Getting Started with PowerApps

Lab Time: 60 minutes

Lab Folder: C:\Student\Modules\04_PowerApps\Lab

Lab Overview: In this lab, you will build a device ordering application using the web-based version of PowerApps Studio which will run in most browsers including Microsoft Edge, Internet Explorer 9+, Google Chrome, Firefox or Safari. You may, however, choose to use the Windows Store version of the PowerApps Studio, which you can install from http://aka.ms/PowerAppsWin. This requires Windows 8.1 or Windows 10. The web and store versions of the Studio offer the same capabilities.

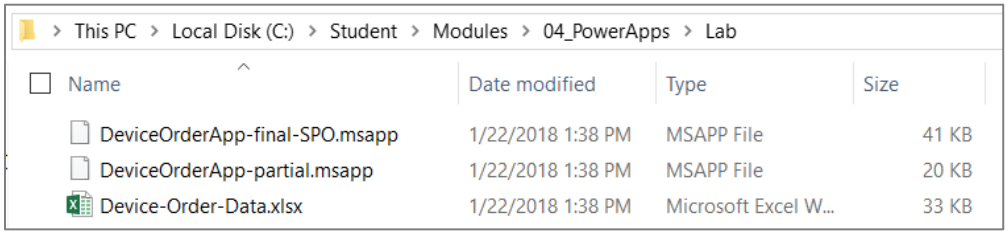
Exercise 1: Create the app in PowerApps

In this exercise, you will use the PowerApps Studio to create an app for selecting and comparing devices. You will then add the approval request process to this app, leveraging a SharePoint list in Exercise 2. You will continue this project in the next lab on Microsoft Flow where you will add an approval workflow.

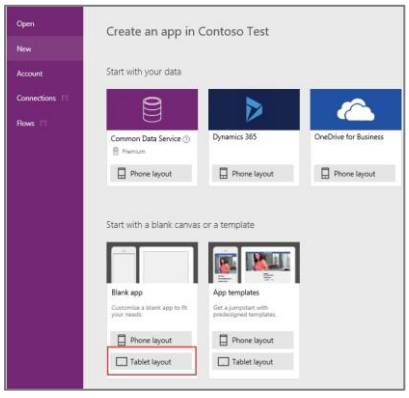
1. Examines the student files associated with this lab.
   a) Using Windows Explorer, examine the folder at the following path.

   ```
   C:\Student\Modules\04_PowerApps\Lab
   ```

   b) You should see the set of files show in the following screenshot.



2. Here's a quick explanation to tell you the purpose of each of these files:
   a) **Device-Order-Data.xlsx:** This is the sample Excel workbook file that includes two tables with devices and manufacturers.
   b) **DeviceOrderApp-partial.msapp**: This is a partially built device ordering app. If you would like to begin Exercise 2 in this lab to connect the app to a SharePoint list and have not completed all the steps in Exercise 1, you may use this partially created app.
   c) **DeviceOrderApp-final-SPO.msapp**: This is the completed app that includes all the steps from Exercise 1 and Exercise 2. Since this lab needs a connection to a SharePoint list, you will need to add the connection after opening the app in your account. You may use this app and jump ahead to the Flow lab.
3. Sign into PowerApps
   a) Navigate to http://web.powerapps.com and sign-in with your trial tenant account.
4. Start a new app
   a) Using PowerApps Studio, create a new Blank app (with a tablet layout).



b) In the Home tab see the **"Make apps like these"** section.

c) Click the **Start from blank** option, click the tablet/desktop icon, and select **Make this app**.

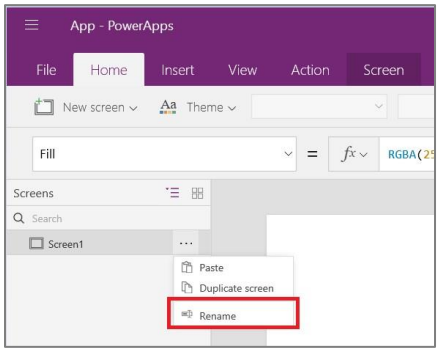
d) If prompted, click **Next** to walk through the getting started tour, and familiarize yourself with PowerApps Studio.

5. Rename **Screen1** to **MainScreen**.

a) Select the screen by clicking the **Screen1** tile in the left navigation bar.

b) Click "…" next to Screen1 and select the **Rename** option

c) Change the name to **MainScreen**



Note: You can also rename the screen by clicking on the screen name in the right pane and selecting the edit (pencil) icon.

6. Add a header containing the app name and logged in user's name.

a) Create a header across the top of your App

b) Select the **Insert** tab in the ribbon, and select **Label** to add a text label.

c) With your new label selected:

d) Rename the label from Label1 to **HeaderLabel.**

i) To do this, click … next to the Label1 control in the tree view and select **Rename**.

e) Select **Text** from the property dropdown list and enter **"Device Ordering App"** in the formula bar.

i) You can also type directly in the label control.

f) Resize the label such that the width is the width of the screen and the height is a reasonable size for a header.

g) You can set a specific value for the height, e.g. 80 pixels, by using the Properties pane on the right of the screen.

h) Select the **Home** tab and set the **Align** option in the ribbon to **Align center**.

i) Set the label fill color to blue and the font color to white. In the **Home** menu, use the following command buttons to set the colors. Or you can also use the Properties pane on the right

j) Similarly, set the **font size** to 24 from the Home ribbon.

i) You can also achieve this by modifying the **Size** property of this label control.

7. In the header, display the logged in user's name.

a) Add a label to display **"Hello [User's Name]"** where **[User's Name]** Is the full name of the logged in user.

b) Insert a label and drag it to the top right corner. Center it vertically to be in line with the header text. You can use the purple alignment markers.

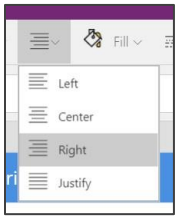
c) Change the value of the Text field to:

```
"Hello, " & User().FullName
```

d) The formula you add should match the one shown in the following screenshot.

Note: All functions in PowerApps are case sensitive. As you start typing "User" you will see a dropdown of available choices. It's a good idea to pick from the autocomplete options. You'll also notice help text at the top showing the required parameters, in this case it requires no input parameters.

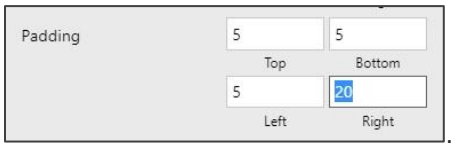
e) You can right-justify the text in the label by clicking the **Home** tab on the ribbon and using the text alignment button (not the same as the **Align** control) as shown in the following screenshot.



f) If necessary, widen the label so the text doesn't wrap.

g) To add some padding between the text and the edge of the screen, change the **PaddingRight** property from 5 to 20.

i) You can do this quickly using the Properties pane on the right side



The **User()** function in PowerApps allows you to retrieve the Email, Full Name and Picture for the currently logged in user. App users will always be logged in with their business or school account (Azure Active Directory (AAD) credentials), so this information will always be available for any PowerApps app.

8. Save the app.

a) Click on the **File** option in the ribbon menu and go to the **App settings** page.

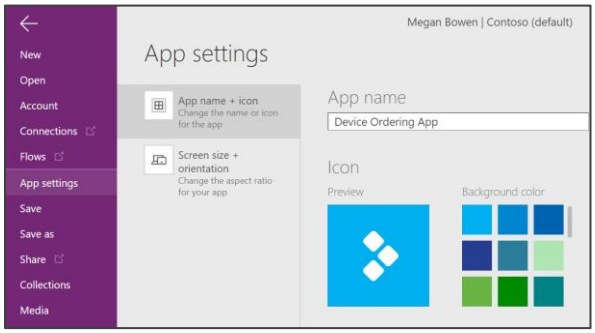
b) Give your app a name, such as "Device Ordering App"

c) Customize the app icon – choose a background color and symbol.

d) Select **Screen size + orientation** to view the available screen orientation and aspect ratio settings.

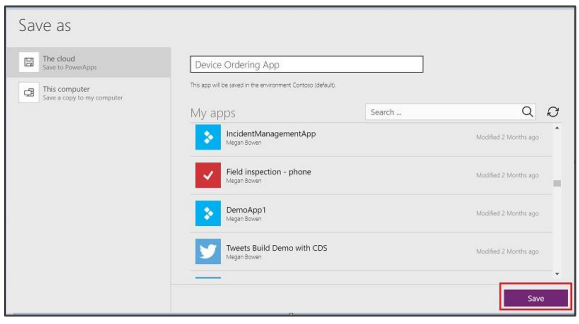
i) For this app we'll leave it at the default setting of Landscape with 16:9 aspect ratio.

e) Once you've named your app, select the **Save** option on the left pane.



f) Make sure the option to save it to **The cloud** is selected, so it's available instantly to run on any device.

g) Click **Save** in the bottom right.

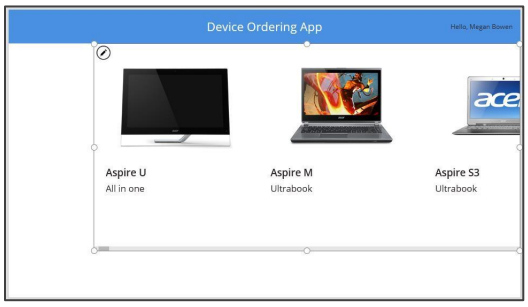


Tip: In PowerApps when you save a version of your app, the changes are only visible to the app maker. You must explicitly publish the app for all app users to get an update. For more details on saving, publishing and sharing apps, see:

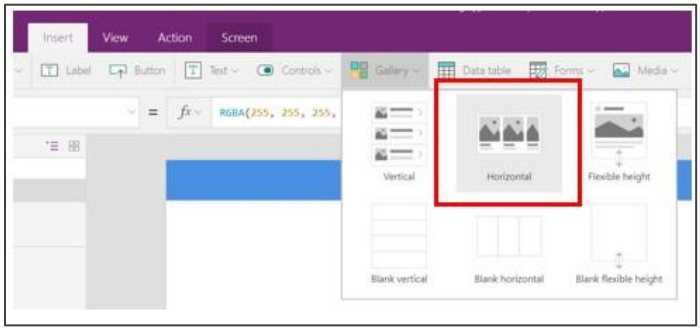
https://powerapps.microsoft.com/en-us/tutorials/save-publish-app/
https://powerapps.microsoft.com/en-us/tutorials/share-app/
https://powerapps.microsoft.com/en-us/blog/saveandpublish/

Add a Device Gallery

Over the next few steps, you'll add a gallery of all the available devices making it easy for users to browse the list and get a quick overview of the devices available. The gallery will look like this image below:



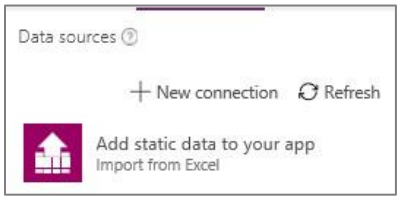
9. Create a gallery that will show all available devices
   a) Select the **Insert** tab in the ribbon
   b) Select **Gallery** to see a list of gallery templates. iii. Select the **Horizontal** gallery.



This will add a gallery called **Gallery1** onto the screen. Notice the control tree view on the left displays this gallery with three controls within it – two labels and an image. A Data pane will popDevice up on the right.

10. Select the data source for the gallery:
    a) In the Data pane on the right, click the **Data source** dropdown and select **+ Add a data source**.

    b) Choose the "Add static data to your app (Import from Excel)" option



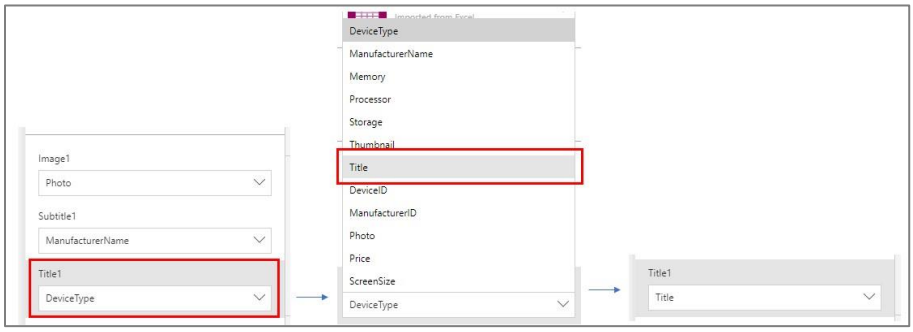
c) In the File Open dialogue, browse to the location to select **Device-Order-Data.xlsx**.

```
C:\Student\Modules\04_PowerApps\Lab\Device-Order-Data.xlsx
```

d) Select both tables, **Devices** and **Manufacturers**, and click the **Connect** button.

Note: In this lab, you will work with tables imported from a static data file and embedded as resources in the app. If you were building a real solution, the same tables would likely be stored in the cloud, such as in a SharePoint list or a table in SQL Server.

11. The first table, **Devices**, will be picked as the source for this gallery. Notice the gallery will update to show devices.
12. Update the field mappings
    a) The system by default will bind each control within the gallery to some data field in the table.
    b) Click on the first text label in the gallery, just below the image. This control is named **Title1**. Use the dropdown control in the data pane to change the binding form DeviceType to **Title**, which is the name of the device.



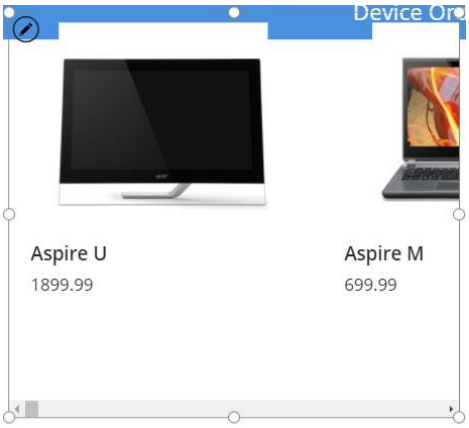
    c) Notice that the gallery UI updates as soon as you do this to display the device name below the image
    d) Also notice that the formula bar shows that **Text** property for this **Title1** control is set to **`ThisItem.Title`** which provides another way to change the binding by directly typing into the formula bar.

e) Notice that **Image1**, which is the image control, is correctly bound to the **Photo** field.
   i) If for some reason it does not show the image, change it to bind to the Photo field.

f) Similarly, change the binding for the **Subtitle1** label from ManufacturerName to **Price**.

g) Close the data pane on the right. To get back to the data pane, select the gallery control using the tree view on the left and in the Properties pane on the right click either Data or Layout.

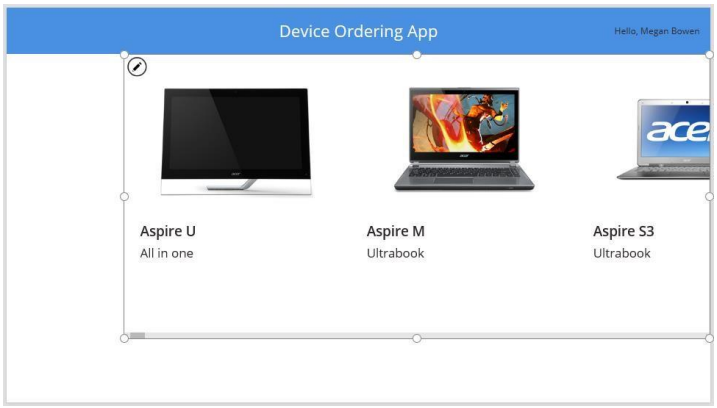
h) Your gallery should look like this:



13. Position and resize the gallery

a) **Position the gallery** to the middle and right of the screen just below the header label, such that there is space on the left to add a vertical gallery and space on the bottom for action buttons.

b) Use the drag handles to resize the gallery so that it covers approximately 80% of the width of the screen. See image below.



14. **Rename** the gallery to **DeviceGallery**. There are two ways to rename a control:

a) in the **tree view on the left**, click the three dots (…) next to the control and select **Rename**, or

b) in the **right pane**, click the **edit icon** (pencil) next to the name of the control.

Galleries are an essential aspect of UI construction because they provide a powerful way to visualize tabular data in PowerApps. It's important to become familiar with customizing a gallery. Key components of a gallery are: the gallery control, the template cell, and controls within the template cell.

To select the **entire gallery** – click on the gallery control in the tree view on the left pane, or click on the second or third cell of the gallery. Clicking any cell that is not the first cell of the gallery will select the entire gallery. Now you can configure properties that apply to the entire gallery, such as the Items property which is the data source, the gallery fill color, borders, etc.

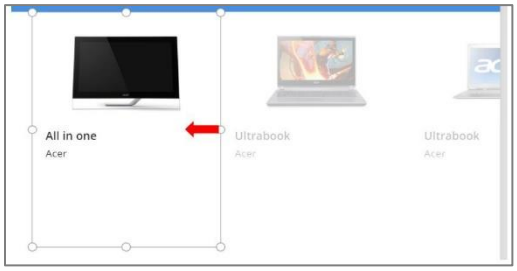
To customize how each item is displayed in the gallery, you will customize the **template cell**. Select the template by clicking in the first cell of the gallery or clicking on the pencil icon in the top left corner when the entire gallery is selected.

15. Arrange the device gallery

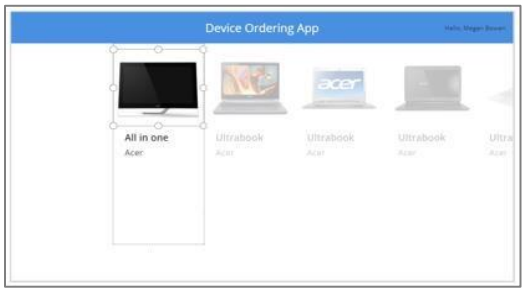
a) Select the **DeviceGallery** control and click the **Edit (pencil) icon** in the top left of the gallery to edit the template cell.

b) Using the right drag control, **resize the first box to be narrower**. Notice that all the items get narrower and more devices are visible on the screen.

c) Narrow the image as well by **clicking on the image control and resizing it using the drag handles**.



Make sure the width of the image control is positioned within the template.



16. Format the **Price** field

a) Show the price in the format of **$##,###.00**.

b) Select the second label, named **Subtitle1**. ii. To add the **$**, use the text format expression:

```
Text(ThisItem.Price,"$##,###.00")
```

c) Here is what your formula bar should look like.

d) The currency formatting should now appear in the app.

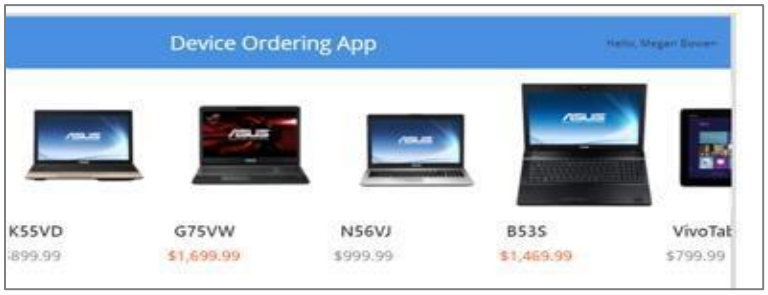
Note: After you enter the above value in the formula bar, notice that it will automatically resolve to including your locale, e.g. [$-en-US]. If you see an error here, it might be because your locale is not yet supported, in which case as a workaround, you can manually change it to [$-en-US].

17. Format the **Price** field to show devices above $1,000 in a different color
    a) Let's say we want to make it easy to spot devices that cost more than $1,000, by displaying the price in red.
    b) To do this, select the label in the template cell that displays the price
    c) Set the **Color** property using the following expression:

```
If(ThisItem.Price>1000,OrangeRed,Gray)
```

    d) The formula you add should match the following screenshot.

    e) The app should now show larger values in a red font.

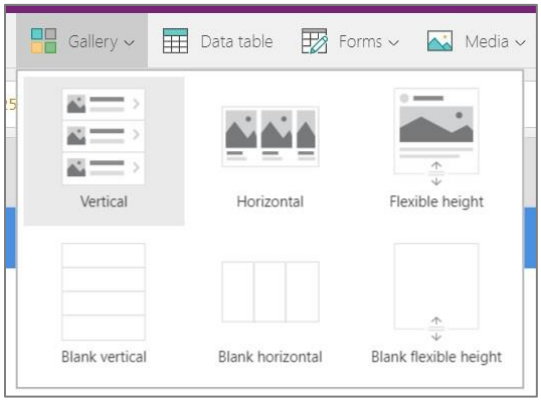


Note: As you're typing this formula notice that the autosuggest shows a choice of matching colors. PowerApps comes with a set of standard colors that you can reference in any property that accepts a color value. You can also set specific RGB values. For a full list of Color functions and colors, see https://powerapps.microsoft.com/en-us/tutorials/function-colors/

Add a gallery to display Manufactures

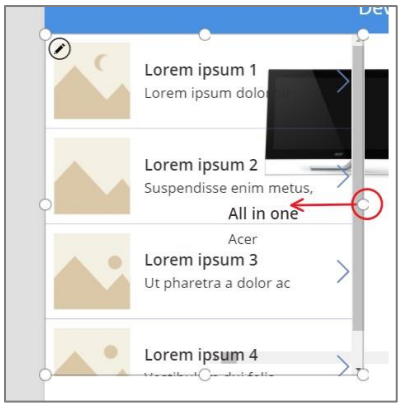
Over the next few steps, you'll add a second gallery that will list the various device manufactures. This gallery will later be used as a filter for the device gallery created later in this lab.

18. Create a second gallery that will show the list of manufactures of all the devices. This should be a single column, vertical gallery down the left side of the screen, with each cell displaying the manufacturer's logo image.
    a) Insert a vertical gallery on the left.
    b) Do this by clicking the **Insert** tab on the ribbon and opening the **Gallery**, then select **Vertical.**

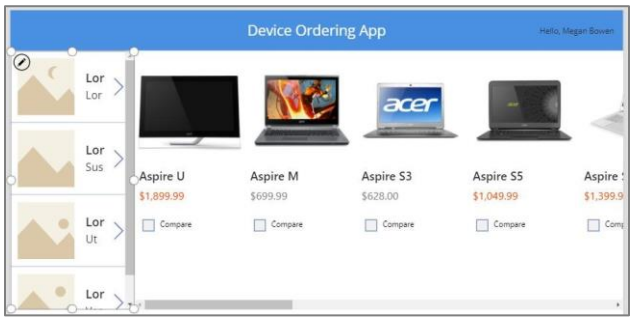


    c) **Close the data pane,** we will come back to it.
    d) Rename the gallery to **ManufacturerGallery**.

19. Reposition this new gallery.
    a) Move it so that it's left aligned with the left edge of the screen and top aligned with the top of the device gallery.



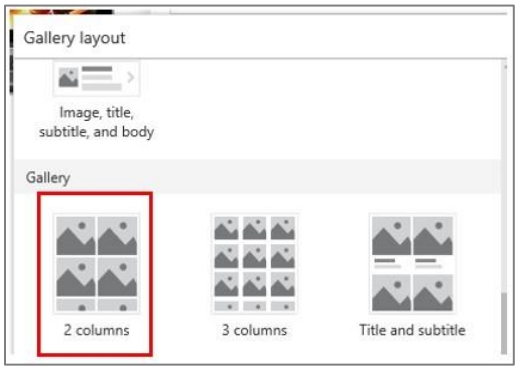
    b) Use the right middle grab handle to make the gallery narrower, such that it's touching the left edge of the device gallery.
    c) Use the bottom grab handle to align the bottom of the gallery with the bottom of the device gallery.



    d) Change the gallery layout
    e) With the gallery selected, in the Properties tab on the right, select **Layout** or **Data**.

    f) This will open a Data pane that allows you to configure the layout template and data source for the gallery.
    g) In this **Data** pane, select the **Layout** dropdown.

h) Scroll down and select the layout called **"2 columns"** with just images and no text.

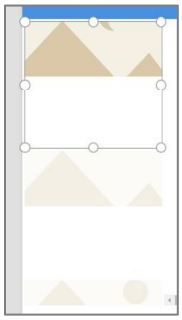


i) Close the data pane.

j) Click the **Properties** tab in the right pane and change the **Wrap count** from 2 to 1.

i) This will change it to a single column gallery.

k) Select the **image control** within the gallery and **reduce its height** by dragging the middle bottom drag control upwards. The image size will reduce whereas the template size will still be expanded.



l) Next, select the **template cell** by clicking the white space just below the image in the first cell. You can also select the template cell by selecting the entire gallery and clicking the pencil icon in the top left.

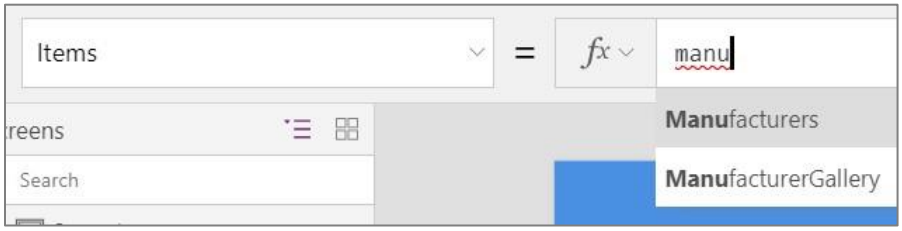
m) **Reduce the height of the template cell** to match the image. We essentially want the image to occupy the entire cell.

n) Connect the gallery to the Manufacturers table

o) For the devices gallery, you connected it to the data source using the Data tab in the right pane.
   i) You can also connect to data via the formula bar

p) Select the **ManufacturerGallery**. Make sure the whole gallery is selected and not just the first cell.

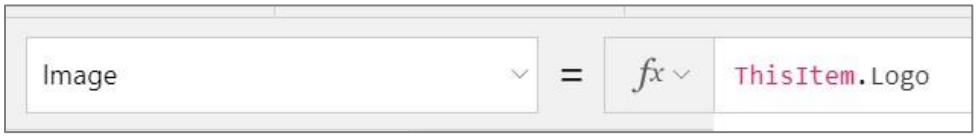
q) Select **Items** property and type Manufacturers in the formula bar.



20. Display manufacturer logos in this gallery

a) Notice that the gallery gets populated with images of buildings.
   i) This is because PowerApps picked a default binding which mapped to the HQ column in the table.

b) Select the image control in the first cell of the gallery and change the value of the **Image** property from `ThisItem.HQ` to **`ThisItem.Logo`**.



c) All the gallery items will now display logo images.

d) Highlight the selected item in the gallery:

e) Use the **TemplateFill** property of the manufacturer gallery to specify a highlight color for the selected item.

f) With the whole gallery selected (not just the first cell), set the **TemplateFill** property to:

```
If( ThisItem.IsSelected,ColorFade(HeaderLabel.Fill,75%) )
```

g) This is conditionally setting a Fill color if the cell is selected.

h) You could have set a specific color or RGB value, but we recommend using the **ColorFade** function so it matches the header label with a 75% fade. If you change the fill color of header label, this template fill color will automatically change.

i) Click different items in the gallery, notice the selected item is highlighted in a light blue color.

Configure manufacturer gallery as a filter for the device gallery

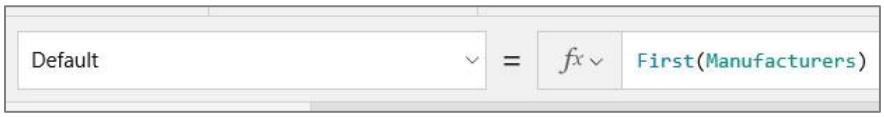
To help users be able to better browser the list of available devices, in this task configure the device gallery to only show devices that match the currently selected manufacturer.

21. Use the `Filter()` function to filter the device gallery based on the selected manufacturer
    a) Use `ManufacturerID` as the key
    b) Select the **DeviceGallery**. With the **Items** property selected, enter the following expression in the formula bar:

```
Filter(Devices, ManufacturerID = ManufacturerGallery.Selected.ManufacturerID)
```

    c) This will filter the device gallery to only display items that match the selected manufacturer based on ManufacturerID.
    d) Go ahead and select different items in the manufacturer gallery on the left and you'll notice the device gallery will update accordingly.

22. Set the default when the app loads to select the first manufacturer. We don't want a blank screen when the app first loads
    a) **Select the entire gallery** (by clicking **ManufacturerGallery** in the tree view on the left)
    b) Set the **Default** property of the gallery in the formula bar to: `First(Manufacturers)`

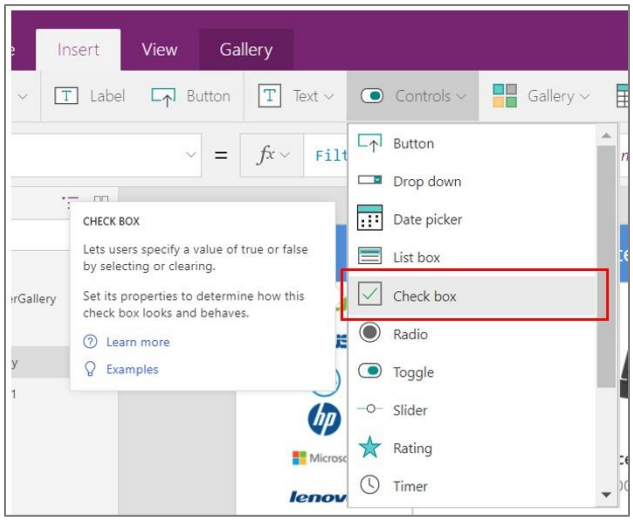


    c) This will set it to the first item in the table.

Enable selection of devices for comparison

Over the next few steps, you'll add checkboxes that allow users to select the devices they want to compare.

23. Add a checkbox that users can select to add a device to the comparison list
    a) Select the device gallery. Click the Pencil edit icon in the top left of the gallery to select the template cell.
    b) Make sure that only the first item in the gallery is selected (not the entire gallery).
    c) Add a check box by clicking **Insert** -> **Controls** -> **Checkbox**.



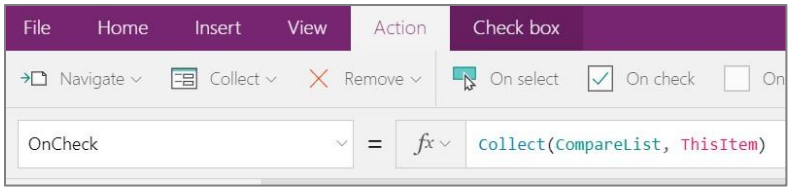
    d) Move the inserted checkbox below the label control for price
    e) Change the checkbox text to **"Compare"**. You can do this by setting the **Text** property or clicking within the text of the control and typing directly into the control.

    f) Confirm that there is a checkbox in each of the gallery items, similar to the image above.

If there is only one checkbox on the screen, it's likely that you placed it outside the context of the gallery. In that case, delete it, click again on the first template cell within the gallery and re-insert the control

24. When checked it should add the selected device to a Collection
    a) When a user selects a device to compare, we will add it to a collection called CompareList. You can think of this as an in-memory collection of devices that have been selected for comparison.
    b) Select the Checkbox control that you just inserted in the gallery, and click on the **Action** tab in the ribbon, click **OnCheck** and set the value in the formula bar to:

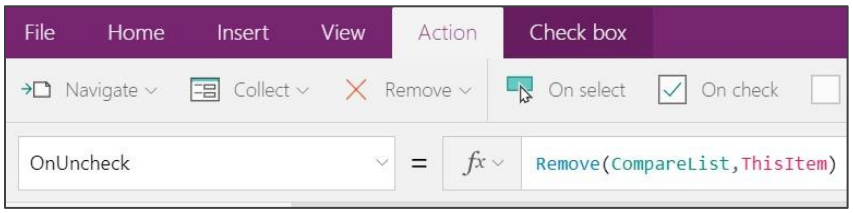
```
Collect(CompareList,ThisItem)
```



25. When unchecked it should remove the device from the collection
    a) Set **OnUncheck** to the following expressions.

```
Remove(CompareList,ThisItem)
```

    b) Your formula bar should match the following screenshot.

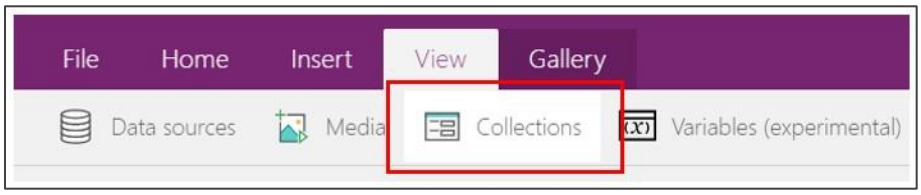


    c) Also, set the **Default** property of the checkbox to:

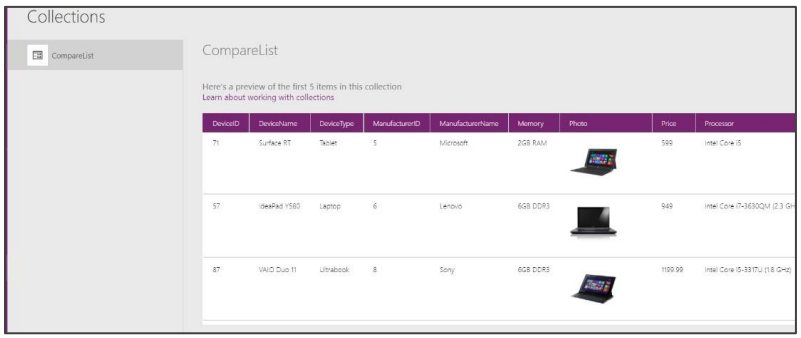
```
ThisItem in CompareList
```

The **Default** setting of the checkbox is a boolean true or false value that determines if the checkbox should be checked or not by default. Setting it to this formula will ensure that the checkbox is checked by default if the item has already been added to the collection, since the result will be true, i.e. this item *is* in CompareList.

26. Test that items have been added to the **CompareList** collection
    a) Click a few checkboxes
    b) Click on **View > Collections**



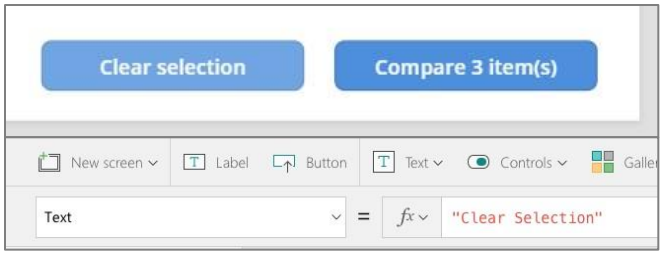
    c) This will show the first five items in the collection



Add buttons to the screen

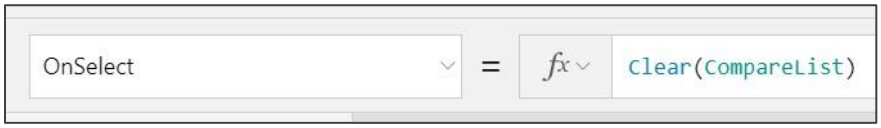
In this task, you will add two buttons to the screen – one to clear the selection and another to navigate to the second screen to submit the approval request.

27. Add the Clear selection button: this should clear all selected check boxes
    a) On the left tree view, click **MainScreen**, click on the **Insert** tab on the ribbon and select **Button** to add a button to the screen.
    b) Position the button just left of the bottom right corner (leave room for the compare button to the right).
    c) Set the button's **Text** property to: "Clear Selection"



    d) Set the **OnSelect** property for this button to:

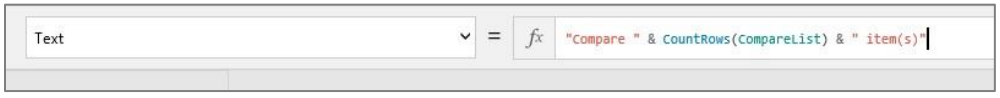
```
Clear(CompareList)
```



This will remove all of the items in the **CompareList** collection

28. Add the "Compare X item(s)" button, where X is the number of devices you have selected via check boxes.
    a) This button won't do anything yet, but will eventually take you to the comparison screen.
    b) On the left tree view, click on **MainScreen**, click **Insert** -> **Button** to add a button to the screen.
    c) Position the button in the bottom right corner.
    d) Set the button's **Text** property to:

```
"Compare " & CountRows(CompareList) & " item(s)"
```



    e) Resize the button so the text fits – use the grab handle to make it wider
    f) Set the Fill property of the button to: `HeaderLabel.Fill`.
        i) This will make sure the button color is the same as the header banner color
    g) To Gray out the button when no items are selected:
        i) Select the button, and set its **DisplayMode** property to:

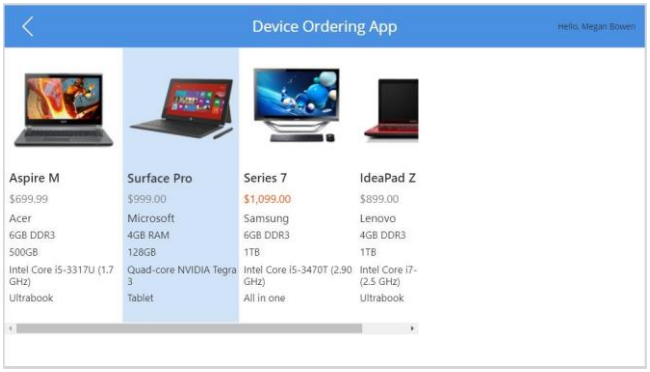
```
If(CountRows(CompareList) > 0, DisplayMode.Edit, DisplayMode.Disabled)
```

h) This will enable the button only if the number of items in **CompareList** is greater than zero.

i) You can test this by unselecting all machines – notice the button is grayed.

Create Second Screen for Device Comparison

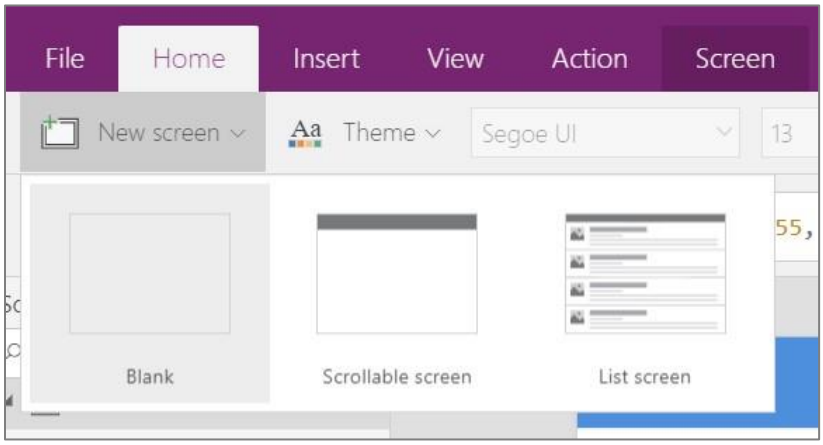
Over the next few steps, you'll create the second screen used for comparing the select devices. When you're done, it will look like this:



29. Add a second screen to your app

a) From the ribbon click **Home** -> **New Screen,** and choose **Blank**.

b) Rename the screen to **CompareScreen**.



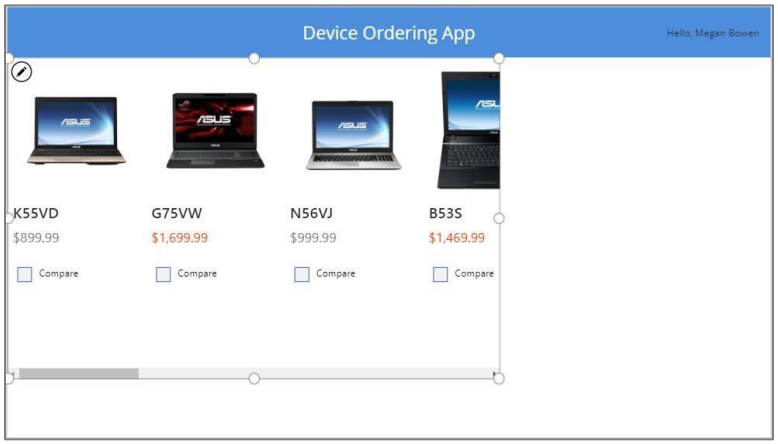
c) Give it the same header as the first screen (you can copy and paste)

d) You can copy/paste items from the MainScreen to the CompareScreen.

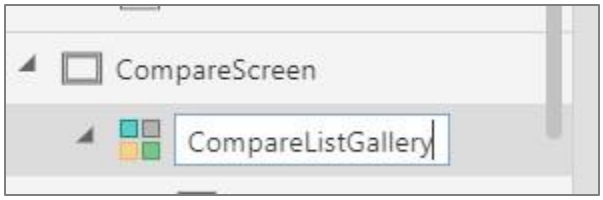
e) Click on **MainScreen** in the left navigation bar and keep the Ctrl key pressed down while multiselecting the blue header label and the "Hello, <user>" label. Once you've multi-selected the controls, type (Ctrl-C) to copy them to the clipboard.

f) Go the **CompareScreen** by clicking in the left navigation bar, and type (Ctrl-V) to paste the controls. You may need to move them up to align with the top of the screen.

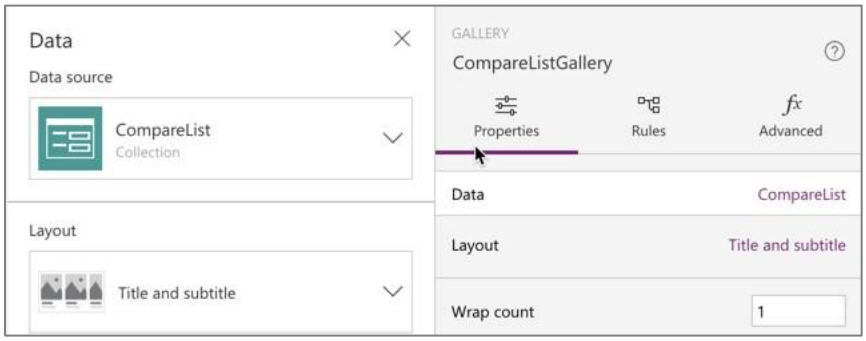
30. Add a gallery for viewing the devices selected for comparison
    a) This can also be a copy/paste from your main screen
        i) Copy the **DeviceGallery** from the MainScreen and paste a copy into to the newly created **CompareScreen**.
        ii) **Move the gallery** to the left edge of the screen. Align the top of the gallery to be just under the header banner. Use the right drag handle to reduce the width of the gallery and create space for a data entry form on the right of the screen. You will insert a Form control here and configure it in a subsequent Exercise.



    b) Rename this gallery to **CompareListGallery**.



    c) Change the data source to the Comparison Collection



    d) Place this gallery on the far left with some empty space left on the far right
    e) Delete the Compare checkbox from this gallery:
    f) In the second screen we're selecting a given item to get approved, so we don't need a Compare checkbox.
    g) Select the **Compare checkbox** on the left most template cell and delete it.
    h) Add some additional fields to the gallery to view more details about the device: Memory, CPU, Storage, etc
    i) A good way to do this is to copy paste an existing label. Select the first label in the gallery that is displaying the device name. Copy it (Ctrl-C) and paste it (Ctrl-V).
    j) Move the pasted label so that it's just below the price. Set the **Text** property to: `ThisItem.ManufacturerName`

k) Use the following ribbon button to change the font weight from **Semibold to Normal**.

l) Change the **Size** property from 20 to 18.

m) Copy and paste this label and move the new fourth label below the third label. Set its **Text** property to: `ThisItem.Memory`

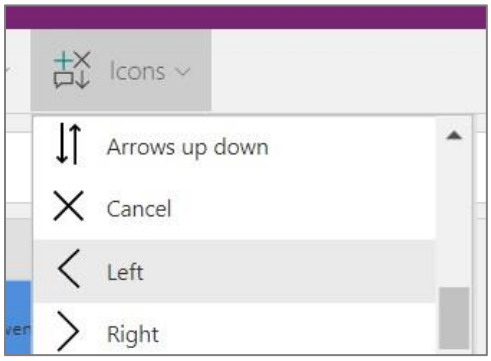
n) Change its **Size** property from 18 to 16.

o) You may optionally repeat this exercise and add text boxes to display additional device properties – Process, Storage, etc. Feel free to customize the labels by changing their Size, Color, Fill and FontWeight properties. In the interest of time, however, you may skip this step.

31. Add a “Back” button/icon to the header to navigate back to the first screen

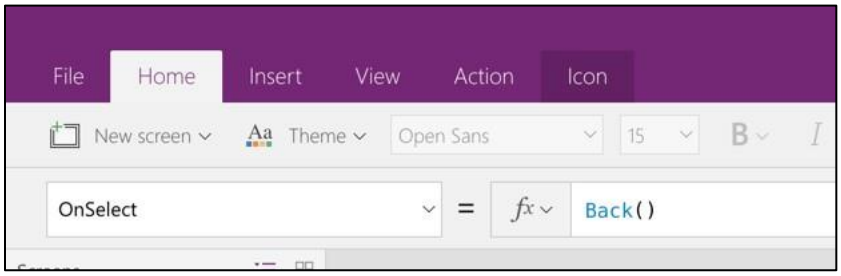
a) On the CompareScreen, click outside the gallery.

b) Go to **Insert**, then **Icons** and select the **Left** icon. Position it in the upper left corner of the screen.



c) With the arrow icon selected, change the **Color** property to **White**. You can change this in the formula bar or through the Properties pane on the right.

d) Set the **OnSelect** action for the icon to **`Back()`**. This will cause a navigation back to the previous screen.



32. Highlight the selected device.

a) Like the behavior in the manufacturer gallery on the first screen, use the **TemplateFill** property to highlight the selected item:

b) Select the gallery (**CompareListGallery**)

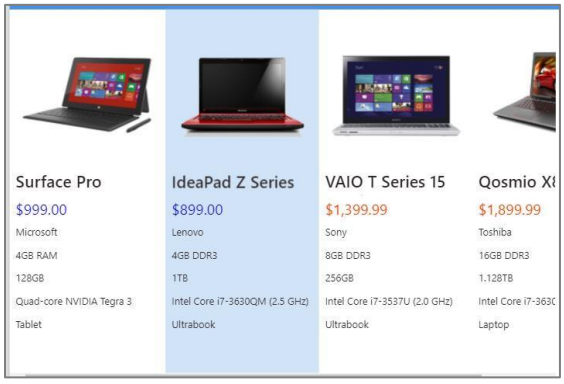
c) Set the **TemplateFill** property to:

```
If(ThisItem.IsSelected,ColorFade(HeaderLabel.Fill,75%))
```

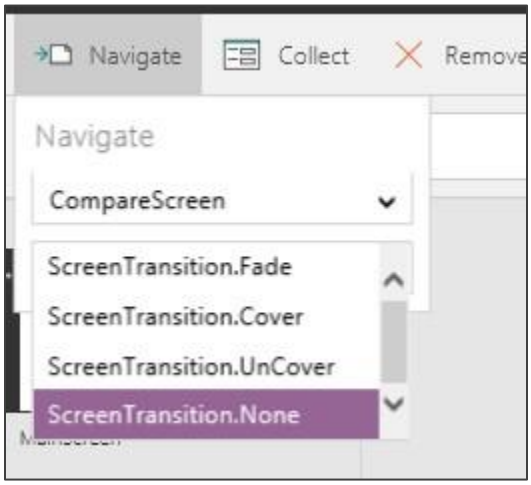
d) This is conditionally setting a Fill color if the cell is selected.

e) You could have set a specific color or RGB value, but we recommend using the ColorFade function so it matches the header label with a 75% fade. If you change the fill color of header label, this template fill color will automatically change.

f) Click different devices in the gallery, notice the selected item is highlighted in a light blue color.



33. Wire up the Compare button on the first screen to navigate to the second screen
    a) With the “Compare” button selected, click on the **Action** tab and select **Navigate**.
    b) Select the name of your second screen from the dropdown
    c) Select a transition type, you can select **ScreenTransition.None**.

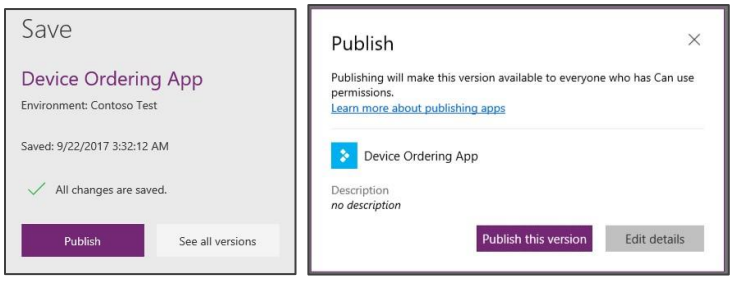


    d) Click the compare button and verify that it takes you to the second screen.

Save and test your App

This takes care of creating most of the app. Now let’s test it out.

34. First, make sure to save the app by clicking **File** -> **Save**
35. Use the “Play” button in PowerApps Studio to test out your app
    a) Test the filter by manufacturer functionality
    b) Select a few devices to compare
    c) Hit the compare button to go to the second screen
    d) Verify that the item you selected is highlighted with a border
    e) Hit the back icon and confirm you get back to the main screen
36. Test your app on a mobile device:
    a) Publish the app. Select **File** -> **Save** and click the **Publish** button. Click **Publish this version** on the confirmation prompt.



    b) You should see a confirmation.

If you haven't already installed the PowerApps mobile application, go to the app store on your device, search for "PowerApps" and install the PowerApps mobile application. This is often referred to as the **PowerApps Player** or the PowerApps container app within which all your published apps will run.

37. Run the PowerApps player app and sign in with the same business or school account credentials that you used to create the app.
    a) Look for the **Device Ordering App** in the app list and run it.
    b) Since this is a tablet app it will run in landscape mode. Although the controls will be a bit small on a mobile phone, you should be able to interact with it and test out the app functionality

[Optional] Share the app with a colleague

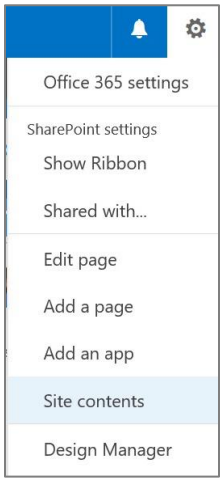
You may share the application with another user who is within the same organizational tenant as the user who created the app. So, if you had logged in as meganb@contoso.com, you may share the app with any other User or Security Group within the @contoso.com tenant. To share the app:

38. In the PowerApps Studio, click on **File** -> **Share**
    a) This will open another browser tab that displays the Share settings for the app
    b) Enter the name or email of the person with whom you would like to share the app
    c) You may provide this user or group either **Can use** or **Can edit** permissions
    d) If the **Notify users via email option** is checked, when you hit **Save**, the user or all users in the group will receive an email letting them know that the app has been shared with them, along with a link to open the app.

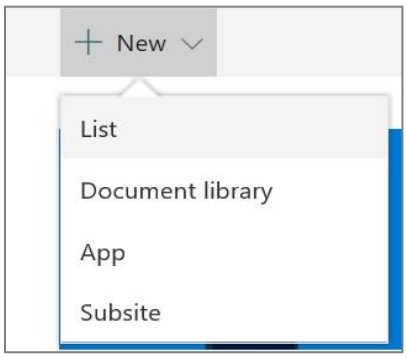
Exercise 2: Connect the app to a SharePoint list

In this exercise you'll take the app created in Exercise 1 and connect it to a SharePoint list, so you can track the orders and order approvals in a central location. In this first task, you'll create a new SharePoint List to use with the app you created in Exercise 1.

1. Navigate to the SharePoint site you would like to use for this exercise.
    a) This lab assumes you have access to a valid Office 365 subscription with permission to create SharePoint online lists.
    b) You can access SharePoint by logging in at http://portal.office.com and clicking the SharePoint icon.
    c) For getting started information on SharePoint, see here.
    d) The SharePoint list must be in the same tenant as the PowerApps application, use the same login credentials that you used to create and save the app.
2. Create a new SharePoint list.
    a) Click on the gear icon and click on **Site contents**



    b) Click on **New** and select List from the menu

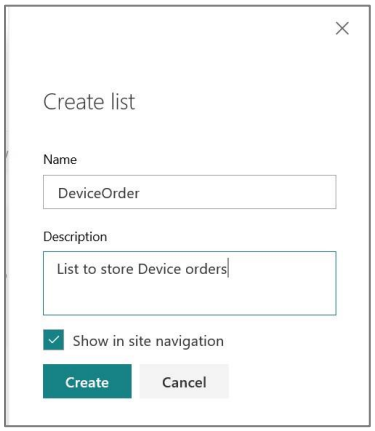


c) Create a new List

i) Enter the List name: **DeviceOrder**

ii) Enter a short description

iii) Click the **Create** button.



d) The list will have a Title column by default

e) In the next step you will click on the **+ sign** next to the Title column to add additional columns

f) Add these **additional columns** step by step to the new list:

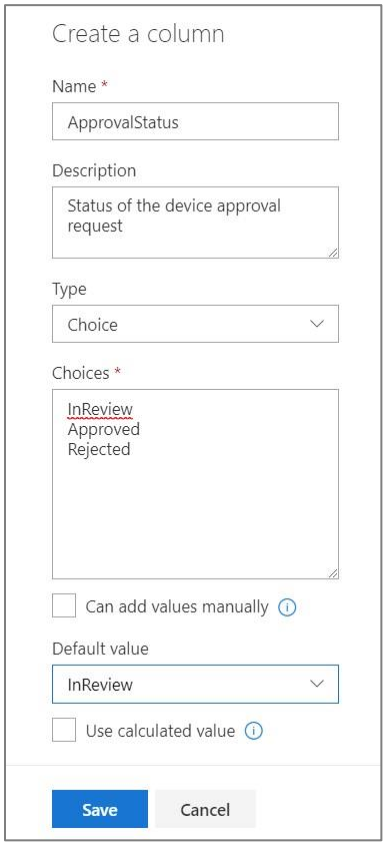
Column	**Type**
DeviceID	Number
Price	Currency
RequestedBy	Single line of text
Approver	Single line of text
ApprovalStatus	Choice*
Comments	Multiple lines of text

g) *For the **ApprovalStatus** column, select the Choice type and enter **InReview**, **Approved**, and **Rejected** as choices.

Note: In this lab, we're using a column of type Choice to specify the valid set of choices that will populate the dropdown in the PowerApps form UI. Alternate approaches:

h) Set the column type to "Single line of text" and store the list of options in the app

i) Create another SharePoint list, e.g. "ApprovalStatusChoices", that serves as a lookup table with the valid set of choices. The advantage of this approach is that you can bind to this list directly from a gallery in PowerApps and use the PowerApps UI to manage updates to this list – perform CRUD (create, remove, update, delete) operations on this list. When using the Choice type, all updates must be done from the SharePoint List Settings UI.



3. View your **list settings** and confirm the columns and types are accurate.

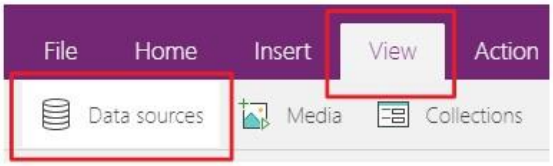
a) Click the gear icon in the top right and select List settings

b) The column settings for the DeviceOrder list should match the image below –

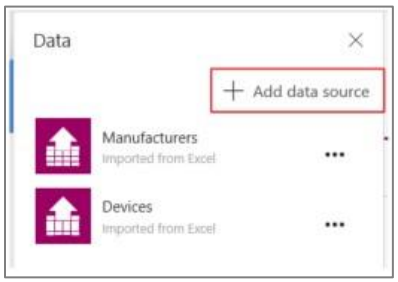
Add SharePoint as a data source and add an Edit form to your app

Now the SharePoint list is created, attach this SharePoint list to your PowerApps app so you can submit data and store it in the list.

4. Open the app you created in Exercise 1
5. Add SharePoint as a data source
   a) Click on the **View** tab and then click **Data sources** to display the Data sources property pane to the right of the canvas.

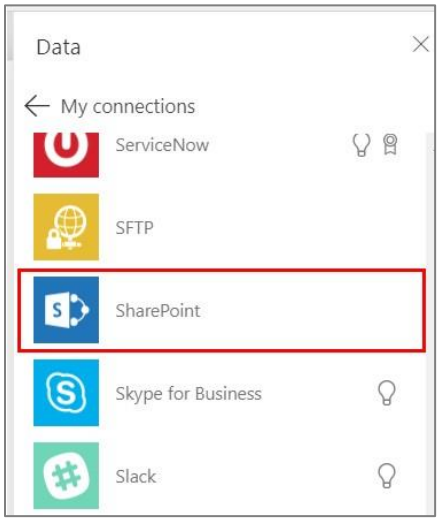


   b) Click + Add data source.

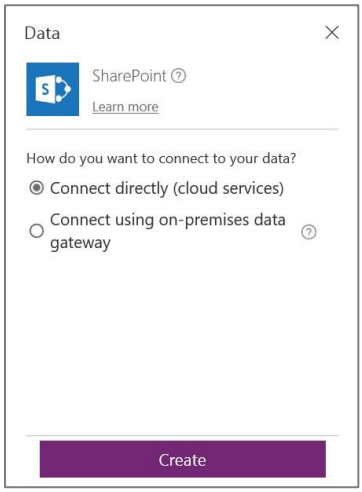


   c) If you see SharePoint in the list of existing connections, select it.
   d) If you don't see SharePoint listed, click **+ New connection**

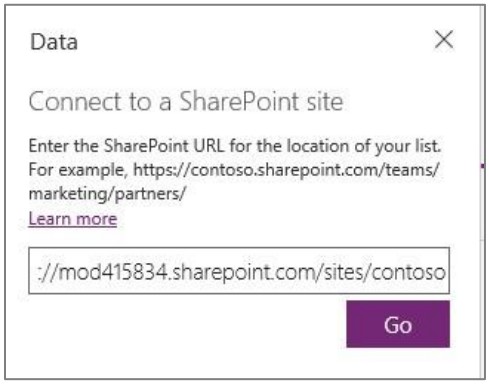
   e) Select the **SharePoint** connection.



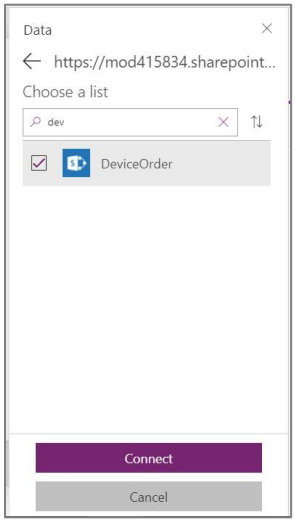
f) Select the Connect directly (cloud services) option and click Create.



g) Enter your **SharePoint site URL** where you created the **DeviceOrder** List and then click **Go**. **Important**: Enter only the site URL and not the full URL to the list. See example below.



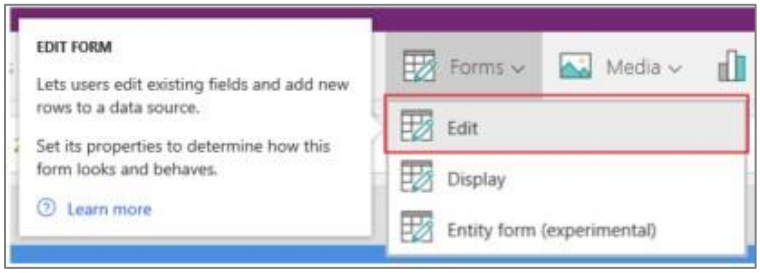
h) You will see all the lists on the site, search for the **DeviceOrder** list, select it and click **Connect**



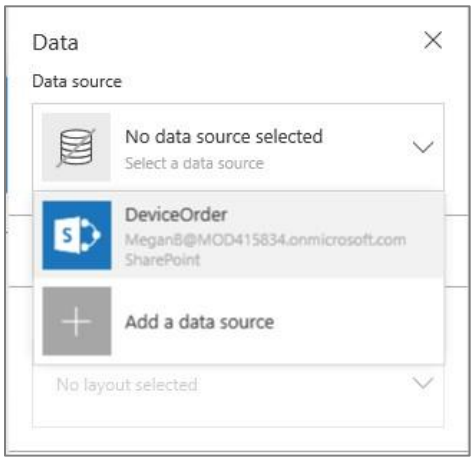
Add a form to your app and connect it to the SharePoint list

In this task, you will add an Edit Form to the app that is connected to the SharePoint list. This will let users input information for their device approval request.

6. On the second screen of your app, Insert an Edit Form and select DeviceOrder as the data source
   a) Click **Insert** in the ribbon and select **Forms** -> **Edit**.

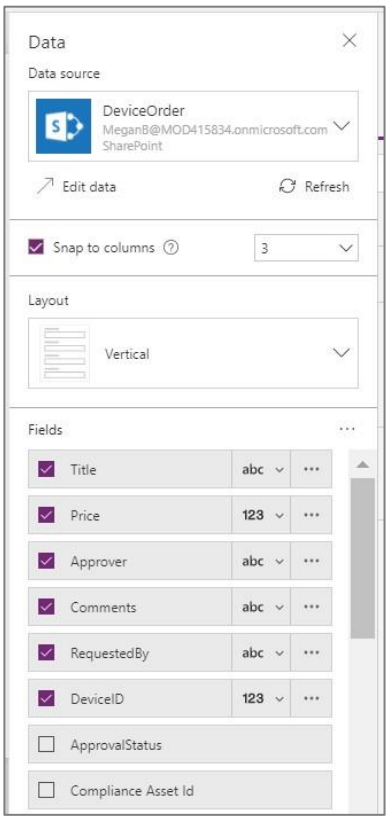


   b) Click the **Data source** dropdown in the Data pane on the right.
   c) Select the **DeviceOrder** list as the data source.



   d) By default, all the fields will be created on the form. Unselect the fields not required, and re-order the fields such that the set of selected fields matches the image below: Title, Price, Approver, Comments, RequestedBy, DeviceID. You can drag the fields up and down as needed.

Tip: Scroll to the bottom of the list of fields and start unselecting from the bottom up.



e) Hit the 'X' button in the top right to close the data pane.

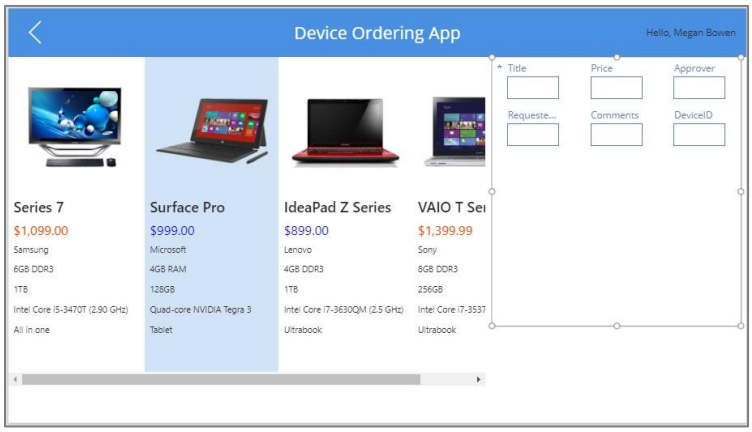
7. Move and resize the new Edit form to fit on the right side of the second screen, see image below:

a) Move and resize the form such that it is just below the header label, aligned with the right edge of the screen, and to the right of the device gallery.

b) Make sure there is enough space below the form to add a Submit button

c) **Note**: To select the entire form, use the tree control on the left and select the Form1 control. If you click within the form you may select a data card within the form and not the entire form.

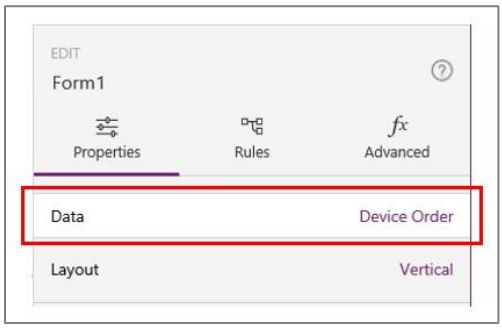
d) Don't worry about the formatting of each field within the form for now, we'll cover that in the next task.



Format and customize the edit Form

Now that you have a new Edit Form that is tied to the SharePoint list, let's go ahead and make some edits to the form.

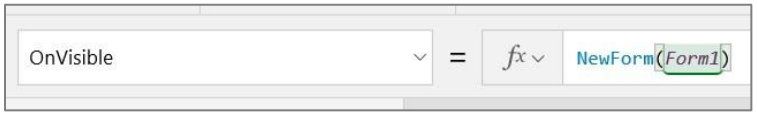
8. Change the form to a single column
   a) Click the **Data** link in the **Properties pane** to the right to open the Data pane.



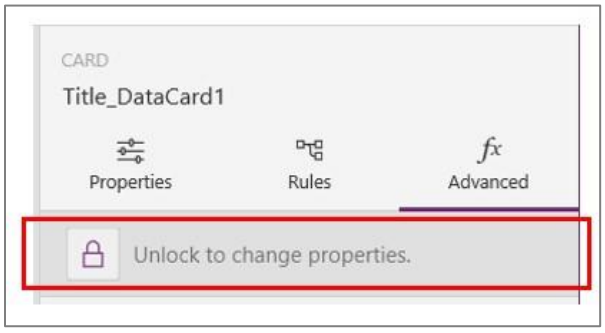
   b) Change the **Snap to columns** setting from 3 to 1.

9. Initialize the form by creating a new instance of the form each time the second screen is loaded
   a) Selected the screen – click **CompareScreen** in left tree view pane.

   b) In the **OnVisible** property of the screen, enter: `NewForm(Form1)`



10. Configure the Title field by setting the value of the Title field to include the manufacturer and device name of the selected device.
    a) Make it read-only. For example, if the user selects the Surface Pro device, the Title should be "Microsoft – Surface Pro". a. Set the default value for the Title field:
    b) Select the Title card. The control name is Title_DataCard1
    c) With the Title card selected, go to the Advanced pane on the right iii. Click the Unlock button so you can customize the card.



d) Click the **more options** button in the DATA section of the Advanced pane.

e) Set the **Default** property to:

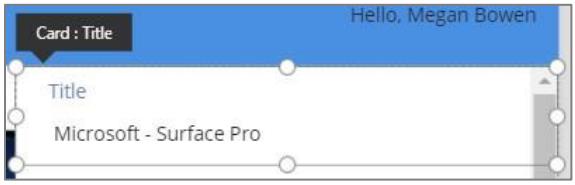
```
CompareListGallery.Selected.ManufacturerName & " - " & CompareListGallery.Selected.Title
```

f) Set the Title field to be Read-Only, i.e. DisplayMode = View

i) Click the **more options** button in the DESIGN section of the **Advanced pane** on the right. ii. Change the **DisplayMode** from Parent.DisplayMode to **DisplayMode.View**.

g) Verify the Title field is correct

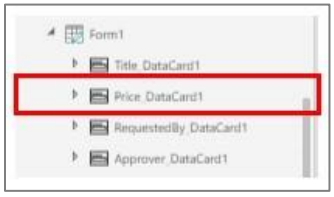
i) Select a device in the gallery ii. The Title field in the form should display the manufacturer and device name



h) Select different devices and notice the Title value changes

11. Configure the Price field – set it to the price of the selected device

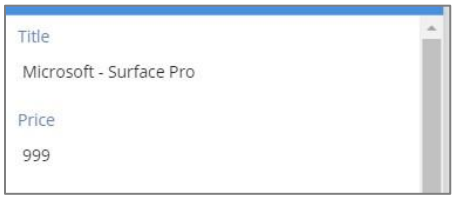
a) Select the data card for the Price field – **Price_Datacard1**



b) Go the Advanced pane and unlock the card.

c) Change the Default property to: `CompareListGallery.Selected.Price`

d) Change the DisplayMode property to: `DisplayMode.View`



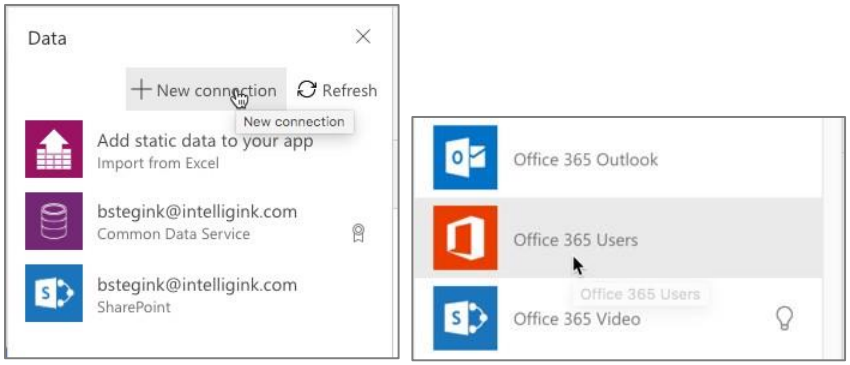
12. Configure the Approver Field

a) Set the default value for the Approver text input to be the email address of the logged in user's manager. **Note**: This step teaches you how to use the **Office 365 Users connection** to get the manager's email. You may treat this step as optional.

b) Connect to the **Office 365 Users** connection

i) Click **View** -> **Data sources**. In the right-hand Data pane, select **+Add data source**.

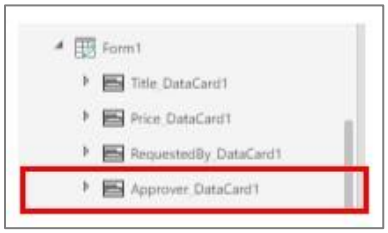
ii) If you see an **Office 365 Users** connection, click it to add it to the app. iii. If you don't see this connection in the list, click **+New connection**. iv. Pick the connection type - **Office 365 Users** and click **Create** to add it.



c) Close the Data pane.

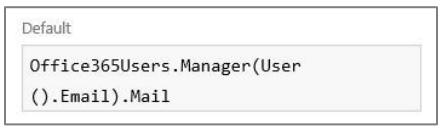
i) Set the value of the text input box to the current user's manager's email address.

ii) Select the Approver card – Approver_DataCard1.



iii) Go to the Advanced pane and Unlock the card.

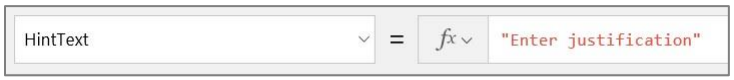
iv) Set the Default value to `Office365Users.Manager(User().Email).Mail`

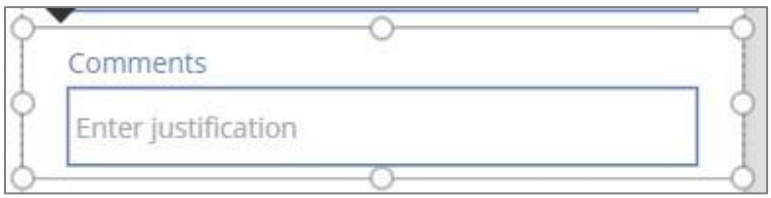


v) This calls the Office365Users.Manager function and passes the current logged in user's email as the input parameter.

Note: If you get an error, the current user may not have a manager setup in the system. In this case, you can hard code the default value to your email address – "myemail@domain.com" (include the quotes).

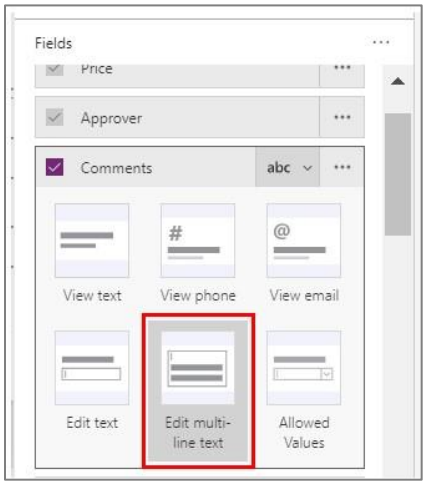
13. Modify the **Comments** Field
    a) Select the text input control for the Comments field
    b) Set the **HintText** property to: "Enter Justification"





    c) Change the input from single line to multi-line
        i) Select the entire form – **Form1** control in the left tree view
        ii) Click on **Data** in the **Properties pane** on the right to open the data pane iii. Select the chevron next to the Comments field to view a list of layout options

    d) Select **Edit multi-line text** to change it from a single line to multi-line input

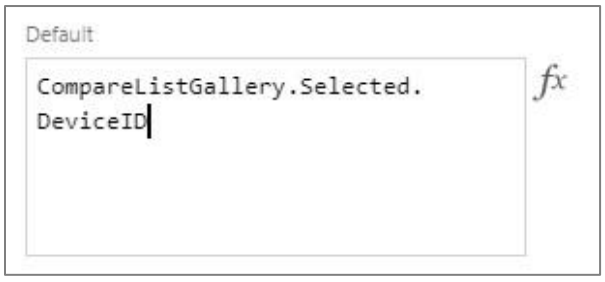


    e) The Comments field in the form should look like this:

14. Modify the **RequestedBy** Field
    a) Default to current user's email
        i) Select the RequestedBy card – **RequestedBy_DataCard1**
        ii) Go to the **Advanced pane** and **Unlock the card**.
        iii) Change the **Default** property to: `User().Email`

    b) Make it Read-only: Change the **DisplayMode** property for the card to: `DisplayMode.View`

15. Modify the **DeviceID** Field
    a) Store the template as part of the order.
        i) Set this card to hidden so it's not visible in the form, but stored as part of the form submit.
    b) Set the value to the DeviceID of the selected device
        i) Select the DeviceID card – **DeviceID_DataCard1**
        ii) Go to the **Advanced pane** and **Unlock the card**
        iii) Change the **Default** property to: `CompareListGallery.Selected.DeviceID.`



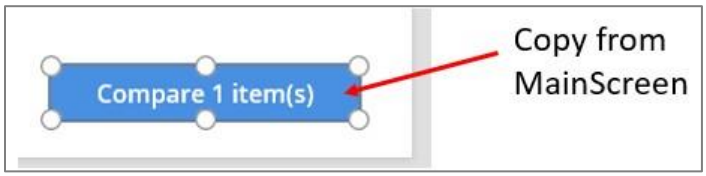
    c) Hide the card
        i) With the card selected, go to the **Properties pane** in the right
        ii) Change **Visible** from On to **Off**

The card won't be visible but the value will get updated as part of the form submit

Add the button to submit the order

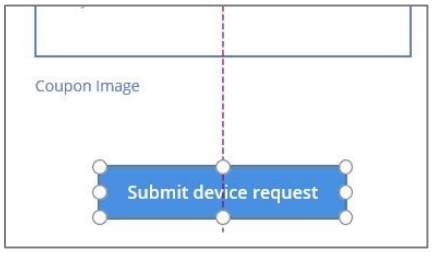
Customizing the edit form in the last task got all the necessary data into the form associated with the SharePoint list. Now we need a button to submit the form.

16. Copy and Paste one of the buttons from the first screen
    a) Copy the **Compare button** from the first screen which has the correct color values.



b) Paste (Ctrl-V) the button on the second CompareScreen.

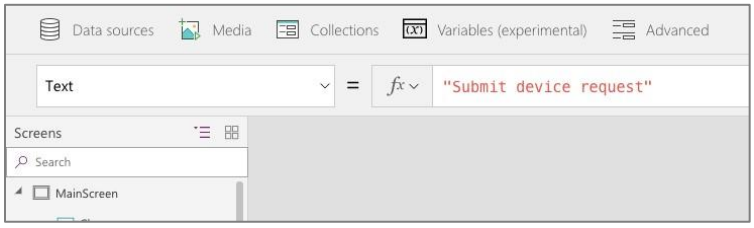
c) Position it in the bottom right of the screen, center aligned with the Form.



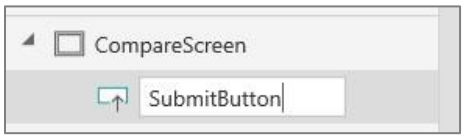
d) Make the button larger – you can resize to 280x60 using the Properties pane on the right.

17. Change the button text to “Submit device request”

a) Set the button’s **Text** property to **“Submit device request”**



b) Rename the button control to SubmitButton



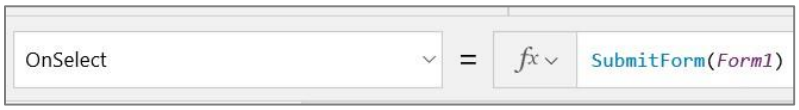
18. Set the Button Display Mode to only be enabled if a device is selected

a) To do this, change the button’s **DisplayMode** property to:

```
If(!IsBlank(CompareListGallery.Selected), DisplayMode.Edit, DisplayMode.Disabled)
```

b) When the button is clicked, submit the Form

i) Set the **OnSelect** property to: `SubmitForm(Form1)`



c) When the button is pressed, the form data will be submitted to the SharePoint Online list.

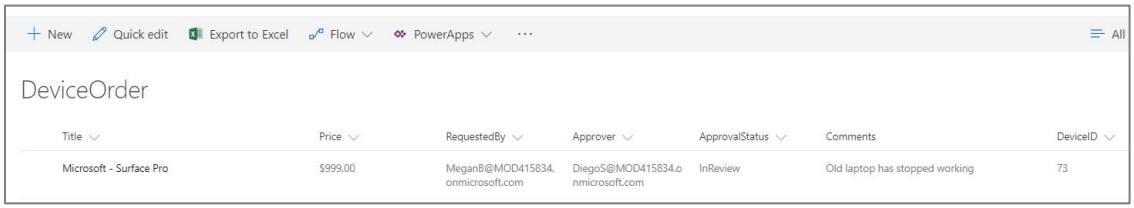
This is a good time to save the changes to your app.

Test a Device Order

That's all there is to submitting data to the SharePoint list, go ahead and give it a try:

19. Go to the first screen and play the app
20. Select a few devices to compare and hit the Compare button
21. As you select difference devices on the compare screen, you should see the information in the form change

22. Add a comment to the form and click Submit Device Request
23. Go back to where you created the DeviceOrder SharePoint list. You should have your recent submission there.
    a) a. You should see a newly added row with your **DeviceOrder** list



You may stop here and proceed to Exercise 3 where you will use Microsoft Flow to create an approval request. Optionally, you may add a third confirmation screen to your app to show that the form submission was successful.

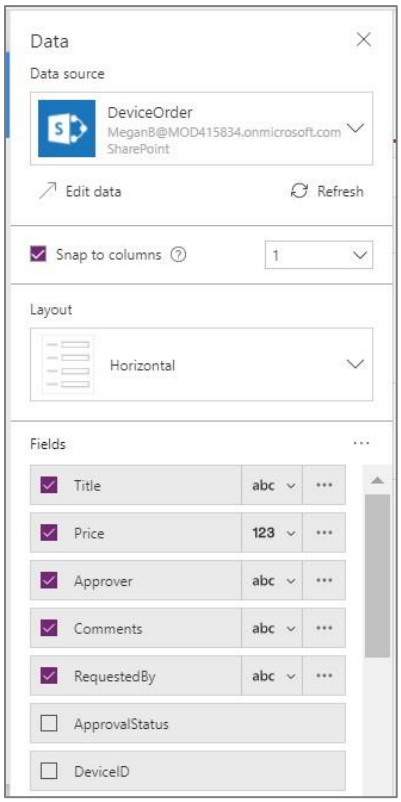
[Optional Step] Navigate to a confirmation screen after the form is submitted

In this optional task, you will show a confirmation screen with details about the device that was just requested.

24. Create a third screen
    a) Click Home -> New screen -> Blank
    b) Rename the screen to SubmitSuccessScreen
    c) Go back to the second screen
25. Add the navigation to the third screen
    a) Select the form control – use the tree view on the left to **select Form1**
    b) Set the **OnSuccess** property to: `Navigate(SubmitSuccessScreen,ScreenTransition.None)` This function to navigate will be run when the form submit is successful.
26. Add a header on the third screen

a) Copy (Ctrl-C) the header from the second screen
b) Go to the third screen and paste the header label

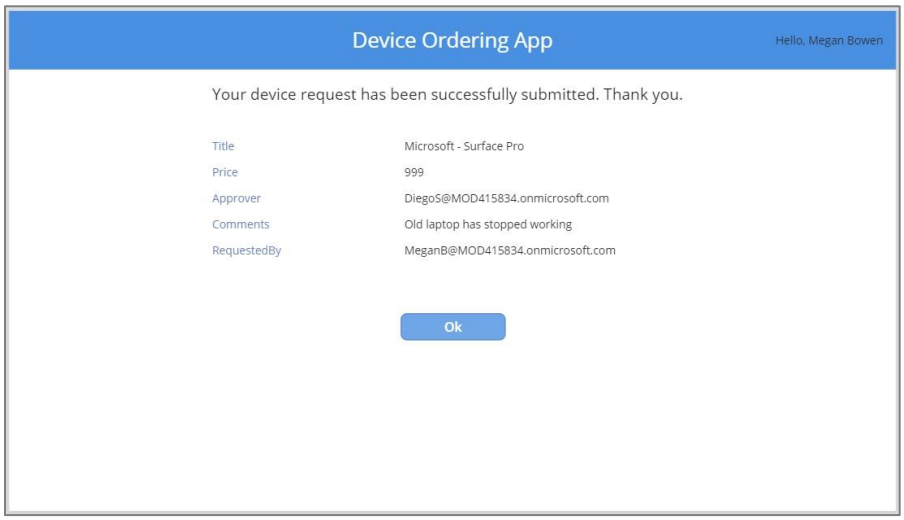
27. Add a thank you message in a label control
    a) Click Insert -> Label
    b) Set the Text to: "`Your device request has been successfully submitted. Thank you.`"
28. Add a Display Form to show the details of the device that was ordered
    a) Click Insert -> Form -> Display
    b) Configure the data source to point to the DeviceOrder list
    c) Change the Snap to columns value from 3 to 1
    d) Change the Layout from Vertical to Horizontal
    e) Uncheck the columns/fields that are not required.
    f) Rearrange the fields by dragging them up/down such that they match the image below



    g) Close the Data pane
    h) Set the **Item** property of the display form to: `Form1.LastSubmit`
    i) This is the item that was last submitted in the form on the second screen (Form1)
    j) Move and resize the display form control (see screenshot below)
29. Add an Ok button and set its behavior to clear the compare list and navigate to the first screen a. Click **Insert** -> **Button**
    a) Set the **Text** property of the button to: **"Ok"**
    b) Set the **OnSelect** property of the button: **Clear(CompareList); Navigate(MainScreen, ScreenTransition.None)**

Note: ';' is a separator used when calling multiple functions one after the other, like above. If you are in a locale where ';' is used instead of a comma as a separator within functions, then use a double semi-colon ';;' here to separate multiple function calls.

30. Test the app
    a) Submit another device approval request
    b) The confirmation screen should look like this image:



31. Make sure to **Save** and **Publish** the app