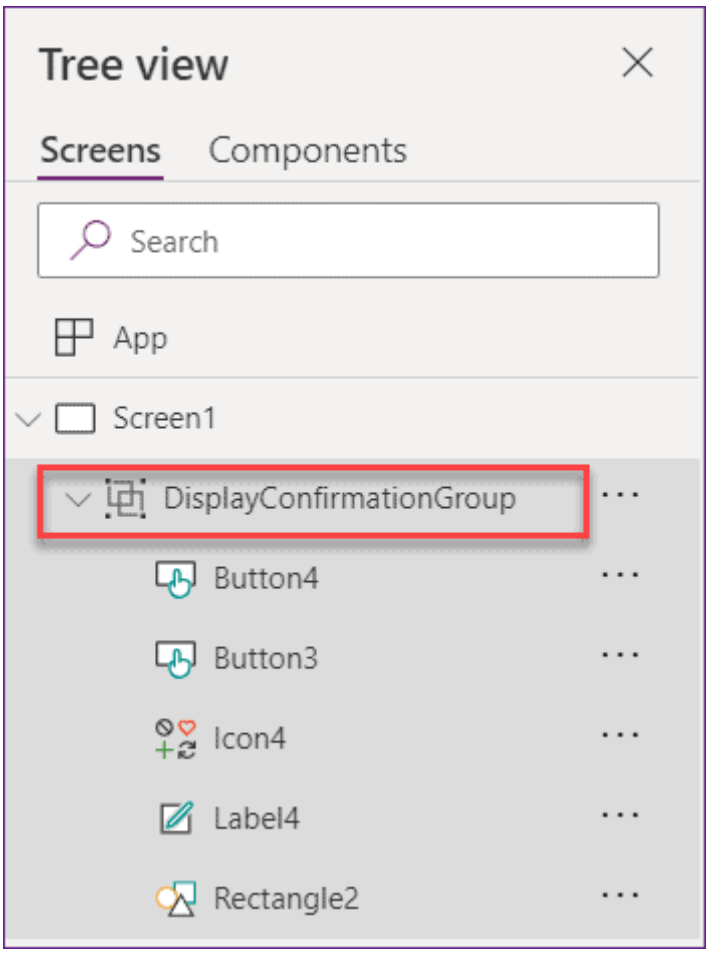
The below screenshot represents a Confirmation Popup dialog box. When a user clicks on the Trash icon from the Powerapps form, then this confirmation Popup will appear to confirm the item deletion.

If the user will click on the Delete button, then the specific item will delete and if click on the Cancel button, then the confirmation Popup box will disappear. So let's start how you can do these things.



To make this Confirmation dialog box, We need these below Powerapps icons and controls as:

- Insert a **Rectangle** (Insert -> Icons -> Rectangle)
- Insert a **Warning** icon (Insert -> Icons -> Warning)
- Add one **Label** (Insert -> Label)
- Insert two **Buttons** (Insert -> Button)
- Next, make a Group by using all the Powerapps icons and control and rename it to "**DisplayConfirmationGroup**" as shown below.

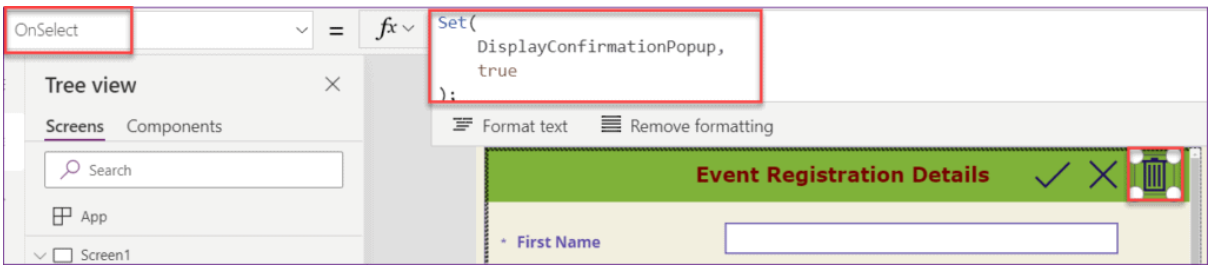


- Now, click on the **Trash** icon and set its **OnSelect** property as:

```
OnSelect = Set(
    DisplayConfirmationPopup,
    true
);
```

Where,

DisplayConfirmationPopup = Variable name which value is true

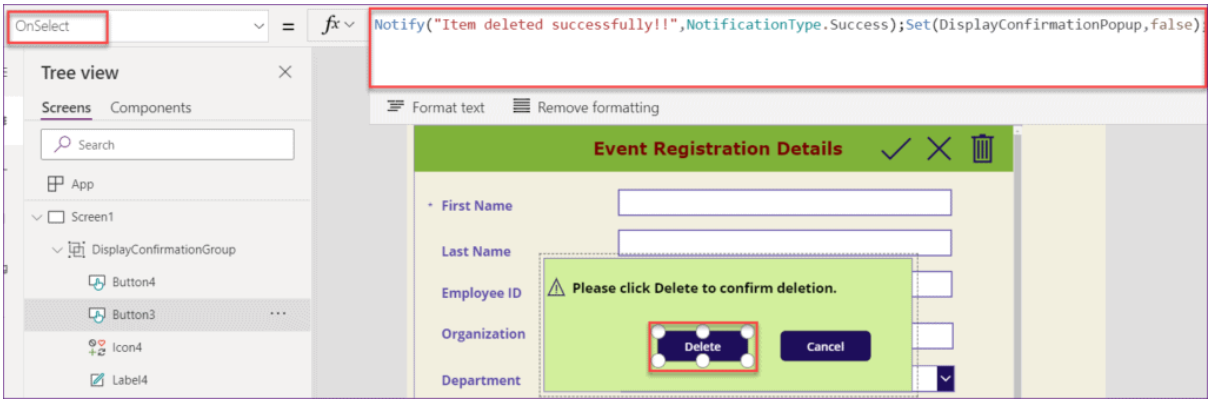


- Next, click on the **Delete** button from the confirmation popup box and set its **OnSelect** property as:

```
OnSelect = Notify("Item deleted
successfully!!",NotificationType.Success);Set(DisplayConfirmationP
opup,false);
```

Where,

- **Notify** = PowerApps Notify function helps to display a notification message to the user
- **NotificationType.Success** = This notification argument specifies the message is a success notification message.
- **DisplayConfirmationPopup** = Variable name that you have created in the Trash icon's **OnSelect** property

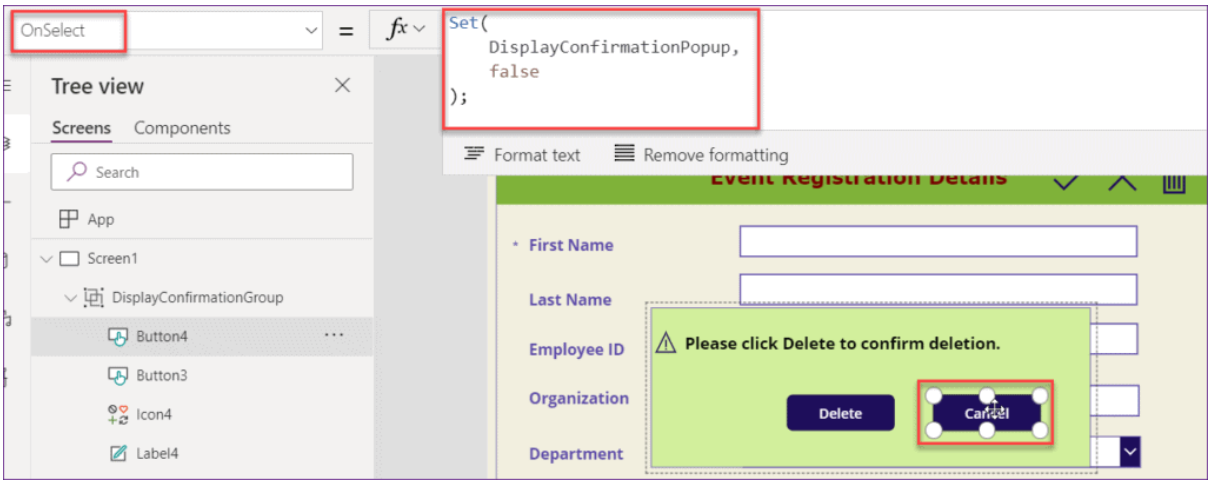


This above code specifies, When a user will click on the **Delete** button, then the confirmation dialog box will disappear and a successful notification will appear on the Powerapps screen.

- Similarly, Select the **Cancel** button and set its **OnSelect** property as:

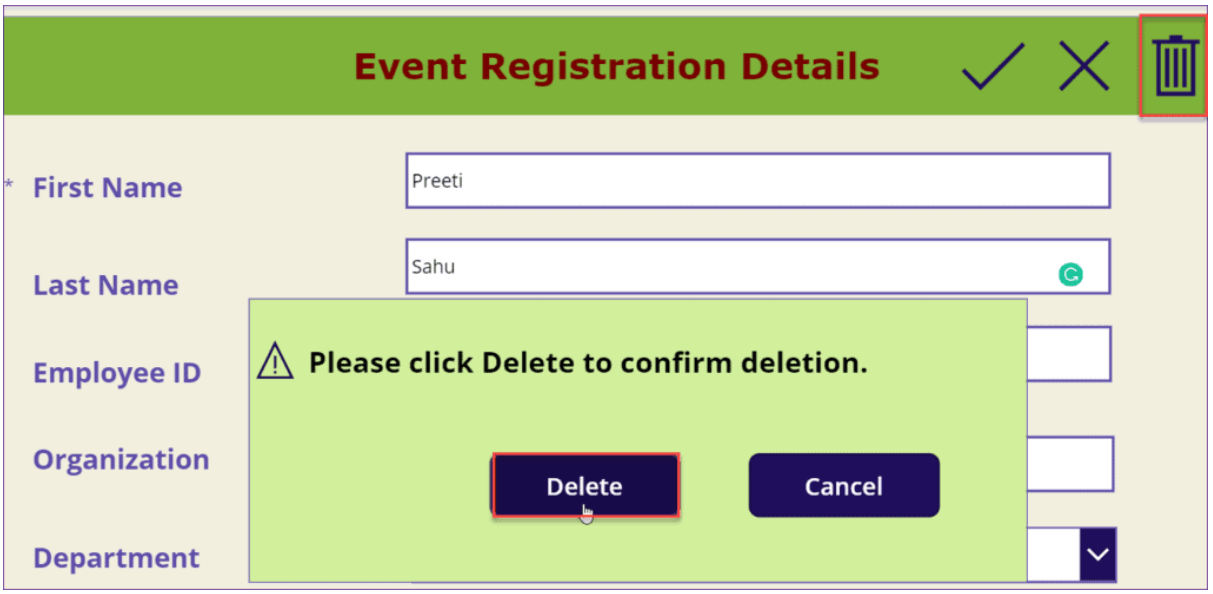
```
OnSelect = Set(
    DisplayConfirmationPopup,
    false
```

```
);
```

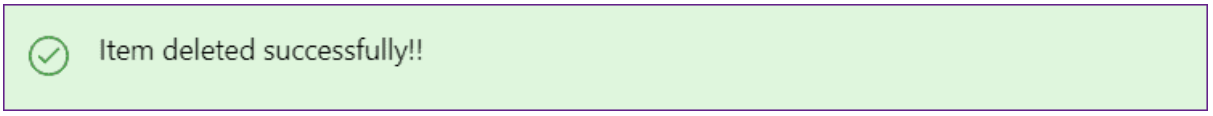


This above code specifies, When a user will click on the Cancel button, then this confirmation dialog box will disappear.

- Preview (F5) the app. Click on the **Trash** icon from the Powerapps form and then **Delete** it.



Then a Delete confirmation message will appear on the Powerapps screen as shown below:



Popup on Button click in PowerApps

Powerapps Popup on Button Click means activity will happen when a user will click on the **Button**. Also, I have shown you in the above scenario that how you can use a Button in the **Powerapps Popup message box**.

For better understanding, below, I will take another simple scenario to use the Popup on Powerapps Button click. Follow these below steps:

- Create a New Powerapps Blank screen and set its **OnVisible** property as:

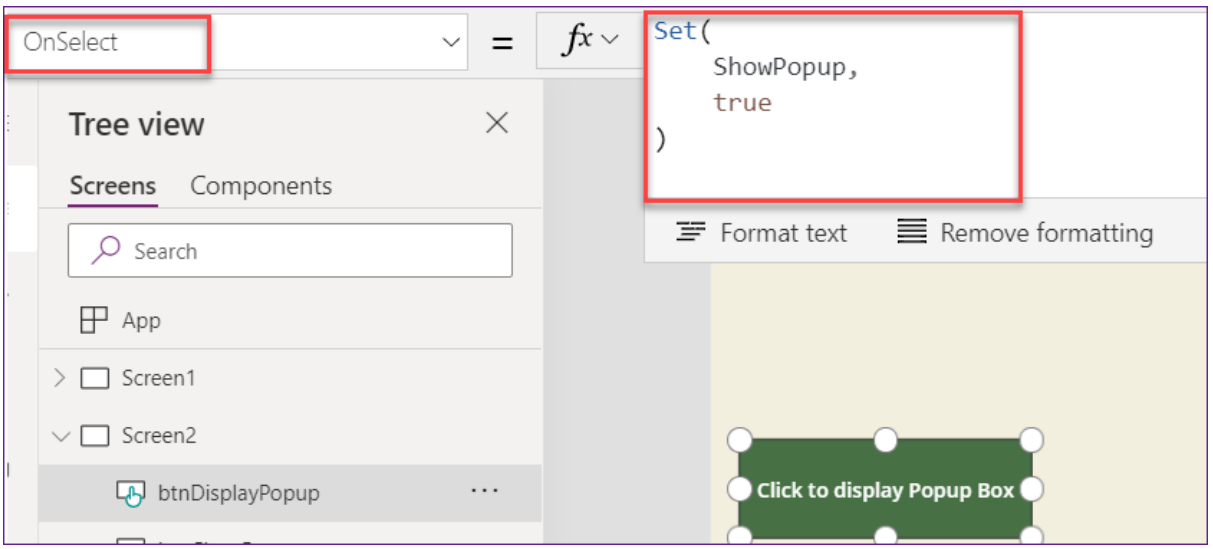
```
OnVisible = Set(ShowPopup,false)
```

Where,

ShowPopup = Variable name and you need to set its value as false. This means the pop up does not appear any event on the screen.

- On the Powerapps screen, Insert a **Button** and set its **OnSelect** property as:

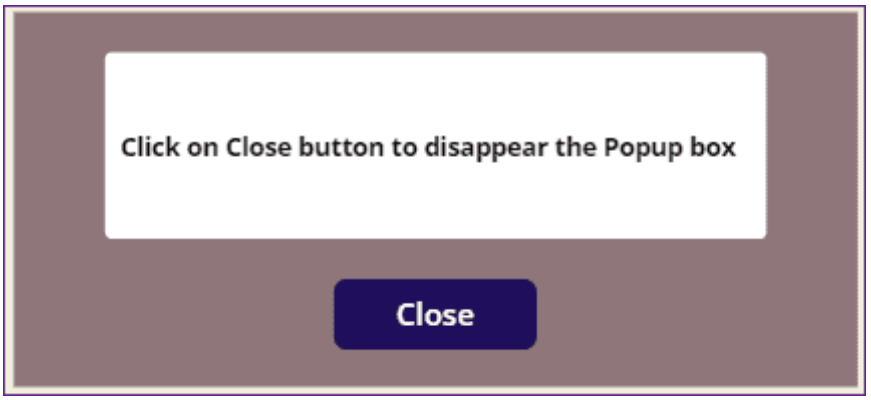
```
OnSelect = Set(ShowPopup,true)
```



In this above code, On the Onclick event of a button you are setting the ShowPopup variable as true.

- Now insert these below Powerapps icons and input controls as:
- Insert a **Rectangle** icon: Set its **Visible** property as "**ShowPopup**".
- Insert a **Text** input: Set its **Default** property as "**Click on Close button to disappear the Popup box**". Also, set its **Visible** property as "**ShowPopup**".
- Add one **Button**: Provide the **Text** property to **Close** and set its **Visible** property as "**ShowPopup**".

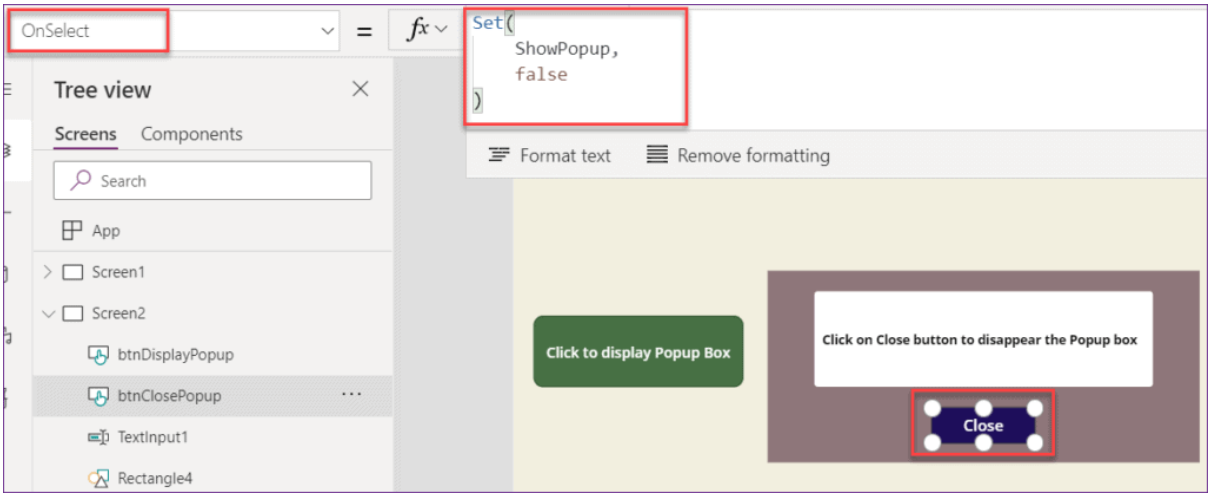
Refer the below screenshot.



- Next, Select the **Close** button and set its **OnSelect** property as:

```
OnSelect = Set(
    ShowPopup,
    false
)
```

As we will close the Popup dialog box, so I set the variable (ShowPopup) to false.



- Now, preview (F5) the app and click on the Button (**Click to display Popup Box**). Now you can see a Popup dialog box. Once you will click on the **Close** button, then the dialogue box will disappear.

PowerApps Notify

PowerApps Notify function is a type of function that helps to show a notification message to the user. A notification message means, it can be either a Success notification message, Informational notification message, Warning message, or an Error message.