

# Building a Data-driven Canvas App

Lab Time: 60 minutes

Lab Folder: C:\Student\Modules\02\_DataDrivenCanvasApps\Lab

**Lab Overview:** In this lab you will build a data-driven canvas app based on data stored in a list in SharePoint Online. You will begin by creating a new SharePoint list to store customer data. Next, you will create a new canvas app using the **Start canvas app from blank** template. Over the course of this lab, you will work to add all the screens, galleries and forms required to create a user experience which allows viewing, adding, editing and deleting customers. In the final exercise, you will integrate a custom connector to retrieve new customer data from an external web service which will be used to streamline creating new customers.

## Exercise 1: Create a SharePoint List to Store Customer Data

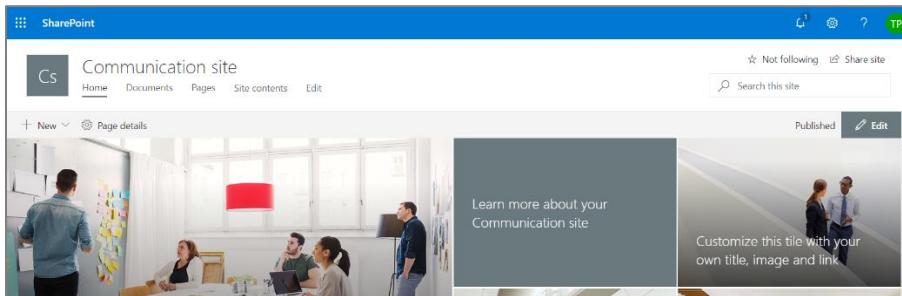
In this exercise, you will create a new modern SharePoint list named **Customers** to track customer data. After creating the **Customers** list, you will then add the necessary site columns and configure the default view to track customer data. In the final steps you will create a few new SharePoint list items to provide some sample data for when you build a canvas app in the next exercise.

### 1. Create a new SharePoint list named **Customers**.

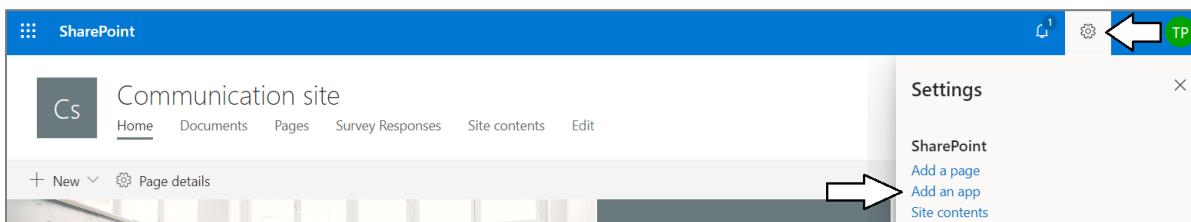
- Navigate to the root SharePoint site for your trial Office 365 tenancy.

The URL for your SharePoint root site can be determined by your Office 365 tenancy name. For example, if your Office 365 tenancy is named **bbspalabs**, then the URL for your SharePoint root site will be <https://bbspalabs.sharepoint.com>.

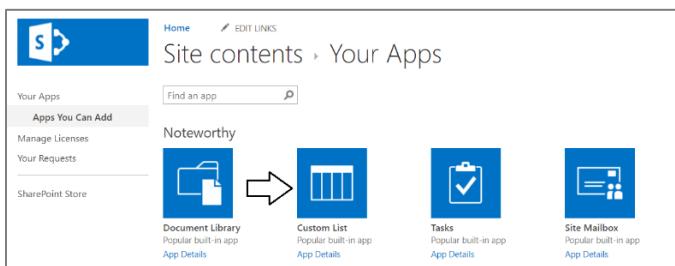
- When you navigate to your SharePoint site, it should be a SharePoint Communications site as shown in this screenshot.



- Click on the gear icon and then click on **Site contents**



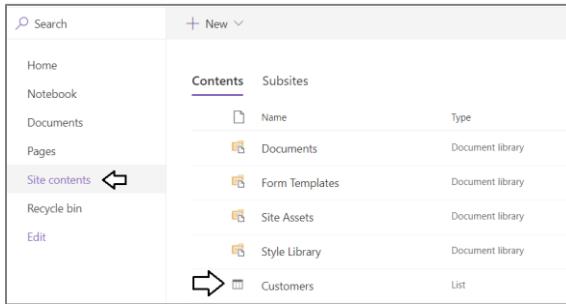
- Click the **Custom List** tile to create a new custom list.



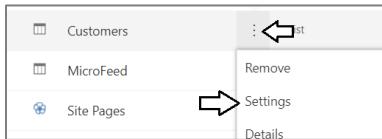
- e) When prompted with the **Adding Custom List** dialog, add a **Name of Customers** and click **Create**.



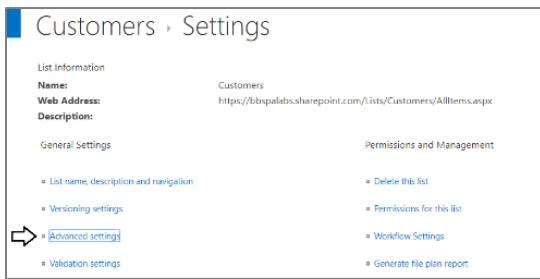
- f) Once the **Customers** list has been created, you should be able to locate this new list on the **Site contents** page.



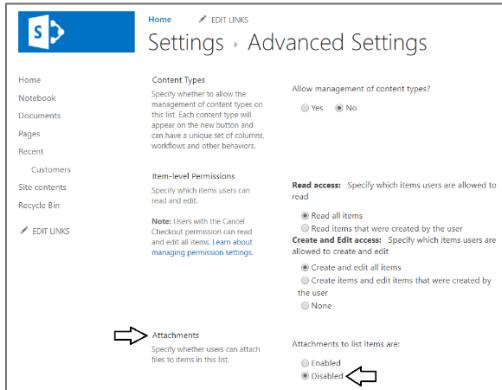
- g) On the **Site contents** page, drop down the context menu to the right of the **Customers** list and then click **Settings**.



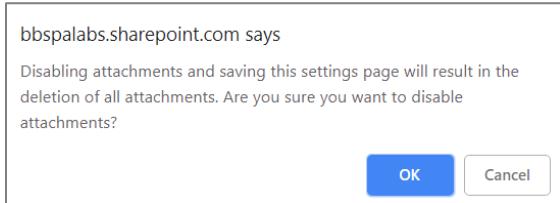
- h) On the **Settings** page, click the **Advanced Settings** link to navigate to the **Advanced Settings** page.



- i) On the **Advanced Settings** page, change the **Attachments** setting to **Disabled**.



- j) When prompted with the warning about disabling attachments, click **OK** to continue.



- k) Scroll to the bottom of the **Advanced Settings** page and click **OK** to save your changes and return to the **Settings** page.
2. Configure the columns for the new **Customers** list.
- In the **Setting** page, scroll down to the **Columns** section.
  - Click on the link for the column named **Title** to navigate to the **Edit Column** page

Columns		
A column stores information about each item in the list. The following columns are currently available in this list:		
Column (click to edit)	Type	Required
→ Title	Single line of text	✓
Modified	Date and Time	
Created	Date and Time	
Created By	Person or Group	
Modified By	Person or Group	

- c) The **Column name** should currently have a value of **Title**.

Home EDIT LINKS

Settings > Edit Column

Name and Type

Column name:  
Type a name for this column. → Title

- d) Update the **Column name** to **Last Name** to change the column's display name.

Home EDIT LINKS

Settings > Edit Column

Name and Type

Column name:  
Type a name for this column. → Last Name

- e) Click the **OK** button at the bottom of the **Edit Column** page to save your changes and return to the **Settings** page.
- f) You should be able to see the column name has been updated to **Last Name**.

Columns		
A column stores information about each item in the list. The following columns are currently available in this list:		
Column (click to edit)	Type	Required
→ Last Name	Single line of text	✓
Modified	Date and Time	
Created	Date and Time	
Created By	Person or Group	
Modified By	Person or Group	

Note that you have updated the column display name to **Last Name** but the underlying site column name is still **Title**.

- g) Click **Add from existing site columns** link to navigate to the **Add from existing site columns** page.

The screenshot shows a list of columns available in the current list:

Column (click to edit)	Type	Required
Last Name	Single line of text	✓
Modified	Date and Time	
Created	Date and Time	
Created By	Person or Group	
Modified By	Person or Group	

Below the list are three navigation links:

- Create column
- Add from existing site columns (highlighted with a red arrow)
- Column ordering

- h) On the **Add from existing site columns** page, drop down the **Select site columns from** menu.  
i) Set the **Select site columns from** dropdown menu to **Core Contact and Calendar columns**.

The screenshot shows the 'Add Columns from Site Columns' page. The 'Select site columns from' dropdown is set to 'Core Contact and Calendar Columns'. The 'Available site columns' list contains 'Address' and 'Anniversary', while the 'Columns to add' list is empty.

- j) In the **Available site columns** list, select **First Name** column and then click the **Add** button.

The screenshot shows the 'Add Columns from Site Columns' page. The 'Available site columns' list has 'First Name' selected. A red arrow points to the 'Add >' button next to the list.

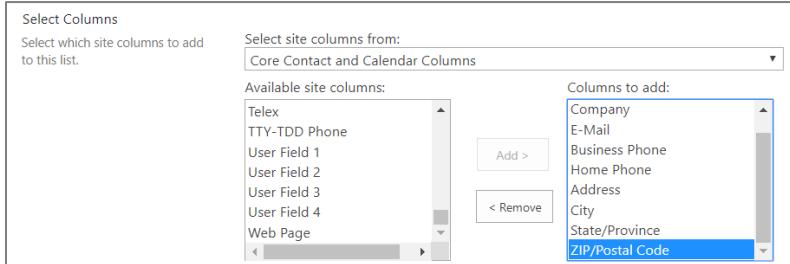
- k) You should now see the **First Name** column in the **Column to add** list.

The screenshot shows the 'Add Columns from Site Columns' page. The 'Available site columns' list has 'First Name' selected. A red arrow points to the 'First Name' column in the 'Columns to add' list.

You will now follow the same steps to add several more site columns to the **Columns to add** list.

- l) Add the column named **Company**.
- m) Add the column named **E-Mail**.
- n) Add the column named **Business Phone**.
- o) Add the column named **Home Phone**.
- p) Add the column named **Address**.

- q) Add the column named **City**.
- r) Add the column named **State/Province**.
- s) Add the column named **ZIP/Postal Code**.
- t) Now that you have added all the site columns you need to the **Columns to add** list, click the **OK** button to save your changes.



- u) You should now be able to see the columns you've added in the **Columns** section of the **Settings** page.

Columns		
A column stores information about each item in the list. The following columns are currently available in this list:		
Column (click to edit)	Type	Required
Last Name	Single line of text	✓
First Name	Single line of text	
Company	Single line of text	
E-Mail	Single line of text	
Business Phone	Single line of text	
Home Phone	Single line of text	
Address	Multiple lines of text	
City	Single line of text	
State/Province	Single line of text	
ZIP/Postal Code	Single line of text	
Modified	Date and Time	
Created	Date and Time	
Created By	Person or Group	
Modified By	Person or Group	

### 3. Configure the columns displayed in the default view of the **Customers** list.

- a) In the **Settings** page, scroll down to the **Views** section.
- b) Click on the **All Items** link to open this view in the **Edit View** page.

Views		
A view of a list allows you to see a particular selection of items or to see the items sorted in a particular order. Views currently configured for this list:		
View (click to edit)	Default View	Mobile View
→ All Items	✓	✓
<a href="#">Create view</a>		

- c) You should be able to see that all site columns you added to the list were also added to the default view.

Columns		Display	Column Name	Position from Left
Select or clear the check box next to each column you want to show or hide in this view of this page. To specify the order of the columns, select a number in the <b>Position from left</b> box.				
<input checked="" type="checkbox"/>	Last Name (linked to item with edit menu)	<input checked="" type="checkbox"/>	Last Name	1 ▾
<input checked="" type="checkbox"/>	First Name	<input checked="" type="checkbox"/>	First Name	2 ▾
<input checked="" type="checkbox"/>	Company	<input checked="" type="checkbox"/>	Company	3 ▾
<input checked="" type="checkbox"/>	E-Mail	<input checked="" type="checkbox"/>	E-Mail	4 ▾
<input checked="" type="checkbox"/>	Business Phone	<input checked="" type="checkbox"/>	Business Phone	5 ▾
<input checked="" type="checkbox"/>	Home Phone	<input checked="" type="checkbox"/>	Home Phone	6 ▾
<input checked="" type="checkbox"/>	Address	<input checked="" type="checkbox"/>	Address	7 ▾
<input checked="" type="checkbox"/>	City	<input checked="" type="checkbox"/>	City	8 ▾
<input checked="" type="checkbox"/>	State/Province	<input checked="" type="checkbox"/>	State/Province	9 ▾
<input checked="" type="checkbox"/>	ZIP/Postal Code	<input checked="" type="checkbox"/>	ZIP/Postal Code	10 ▾

- d) Uncheck the **Display** checkbox for **Address**, **City**, **State/Province**, **ZIP/Postal Code**.

The screenshot shows the 'Columns' settings dialog in SharePoint. It lists various columns with checkboxes for 'Display'. The columns are: Last Name (linked to item with edit menu), First Name, Company, E-Mail, Business Phone, Home Phone, Address, City, State/Province, and ZIP/Postal Code. The checkboxes for Address, City, State/Province, and ZIP/Postal Code are unselected. To the right, there's a 'Position from Left' column with numbered dropdowns for reordering.

- e) Click **OK** at the bottom of the **Edit View** page to save your changes to the **All Items** view and navigate to the default view.  
f) The columns in view should now include **Last Name**, **First Name**, **Company**, **E-Mail**, **Business Phone** and **Home Phone**.

The screenshot shows the 'Customers' list in SharePoint. The columns visible are Last Name, First Name, Company, E-Mail, Business Phone, and Home Phone. The 'Last Name' column has a dropdown arrow indicating it can be sorted.

4. Add a new sample customer item to the **Customers** list.

- a) Click the **New** button to display the SharePoint add item form.

The screenshot shows the SharePoint ribbon with the 'New' button highlighted. The ribbon tabs include Home, BL, BBSPA Labs Team Site, Search, Quick edit, Export to Excel, Flow, PowerApps, and more.

- b) Enter a **Last Name**, **First Name**, **Company** and **E-Mail** using the sample data below then click **Save** to add the new item.

The screenshot shows the 'New item' form for the Customers list. The fields filled are: Last Name (Smith), First Name (John), Company (Some Company), and E-Mail (SomeEmail@SomeDomain.com).

- c) You should be able to see the new customer item you've just created in the **Customers** list.  
d) Click the **Quick edit** button to enter quick edit mode to add a second customer item

The screenshot shows the 'Customers' list in SharePoint. A new item has been added with the following data: Last Name (Smith), First Name (John), Company (Some Company), and E-Mail (SomeEmail@SomeDomain.com). The 'Quick edit' button is highlighted in the ribbon.

- e) After adding a second customer item, click **Exit quick edit** to exit quick edit mode and save your changes.

Last Name	First Name	Company	E-Mail
Smith	John	Some Company	SomeEmail@SomeDomain.com
Doe	Jane	Doe Consulting	janedoe@gemail.com

- f) The **Customers** list should now have at least two customer items.

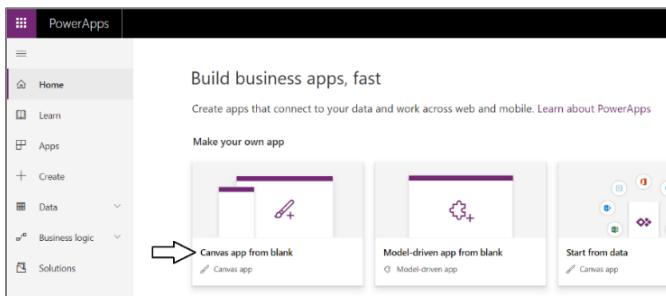
Last Name	First Name	Company	E-Mail
Smith	John	Some Company	SomeEmail@SomeDomain.com
Doe	Jane	Doe Consulting	janedoe@gemail.com

You are now done creating the SharePoint list to store customer data. In the next exercise, you will begin building a canvas app to manage the customer data in this list.

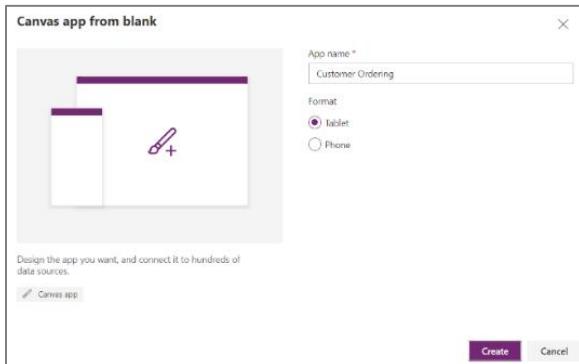
## Exercise 2: Create the Customer Ordering Canvas App

In this exercise, you will create a new canvas app named **Customer Ordering** using the **Canvas app from blank** template. You add the screens required to view and edit customer data and then you will create a navbar allowing users to navigate between screens.

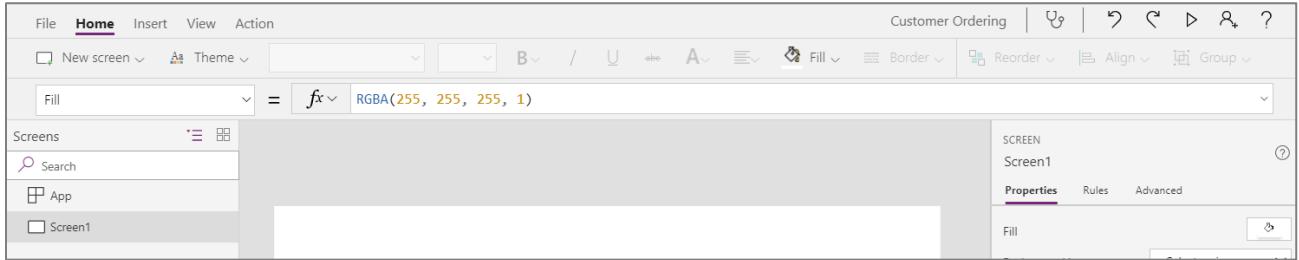
1. Create a new canvas app using the **Canvas app from blank** data template.
  - a) Navigate to the PowerApps portal at <https://web.powerapps.com>.
  - b) Create a new canvas app by clicking the **Canvas app from blank** button.



- c) When prompted with the **Canvas app from blank** dialog, enter an **App name** of **Customer Ordering** and click **Create**.

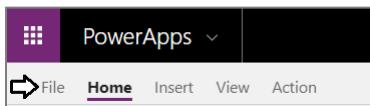


- d) After the new canvas app has been created, it should appear in PowerApps Studio as shown in the following screenshot.



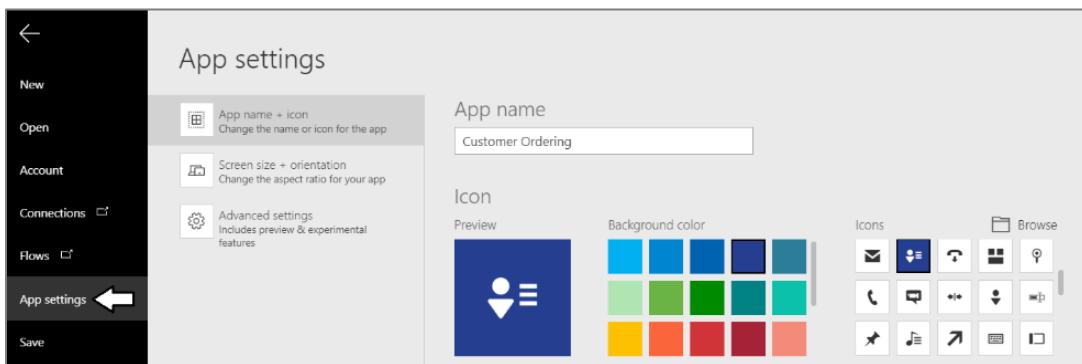
2. Configure the background color and icon for the new canvas app.

- a) Click the **File** menu to display the PowerApps backstage area.

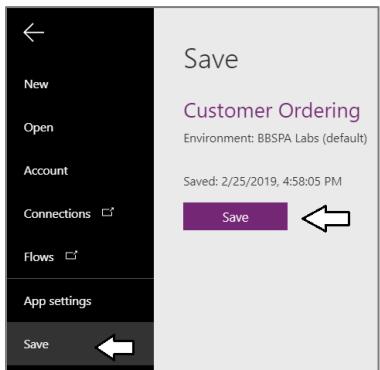


- b) Click on the **App settings** link to view the **App settings** page.

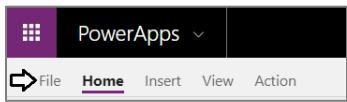
- c) Select a **Background color** and an **Icon** of your choosing.



- d) Click the **Save** link in the left navigation and then click the **Save** button to save your changes.

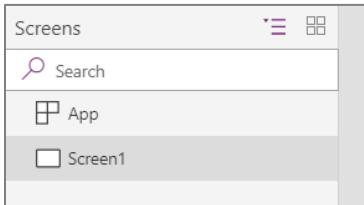


- e) Click the File menu once more to leave the backstage area and return the PowerApps Studio designer.

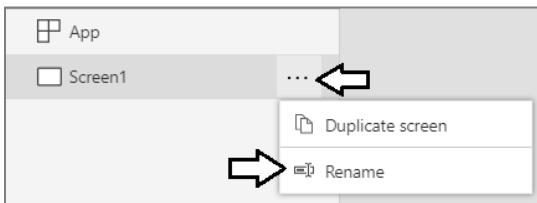


A new canvas app created from the **Canvas app from blank** template contains a single screen named **Screen1**. Over the next few steps, you will add and configure the first three screens for the canvas app and also create a user interface experience for navigation.

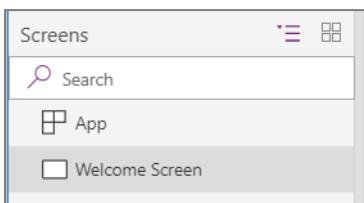
3. Create the three screen required for the canvas app.
  - a) Currently, the canvas app contains a single screen named **Screen1**.



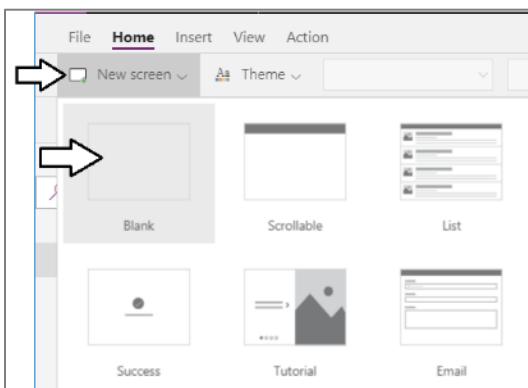
- b) Drop down the context menu to the right of **Screen1** and select the **Rename** command.



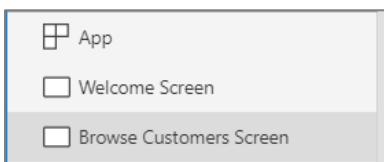
- c) Rename **Screen1** to **Welcome Screen**.



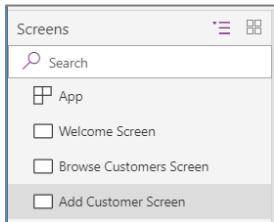
- d) From the **Home** tab in the ribbon, drop down the **New screen** menu and select **Blank**.



- e) Once the new screen has been added, rename the screen to **Browse Customers Screen**.

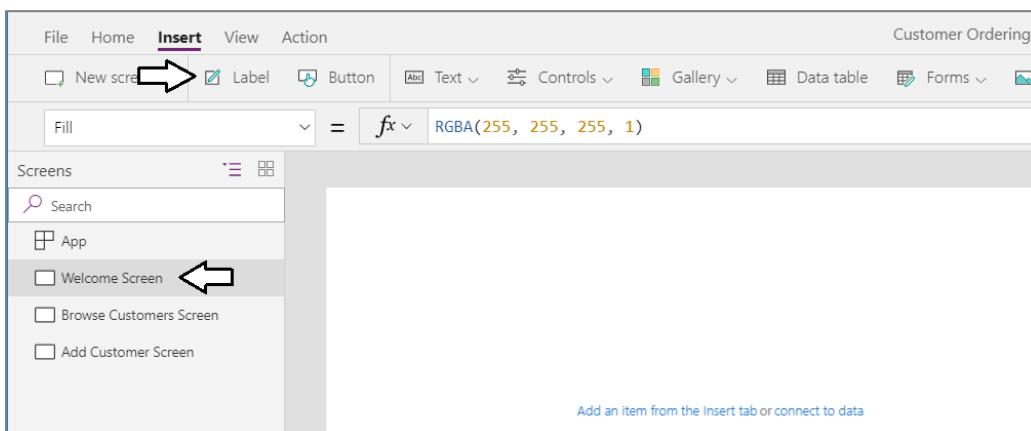


- f) From the **Home** tab in the ribbon, drop down the **New screen** menu and select **Blank** to add a third screen.
- g) Once the third screen has been added, rename the screen to **Add Customer Screen**.

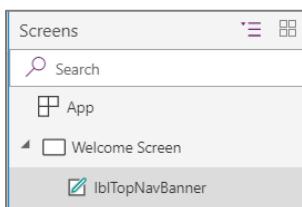


Now that you have added the three basic screens, you will now create a Top Nav Banner allowing users to navigate between screens.

4. Add a Top Nav Banner for navigation
- a) Navigate to the **Welcome Screen** by selecting it in the tree view on the left.



- b) Add a new label control onto the screen and rename it to **lblTopNavbanner**.



- c) Change the **Color** property of the label control to **White**.



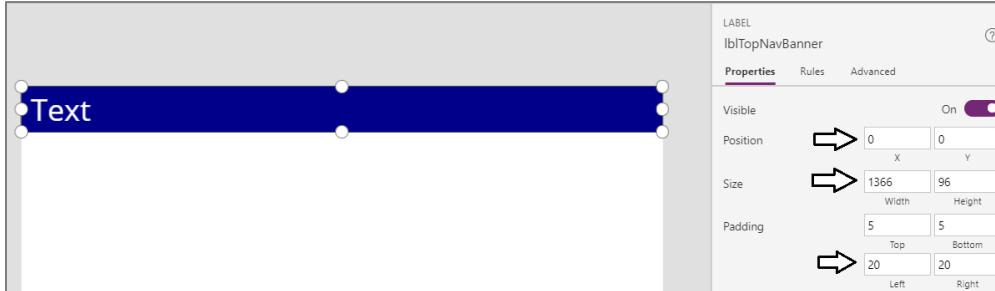
- d) Change the **Fill** property of the label control to **DarkBlue**.



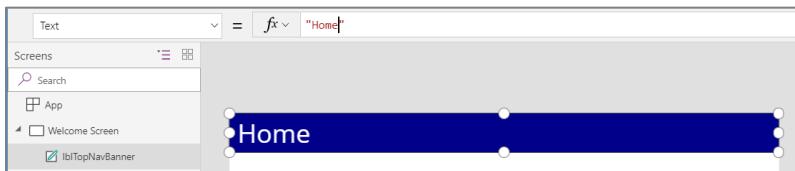
- e) Change the **Size** property of the label control to **48**.



- f) Position the label at the top of the screen.
- g) Set the label **Width** to the width of the screen which is **1366**.
- h) Set the label **Height** property to a value of **96**.
- i) Set the **Padding** properties to **Left** and **Right** to a value of **20**.



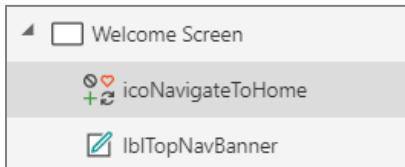
- j) Update the label **Text** property to **Home**.



- k) From the **Icons** menu on the **Insert** tab, add the **Home** icon to serve as a navigation button to the **Home Screen**.



- l) Update the name of the new icon to **icoNavigateToHome**.



- m) Change the **Color** property of **icoNavigateToHome** to **White**.



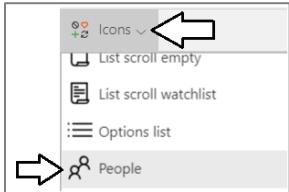
- n) Position **icoNavigateToHome** on top of **lblTopNavbanner** to the right as shown in the following screenshot.



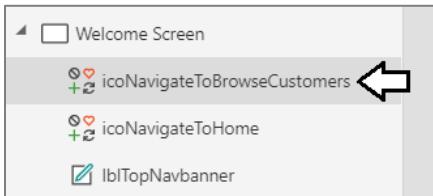
- o) Update the **OnSelect** property of on **icoNavigateToHome** using the following expression.

```
Navigate('Welcome Screen', ScreenTransition.None)
```

- p) Add a second icon using the **People** icon from the **Icons** menu.



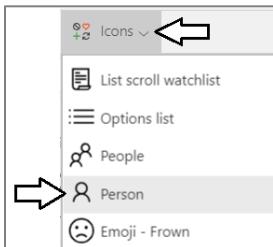
- q) Rename the new icon to **icoNavigateToBrowseCustomers**.



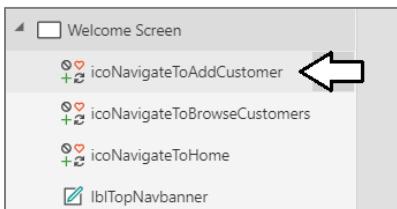
- r) Update the **OnSelect** property of **icoNavigateToBrowseCustomers** using the following expression.

```
Navigate('Browse Customers Screen', ScreenTransition.None)
```

- s) Add a third icon using the **Person** icon from the **Icons** menu.



- t) Rename the new icon to **icoNavigateToAddCustomer**.



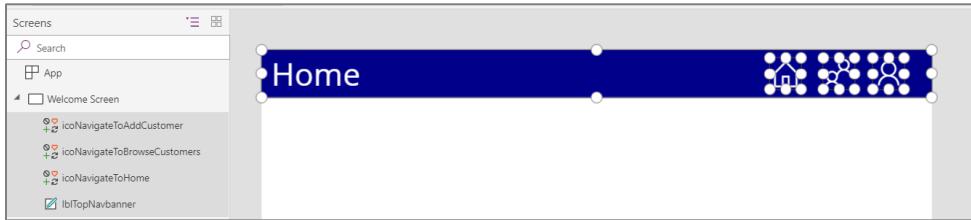
- u) Update the **OnSelect** property of **icoNavigateToAddCustomer** using the following expression.

```
Navigate('Add Customer Screen', ScreenTransition.None)
```

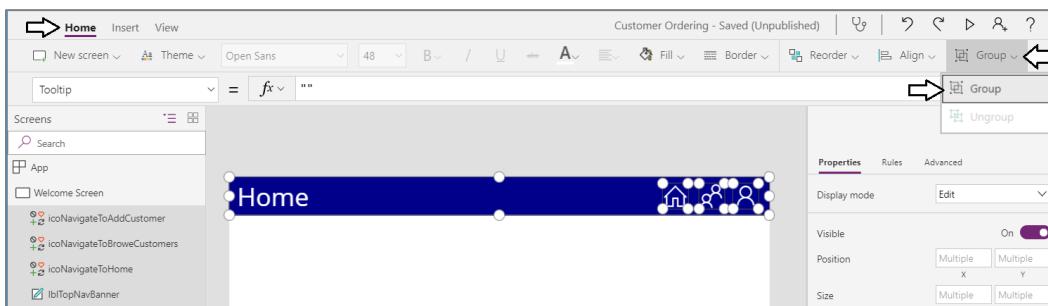
- v) Now you have completed work on the top nav banner which should match the following screenshot.



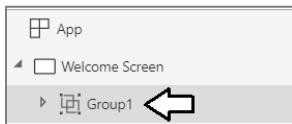
5. Combine the controls used to create the top nav banner into a group so it is **easier** to manage.
  - a) Select all four controls on the **Welcome Screen** by holding down the **SHIFT** key and selecting each one in the left tree view.
  - b) Make sure you have the label control and the three icons all selected at the same time as shown in the following screenshot.



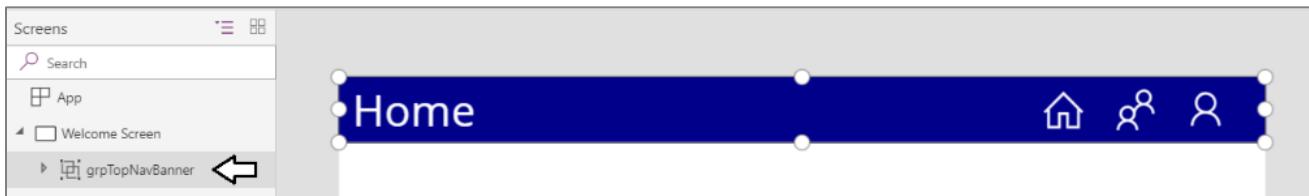
- c) Select the **Home** tab in the ribbon.
- d) Click the **Group** dropdown menu and then select the **Group** command to combine all 4 controls into a single group.



- e) You should see that all four controls have been added to a new group named **Group1**.

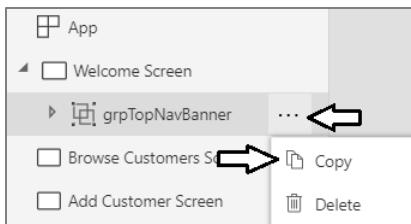


- f) Change the name of the new group from **Group1** to **grpTopNavBanner**.

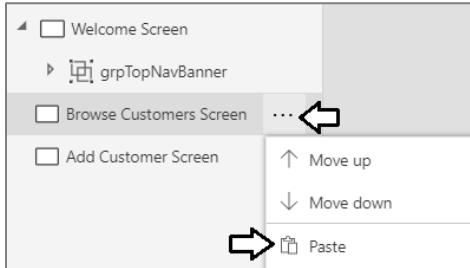


Over the next few steps you will copy and paste **grpTopNavBanner** to the other two screens to provide navigation across screens.

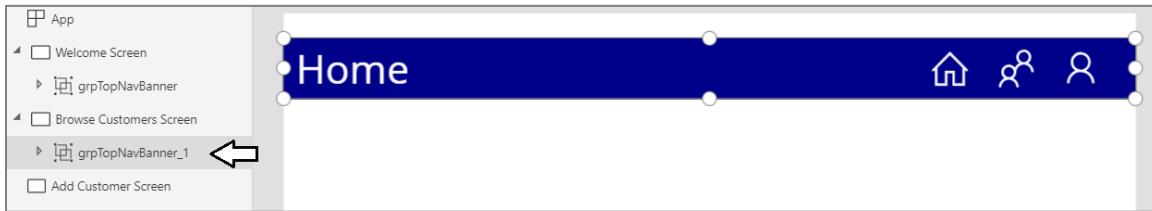
6. Copy **grpTopNavBanner** and paste it into the **Browse Customers Screen**.
  - a) Drop down the context ellipse (...) menu for **grpTopNavBanner** and then select the **Copy** command.



- b) Drop down the context men for the **Browse Customers Screen** and then select the **Paste** command.



- c) You should see that **grpTopNavBanner** has been copied and renamed to **grpTopNavBanner\_1**.



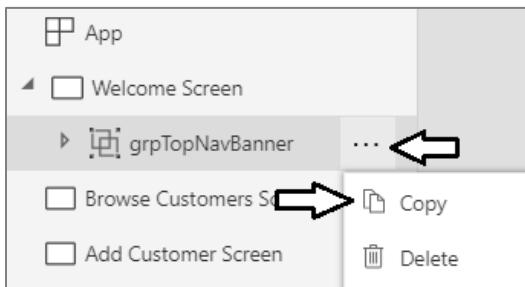
- d) Move **grpTopNavBanner\_1** to the top of the **Browse Customers Screen**.  
e) Expand **grpTopNavBanner\_1** and locate the label control named **lblTopNavbanner\_1**.  
f) Update the **Text** property of **lblTopNavbanner\_1** to **Browse Customers** as shown in the following screenshot.



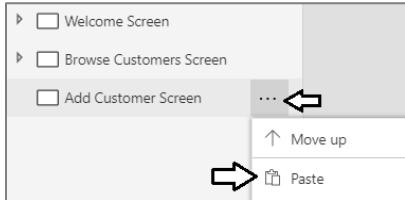
You will now move through a similar set of steps to copy **grpTopNavBanner** to the **Add Customer Screen**.

7. Copy **grpTopNavBanner** and paste it into the **Add Customer Screen**.

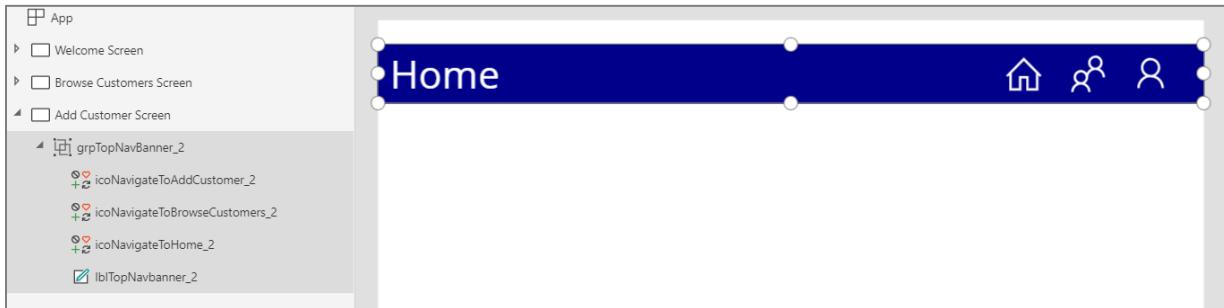
- a) Expand the **Welcome Screen** in the left tree view.  
b) Drop down the context ellipse (...) menu for **grpTopNavBanner** and then select the **Copy** command.



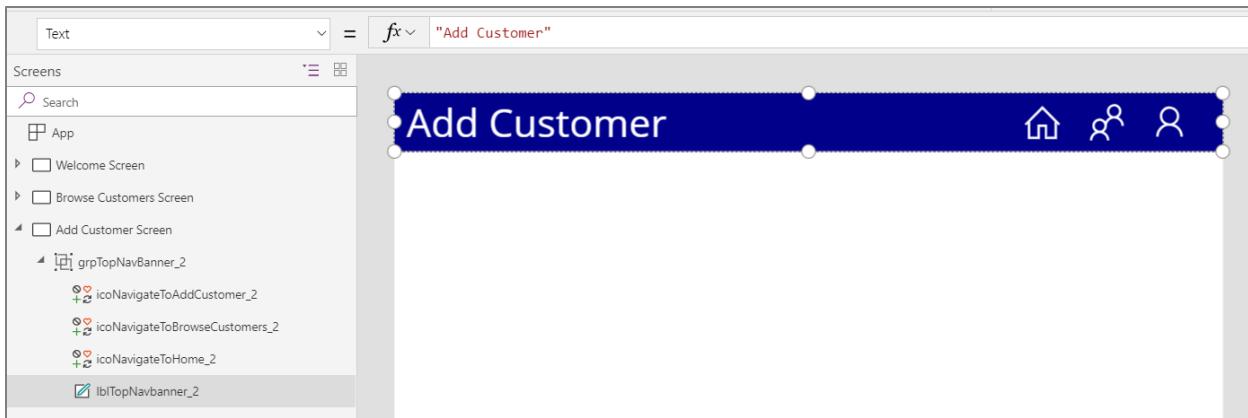
- c) Drop down the context men for the **Add Customer Screen** and then select the **Paste** command.



- d) You should see that **grpTopNavBanner** has been copied and renamed to **grpTopNavBanner\_2**.

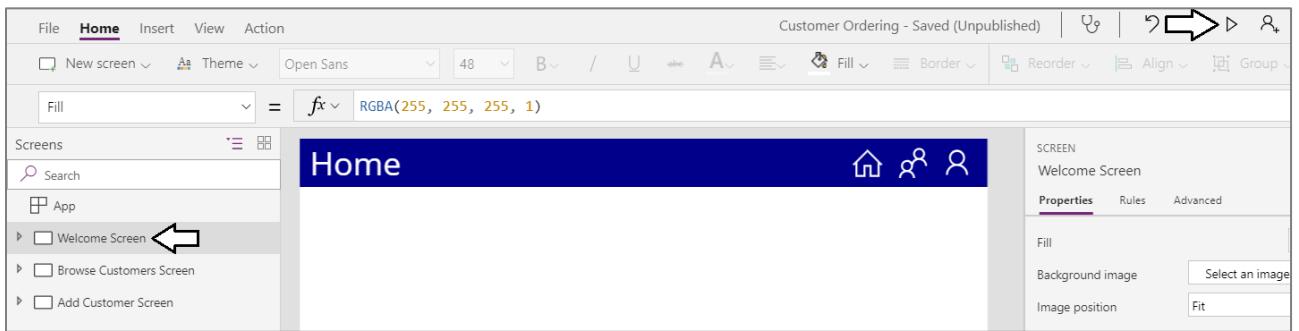


- e) Move **grpTopNavBanner\_2** to the top of the **Add Customer Screen**.  
f) Expand **grpTopNavBanner\_2** and locate the label control named **IblTopNavbanner\_2**.  
g) Update the **Text** property of **IblTopNavbanner\_2** to **Add Customer** as shown in the following screenshot.



8. Run the canvas app to test out navigation from screen to screen.

- a) Select the **Welcome Screen** in the left tree view and then click the **Play** button in the upper right corner.



- b) Click on the **People** icon and verify that it provides behavior to navigate to the **Browse Customers Screen**.



- c) Click on the **Person** icon and verify that it provides behavior to navigate to the **Add Customer Screen**.



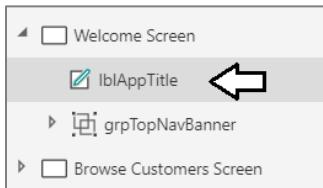
- d) Click on the **Home** icon and verify that it provides behavior to navigate to the **Home Screen**.  
e) When you are done testing, click the X on the far right to stop the canvas app from running.



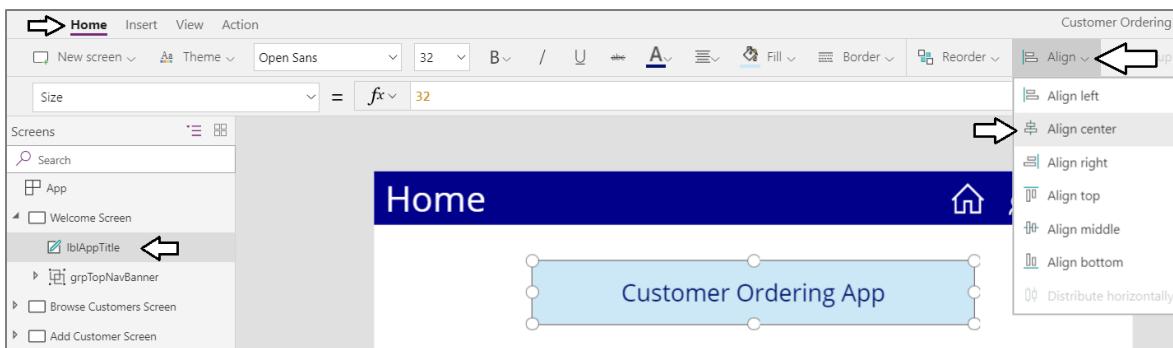
You will finish up this exercise by adding a few controls to build out the **Welcome Screen**.

9. Add controls to build out the **Welcome Screen**

- a) Add a new label and rename it to **IblAppTitle**.



- b) Set the **Size** property of the label to **32**.  
c) Set the **Width** property to **800** and the **Height** property to **120**.  
d) Change the **Fill** property to **LightBlue** and the **Color** property to **DarkBlue** (*or feel free to choose your own colors*).  
e) Change the **BorderThickness** to **1** and the **Border Color** to **Black**.  
f) From the **Home** tab in the ribbon, use the **Align > Align Center** command to center **IblAppTitle** in the middle of the screen.



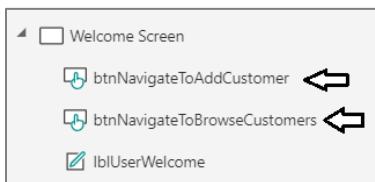
- g) Make a copy of the **IblAppTitle** control by copying it to the clipboard and then pasting it to the **Welcome Screen**.
- h) Once you have copied the label control, rename it to **IblUserWelcome**.
- i) Change the **Size** property of **IblUserWelcome** from **32** down to **24**.
- j) Update the **Text** property of **IblUserWelcome** using the following expression.

```
"Welcome " & User().FullName
```

- k) The **Welcome Screen** in your canvas app should match the following screenshot.



- l) Add two new button controls to the **Welcome Screen**.
- m) Once you have added the buttons, rename them to **btnNavigateToBrowseCustomers** and **btnNavigateToAddCustomer**.



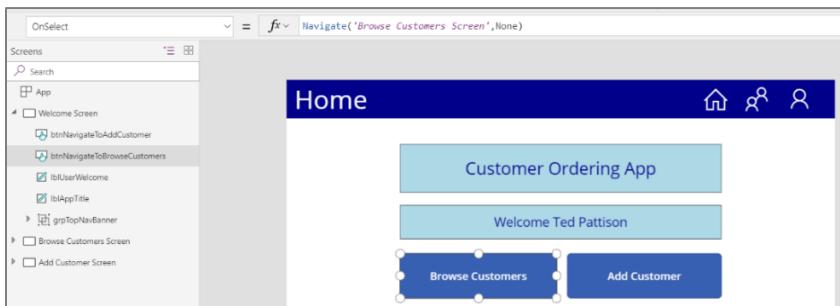
- n) Adjust the font size for each button to **20**.
- o) Set the **Text** property of **btnNavigateToBrowseCustomers** to **Browse Customers**.
- p) Set the **OnSelect** property of **btnNavigateToBrowseCustomers** using the following action expressions.

```
Navigate('Browse Customers Screen', ScreenTransition.None)
```

- q) Set the **Text** property of **btnNavigateToAddCustomer** to **Add Customer**.
- r) Set the **OnSelect** property of **btnNavigateToAddCustomer** using the following action expressions.

```
Navigate('Add Customer Screen', ScreenTransition.None)
```

- s) Reposition to two new buttons to match the following screenshot.



- t) Run the canvas app and make sure the buttons on the **Welcome Screen** provide the other two screens.

You have now completed this exercise. In the next exercise, you will create a connection to the **Customers** list in SharePoint and then you will add a gallery to the **Browse Customers Screen** to display customer data to the user.

### Exercise 3: Implement the Browse Customers Screen

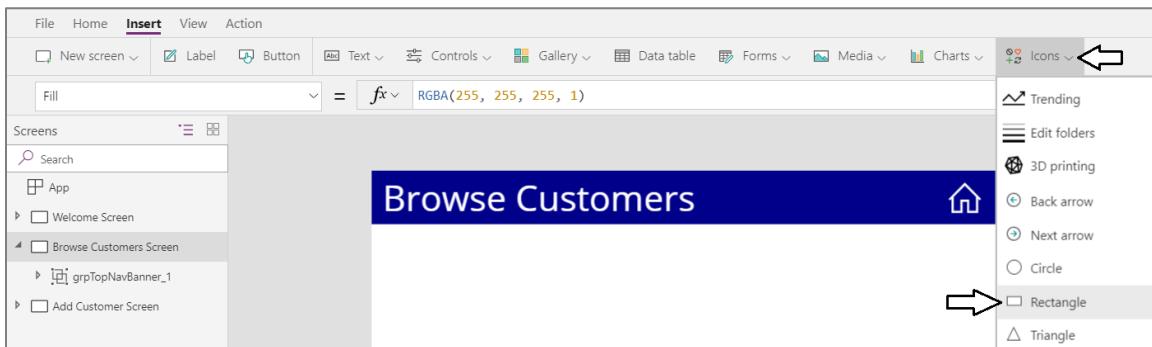
In this exercise, you will implement the **Browse Customers Screen** by adding a toolbar and a gallery control for displaying customers.

#### 1. Add a toolbar the **Browse Customers Screen**.

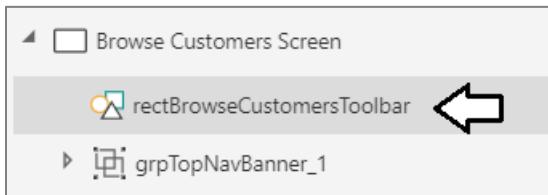
- Select the **Browse Customers Screen** from the left tree view.



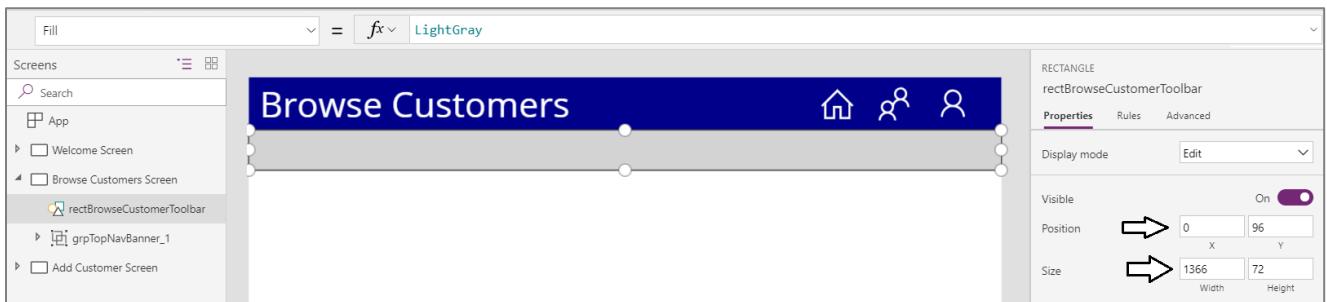
- Drop down the **Icons** menu, scroll towards the bottom and select **Rectangle** to add a new **Rectangle** control.



- Rename the **Rectangle** control to **rectBrowseCustomersToolbar**.



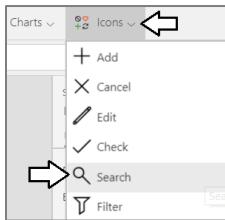
- Set the **Fill** property of **rectBrowsesCustomerToolbar** to **LightGray**.
- Change the **BorderThickness** to **1** and the **Border Color** to **Black**.
- Set the **Height** property to **72**.
- Reposition the **rectBrowseCustomerToolbar** control to match the following screenshot.



Now you will use the rectangle as the background placeholder for several more controls.

2. Add controls to implement the search toolbar.

- a) From the **Icons** menu, select **Search** to add a new search icon to the **Browse Customers Screen**.

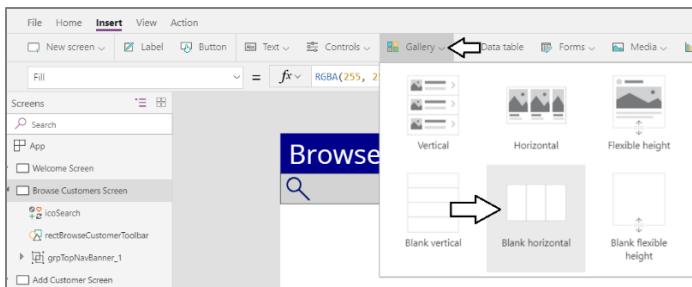


- b) Rename the icon control to **icoSearch** and position it on the left side of the toolbar as shown in the following screenshot.



3. Add a new gallery to display search buttons for each letter of the alphabet.

- a) Add a new gallery control by selecting the **Blank horizontal** option from the **Gallery** menu.



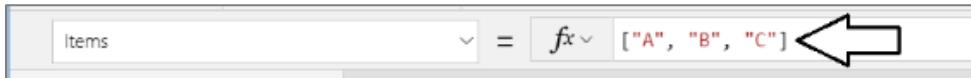
- b) Once the gallery control has been added, close the **Data** pane.



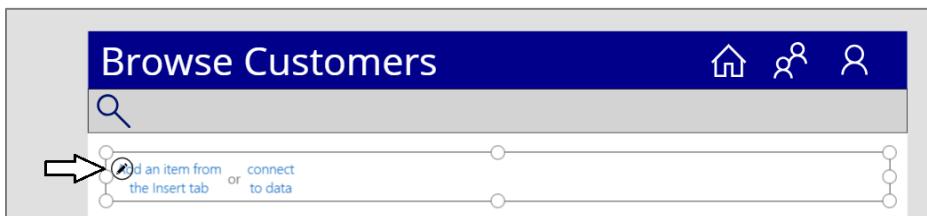
- c) Rename the new gallery to **galSearchButtons** and reposition its width and height to match the following screenshot.



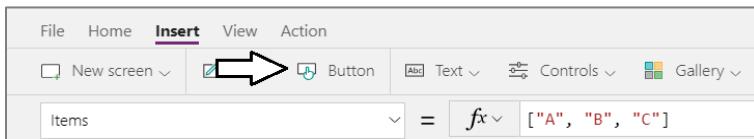
- d) Update the **Items** property of **galSearchButtons** with a value of ["A", "B", "C"].



- e) Hover over the new gallery and click the pen icon to edit the gallery template.



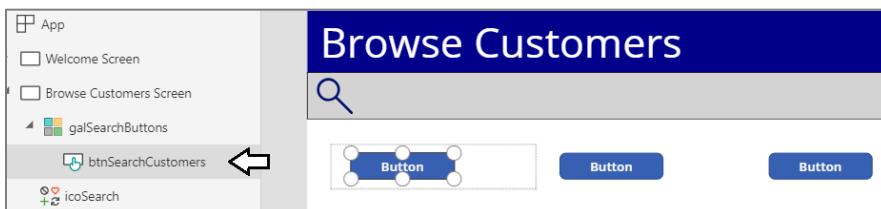
- f) Click the **Button** command on the **Insert** tab in the ribbon to add a new button into the gallery template.



- g) You should now see three buttons because there are three items in the **Items** collection for the gallery.



- h) Rename the button to **btnSearchCustomers** in the left tree view.

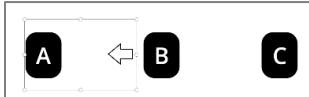


- i) Set the **Text** property of **btnSearchCustomers** to **ThisItem.Value**.

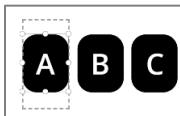


At this point, the three buttons should display the letters **A**, **B** and **C**.

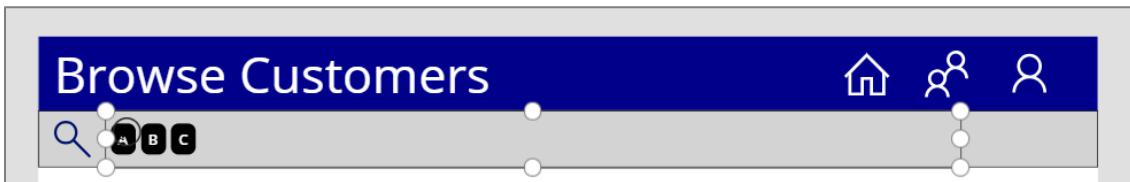
- j) Set the **Fill** property of **btnSearchCustomers** to **Black**.
- k) Change the **BorderThickness** to **0** and the **Border Color** to **Black**.
- l) Set the **Width** property to **32** and the **Height** to **40**.
- m) Position the button the on the left side of the template as shown in the following screenshot.



- n) Make the template narrower so it take up the same width as the button which as a Width of **32**.



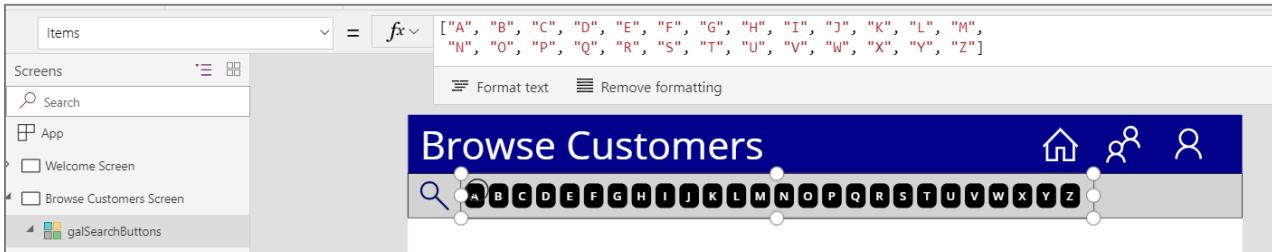
- o) Move **galSearchButtons** gallery control to it lays on top of the toolbar as shown in the following screenshot.



- p) Update the **Items** property of **galSearchButtons** with the following expression to include all the letters of the alphabet.

```
[ "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M",  
"N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z" ]
```

- q) The **galSearchButtons** gallery control should now display a button for each letter in the alphabet.



4. Add a new button to remove the search filter and show all customers.

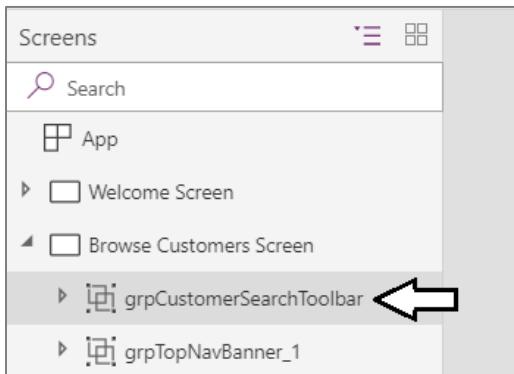
- a) Add a new button and rename it to **btnShowAllCustomers**.
- b) Set the **Text** property of **btnShowAllCustomers** to **Show All**.
- c) Set the **Fill** property to **Black** and change the **BorderThickness** to **0**.
- d) Position the **Show All** button to the right of the **galSearchButtons** gallery as shown in the following screenshot.



5. Create a new group with the four controls you added to create the customer search toolbar.
  - a) In tree view on the left, hold down the **SHIFT** key and select the following controls.
    - i) **btnShareAllCustomers**
    - ii) **galSearchButtons**
    - iii) **icoSearch**
    - iv) **rectBrowseCustomersToolbar**
  - b) With the four controls selected, use the **Group > Group** command from the **Home** tab to add them into a new group.

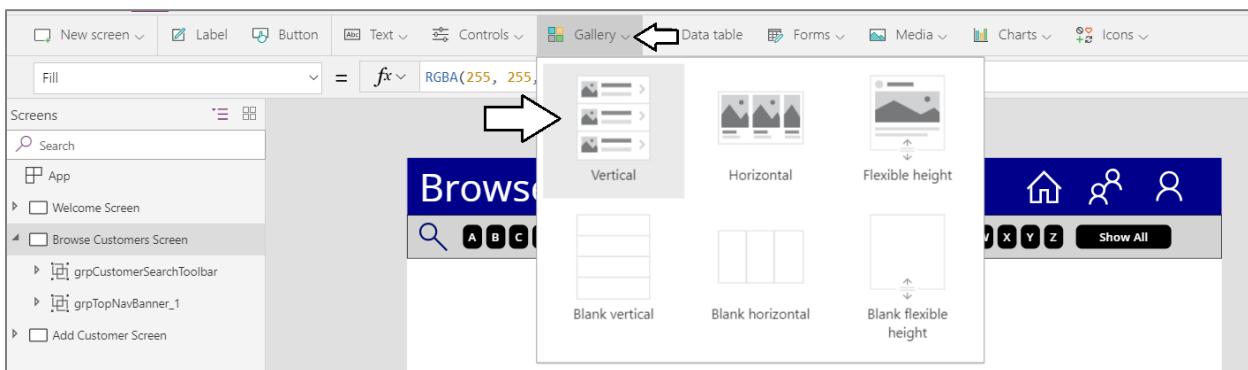


- c) Once you have created the new group, rename it to **grpCustomerSearchToolbar**.

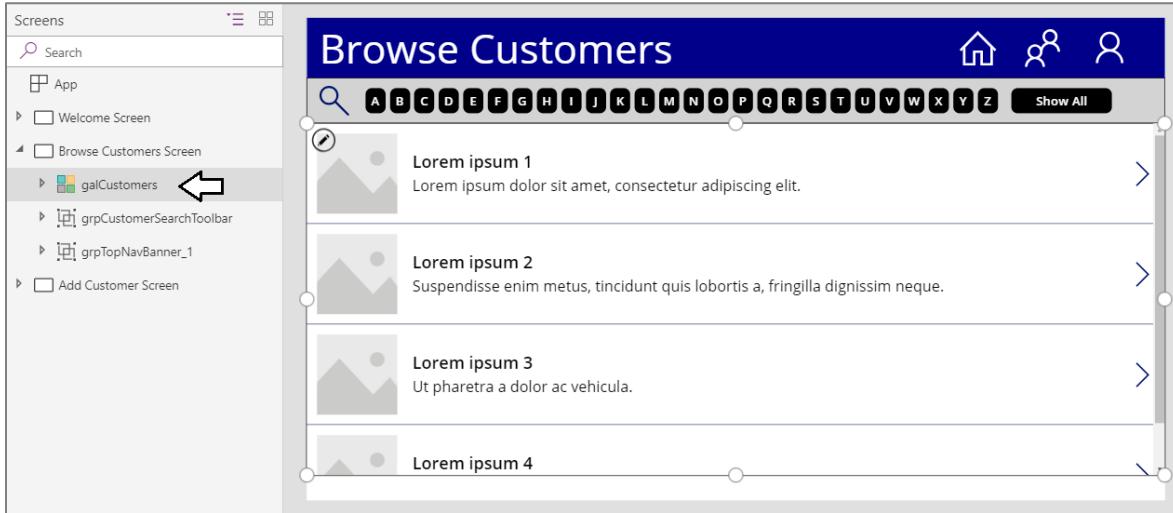


You have now finished creating the search toolbar UI and now you will add a new gallery to display customer data. After adding the gallery to display customer data you will then return to the search toolbar and implement the search functionality.

6. Add a new **Gallery** to display customer data.
  - a) Add a new **Gallery** control to the **Browse Customers Screen** by selecting the **Vertical** option from the **Gallery** menu.

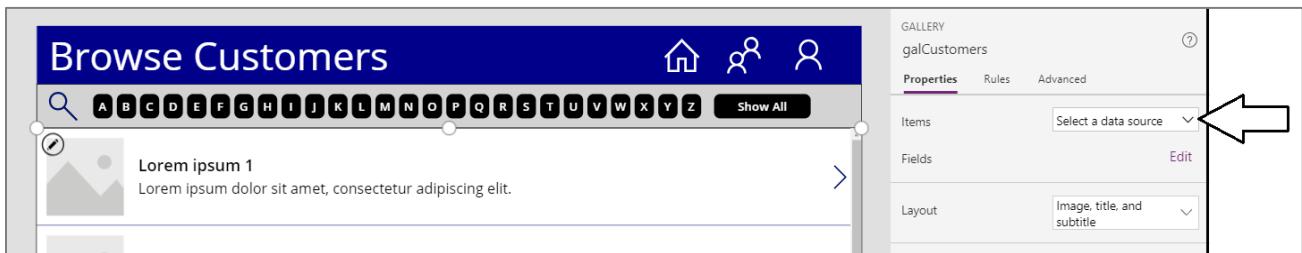


- b) Rename the gallery to **galCustomers** and reposition below the toolbar to match the following screenshot.

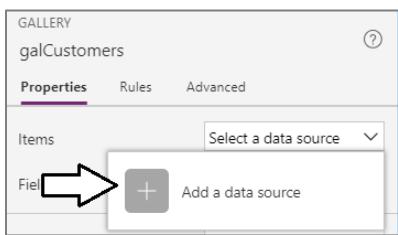


7. Configure the **Items** property of the **galCustomers** gallery by adding a new connection to the SharePoint customers list.

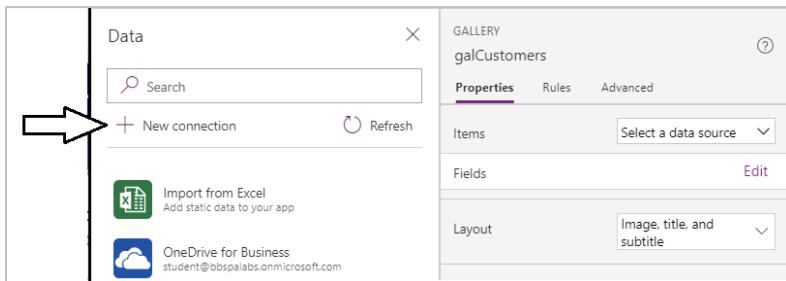
- In the **Property** pane for **galCustomers** on the right, locate the **Items** property.
- Click the **Items** property dropdown menu which has the caption **Select a data source**.



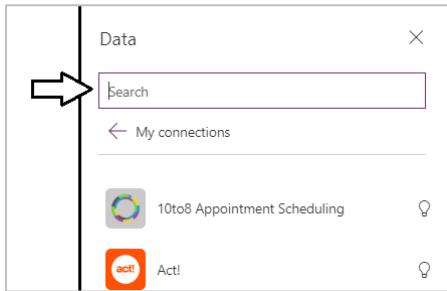
- Click the **Add a data source** menu command



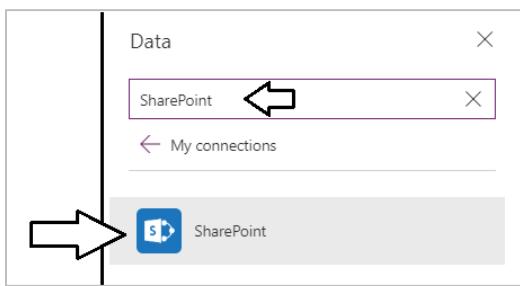
- Click the **New connection** button.



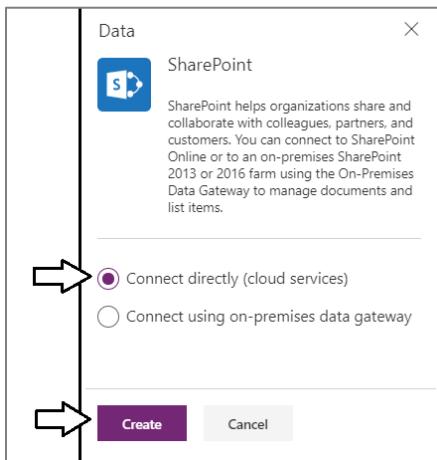
- e) In the **Data** pane, you will see a search box to search for connectors.



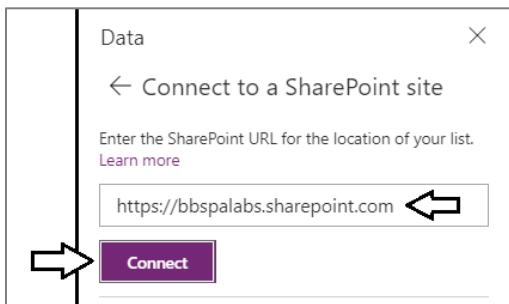
- f) Enter a search value of **SharePoint**.  
g) Select the **SharePoint** connector to begin creating the new connection.



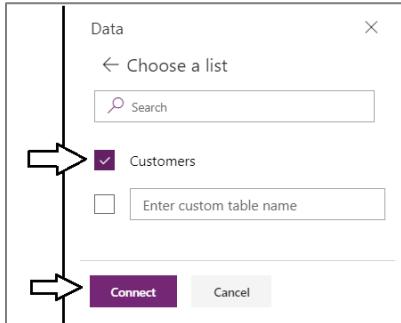
- h) On the SharePoint connector page, select **Connect directly (cloud services)** option and then click the **Create** button.



- i) When prompted to **Connect to a SharePoint site**, enter the URL for your SharePoint team site from Exercise 1.



- j) When prompted to **Choose a list**, select the **Customers** list and then click **Connect**.

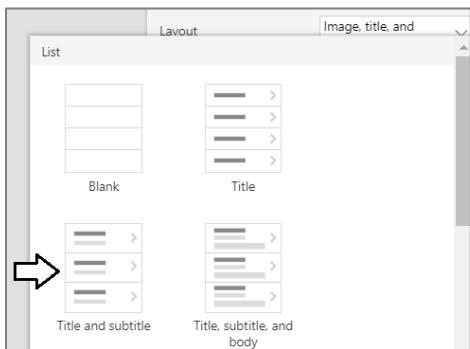


- k) Once the connection is established, you should see SharePoint customer data displayed in **galCustomers**.

You can see that the **Layout** property of **galCustomers** is automatically set to **Image, title and subtitle**.

8. Configure the gallery template for **galCustomers** to display customer data.

- a) Using the dropdown menu in the **Property** pane, change the gallery **Layout** to **Title and subtitle**.

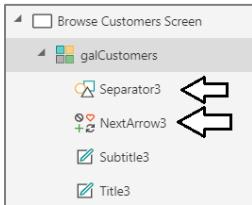


- b) Update the **WrapCount** property of **galCustomers** to 3 to display three customers per row in the gallery

- c) Click on the pen icon in **galCustomers** to enter editing mode for the gallery template.



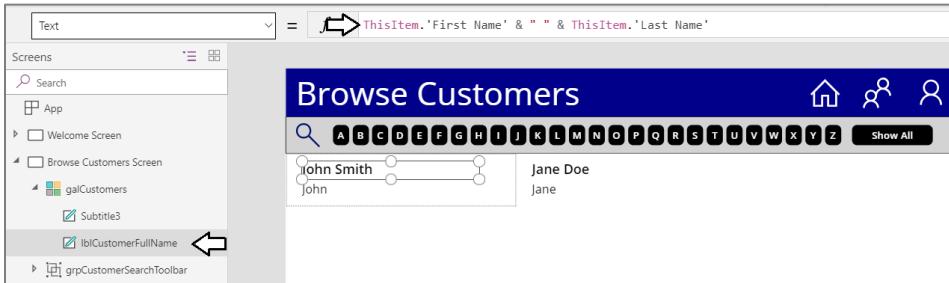
- d) Remove the separator control and the next arrow control.



- e) Rename the title label to **IblCustomerFullName**.  
f) Set the **Text** property of the **IblCustomerFullName** to the following expression.

```
ThisItem.'First Name' & " " & ThisItem.'Last Name'
```

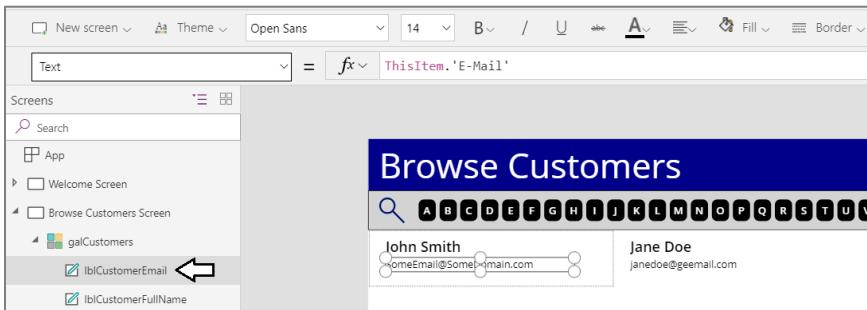
- g) The **Browse Customers Screen** in your canvas app should now match the following screenshot.



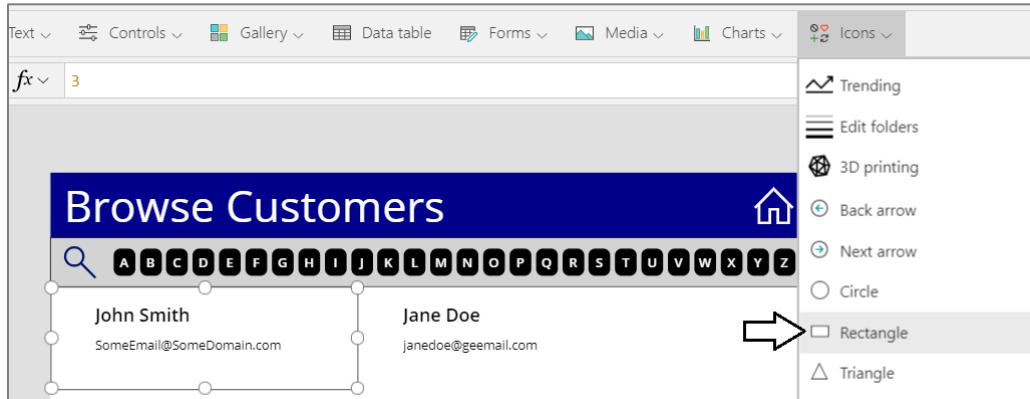
- h) Rename the subtitle label to **IblCustomerEmail**.  
i) Set the **Size** property of **IblCustomerEmail** to 14.  
j) Set the **Text** property of the **IblCustomerEmail** to the following expression.

```
ThisItem.'E-Mail'
```

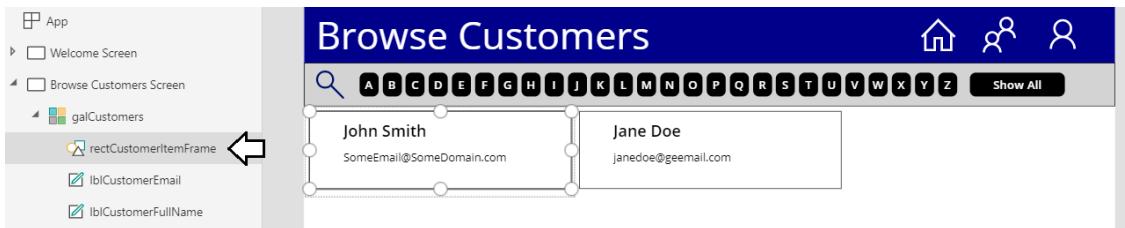
- k) The **Browse Customers Screen** in your canvas app should now match the following screenshot.



9. Add a new rectangle control to the template gallery of **galCustomers** to provide a frame around each customer item.
  - a) Make sure you are in edit mode for the gallery template of **galCustomers**.
  - b) Add a new Rectangle from the **Icons** menu.

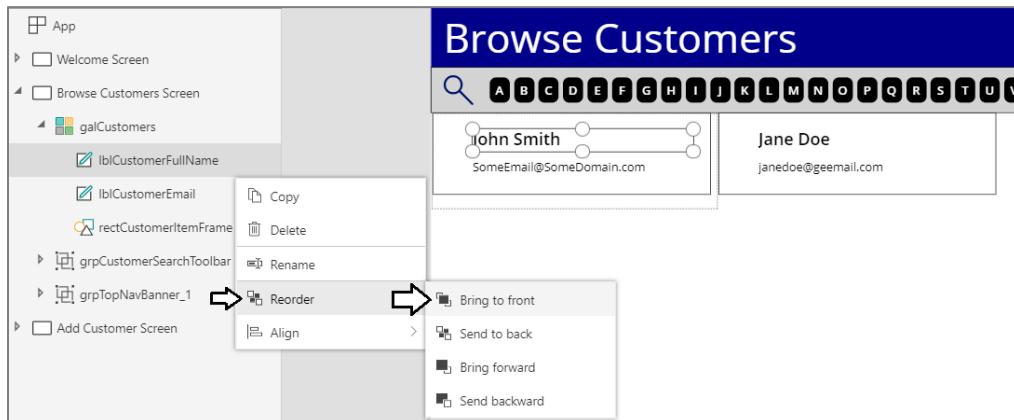


- c) Rename the rectangle control to **rectCustomerItemFrame**.
- d) Set the **Fill** property of **rectCustomerItemFrame** to **RGBA(0, 0, 0, 0)** to make it transparent.
- e) Set the **BorderThickness** to **1** and the **BorderColor** to **Black**.
- f) Reposition **rectCustomerItemFrame** to the top and full width of the gallery template as shown in the following screenshot.



You might find it hard to select **IblCustomerFullName** and **IblCustomerEmail** because they are layered behind the rectangle control named **rectCustomerItemFrame**. Over the next two steps you will move the labels on top of the rectangle.

- g) From the **IblCustomerFullName** context menu, select **Reorder > Bring to front** to move it above **rectCustomerItemFrame**.

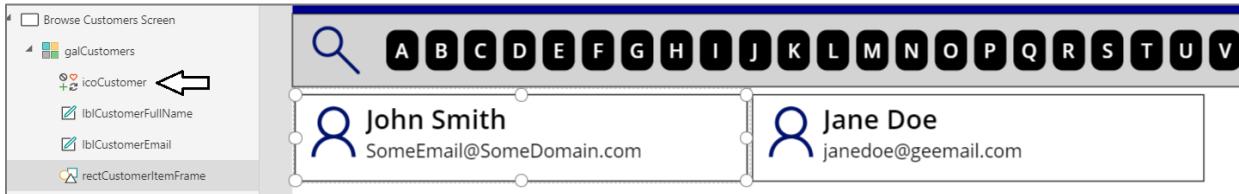


- h) From the **IblCustomerEmail** context menu, select **Reorder > Bring to front** to move it above **rectCustomerItemFrame**.

You can now select **IblCustomerFullName** and **IblCustomerEmail** because they are layered above **rectCustomerItemFrame**.

10. Add a new icon to the **galCustomers** item template.

- Make sure you are in edit mode for the gallery template of **galCustomers**.
- Add a new **Person** icon from the **Icons** menu.
- Rename the new icon to **icoCustomer** and rearrange its controls in the item template to match the following screenshot..



11. Set the **TemplateFill** property of **galCustomers** to indicate to the customer item which item is currently selected.

- Update the **TemplateFill** property of **galCustomers** with the following expression.

```
If(ThisItem.IsSelected, LightYellow, White)
```

- You should see that the first item in the gallery is highlighted in a light yellow color.



- Start the app and test out clicking customer items. You should observe that the yellow fill follows the currently selected item.



- After testing, click the X at the top right to stop the app and return to editing mode in PowerApps Studio.

12. Implement search filtering behavior in the search toolbar.

- Select **btnSearchCustomers** inside **galSearchButtons** and update its **OnSelect** property using the following expressions.

```
UpdateContext({ customerFilter: ThisItem.Value })
```

- Your expression for the **OnSelect** property of **galSearchButtons** should match the following screenshot.



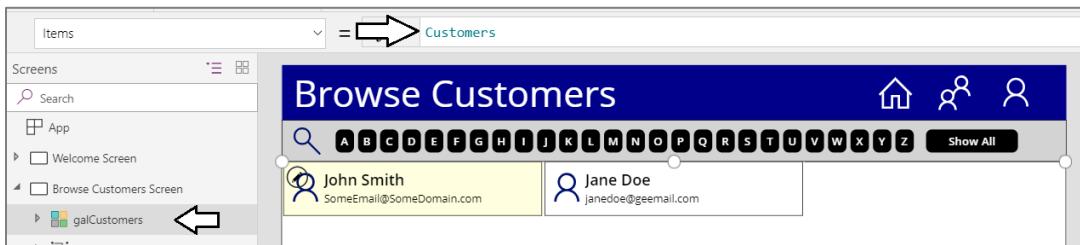
- c) Select **btnShowAllCustomers** and update its **OnSelect** property using the following expressions.

```
UpdateContext({ customerFilter: Blank() })
```

- d) Your expression for the **OnSelect** property of **btnShowAllCustomers** should match the following screenshot.



- e) Select the **galCustomers** gallery control and observe that the **Items** property is currently set to **Customers**



- f) Update the **Items** property of **galCustomers** gallery to the following expressions.

```
If(
    IsBlank(customerFilter),
    Customers,
    Filter(Customers, StartsWith('Last Name', customerFilter))
)
```

- g) Start the app so you can test out the filtering behavior.



- h) Click the **D** button and verify the app only displays customers whose last name starts with D.



- i) Click the **Show All** button and verify the gallery displays all customers.



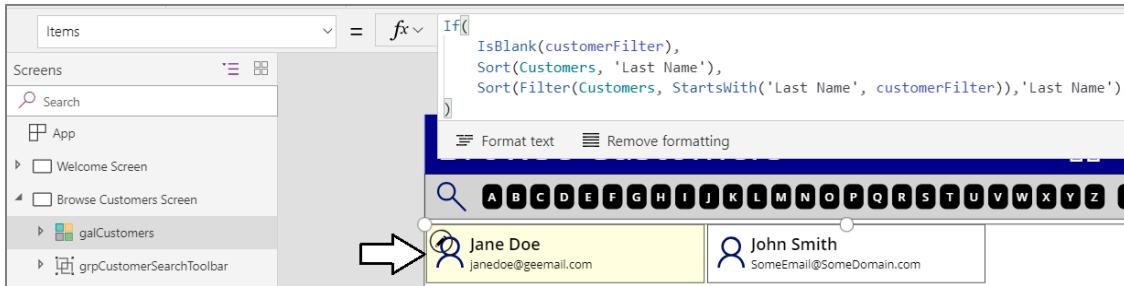
- j) After you have tested the filtering behavior, stop the app and return to edit mode in PowerApps Studio.

13. Modify the sort order of items in the **galCustomers** to sort by customer last name

- a) Modify the **Items** property of **galCustomers** with the following expression.

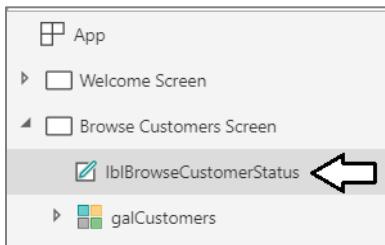
```
If(
    IsBlank(customerFilter),
    Sort(Customers, 'Last Name'),
    Sort(Filter(Customers, StartsWith('Last Name', customerFilter)), 'Last Name')
)
```

- b) You should now see the customer items are now sorted by last name.



14. Add a status indicator at the bottom of the **Browse Customer Screen**.

- a) Add a new label to the **Browse Customer Screen** and rename it to **IblBrowseCustomerStatus**.



- b) Set the **Fill** property of **IblBrowseCustomerStatus** to **LightGray**.  
 c) Change the **BorderThickness** to **1** and the **Border Color** to **Black**.  
 d) Set the **Height** property to **40** and the **Width** to **1366**.  
 e) Move the **IblBrowseCustomerStatus** control to the bottom of the **Browse Customer Screen**.  
 f) Set the **Text** property of **IblBrowseCustomerStatus** using the following expression.

```
"Filter: " & If(
    IsBlank(customerFilter),
    "All customers",
    "Last name starting with '" & customerFilter & "'"
) & " - " & CountRows(galCustomers.AllItems) & " Customers Found"
```

- g) Now when there is no filter set, the status bar will display the following message to the user.

Filter: All customers - 2 Customers Found

- h) If the user clicks a filter button with a letter such as **D** to apply a filter. the status message will update accordingly.

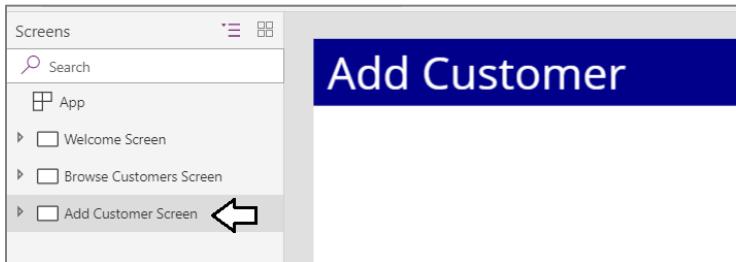
Filter: Last name starting with 'D' - 1 Customers Found

You have now completed the initial work on the **Browse Customers Screen** and its time begin work on the **Add Customer Screen**.

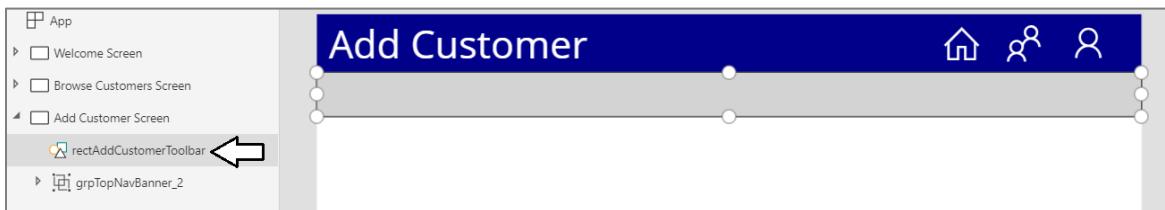
## Exercise 4: Implement the Add Customer Screen

In this exercise, you will implement the **Add Customer Screen** to allow users to add new items to the SharePoint **Customers** list.

1. Add a toolbar the **Add Customer Screen**.
  - a) Select the **Add Customer Screen** from the left tree view.

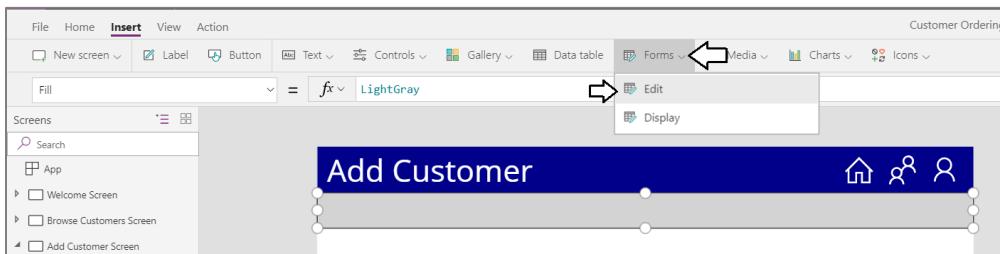


- b) Drop down the **Icons** menu, scroll towards the bottom and select **Rectangle** to add a new **Rectangle control**.
- c) Rename the **Rectangle** control to **rectAddCustomerToolbar**.
- d) Set the **Fill** property of **rectAddCustomerToolbar** to **LightGray**.
- e) Change the **BorderThickness** to **1** and the **Border Color** to **Black**.
- f) Set the **Height** property to **72**
- g) Reposition the **rectBrowseCustomerToolbar** control to match the following screenshot.

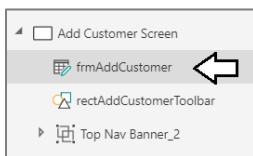


Now you will use the rectangle as the background placeholder for icons that will be used as command buttons.

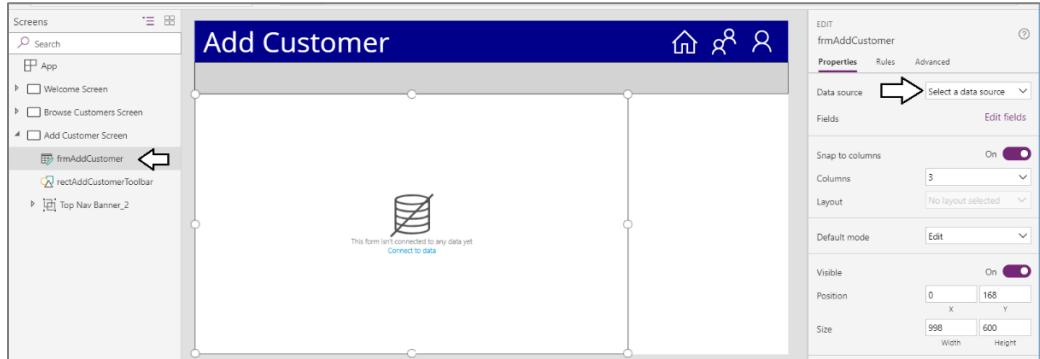
2. Add an edit form to the **Add Customers Screen**.
  - a) Select the **Insert** tab and select the **Forms > Edit** command to add a new edit form.



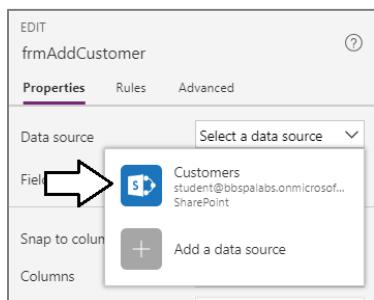
- b) Rename the edit form to **frmAddCustomer**.



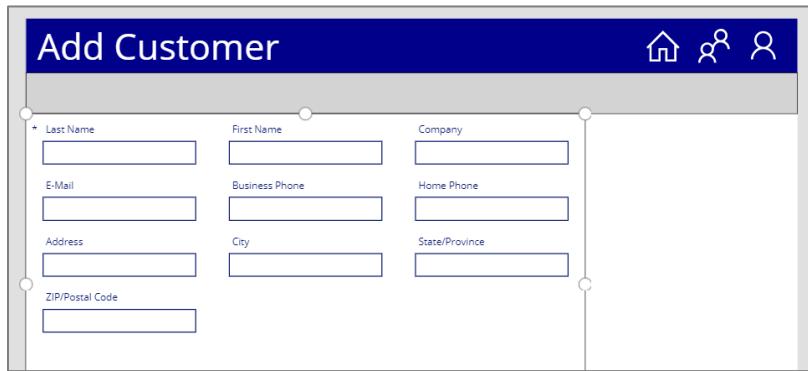
- c) Reposition the edit form as shown in this screenshot and click the **Data source** menu which displays **Select a data source**.



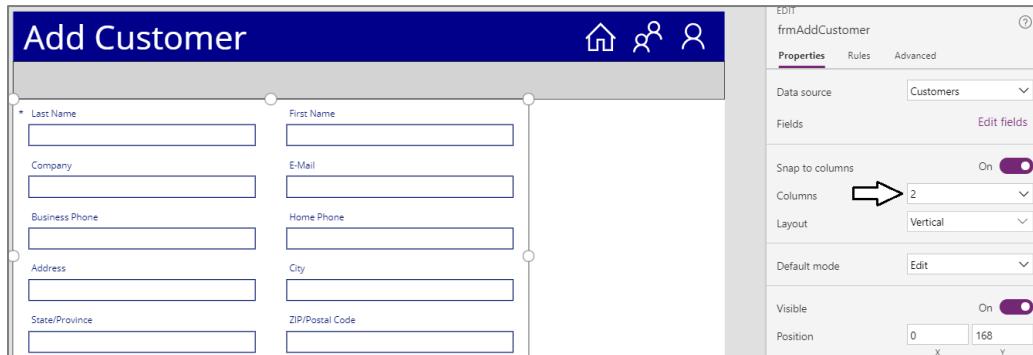
- d) Select **Customers** from the **Select a data source** dropdown menu.



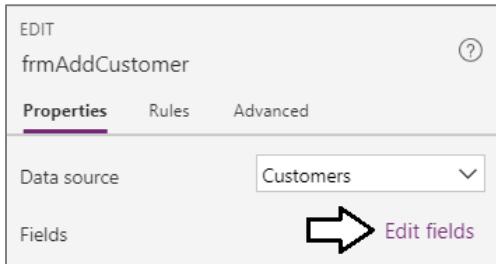
- e) Once you have set the connection, the edit form should display three columns of data cards



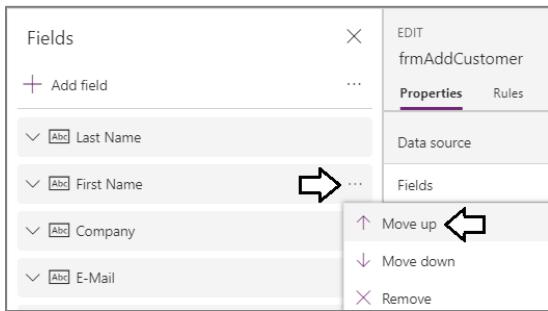
- f) Change the **Columns** property of the edit form from **3** to **2** so that the data cards are laid out in 2 columns.



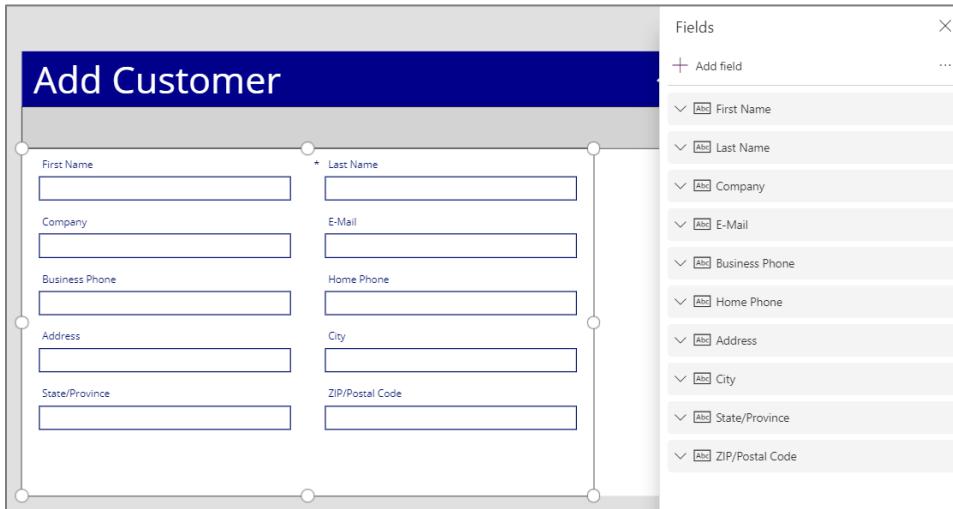
- g) Click the **Edit fields** link in the **Property** pane to display the **Fields** pane.



- h) Drop down the context menu for the **First Name** column and select **Move Up** to place it above the **Last Name** column.

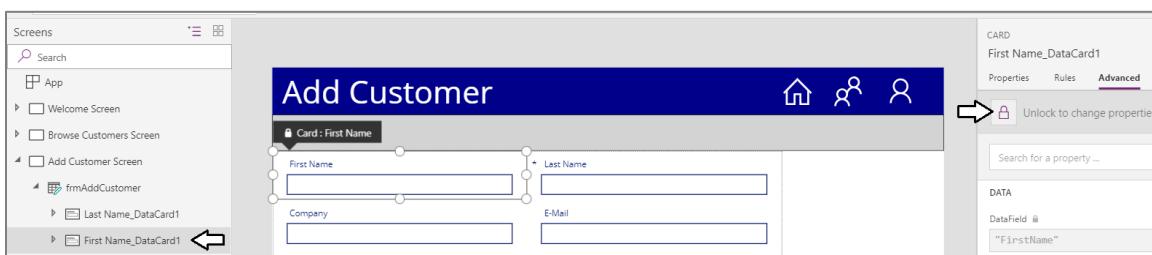


- i) The edit form on the **Add Customer Screen** and the **Fields** pane should match the following screenshot.

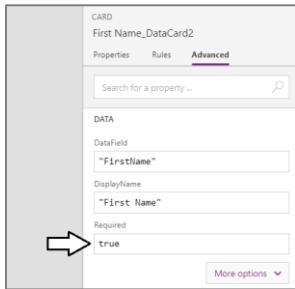


3. Set the **First Name** field to be a required field.

- a) Select the **First Name** data card and then click **Unlock to change properties** from the **Advanced** tab to unlock the data card.



- b) Set the **Required** property of the **First Name** data card to **true**.

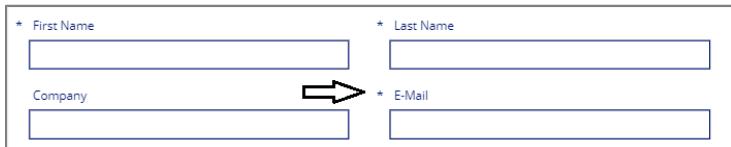


- c) The **First Name** data card should display an asterisks just like the **Last Name** data card indicating that it is a required field.



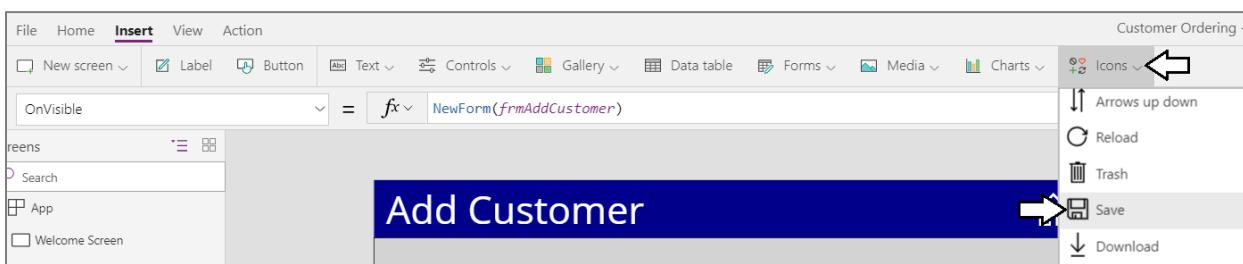
4. Set the **E-Mail** field to be a required field.

- a) Select the **E-Mail** data card and then click **Unlock to change properties** from the **Advanced** tab to unlock the data card.  
 b) Set the **Required** property of the **First Name** data card to **true**.  
 c) The **E-Mail** data card should display an asterisks just like the **Last Name** data card and the **First Name** data cards.

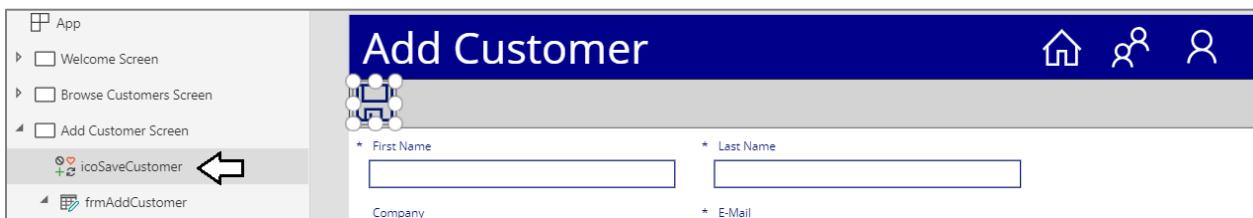


5. Add a save icon button to the toolbar

- a) Add a new **Save** icon to the **Add Customer Screen**.



- b) Rename the icon to **icoSaveCustomer** and position it on the toolbar as shown in the following screenshot.



6. Add behavior to the **Add Customer Screen** to create and save new customer items.
- Select the **Add Customer Screen** in the left tree view.
  - Update the **OnVisible** property of the **Add Customer Screen** with the following expression.

```
NewForm(frmAddCustomer)
```

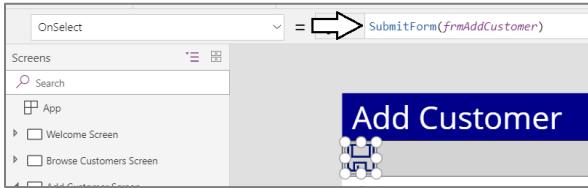
- The formula bar for the **OnVisible** property should match the following screenshot.



- Update the **OnSelect** property of **icoSaveCustomer** with the following expression.

```
SubmitForm(frmAddCustomer)
```

- The formula bar for the **OnSelect** property should match the following screenshot.



- Update the **DisplayMode** property of **icoSaveCustomer** with the following expression.

```
If(frmAddCustomer.Valid, DisplayMode.Edit, DisplayMode.Disabled)
```

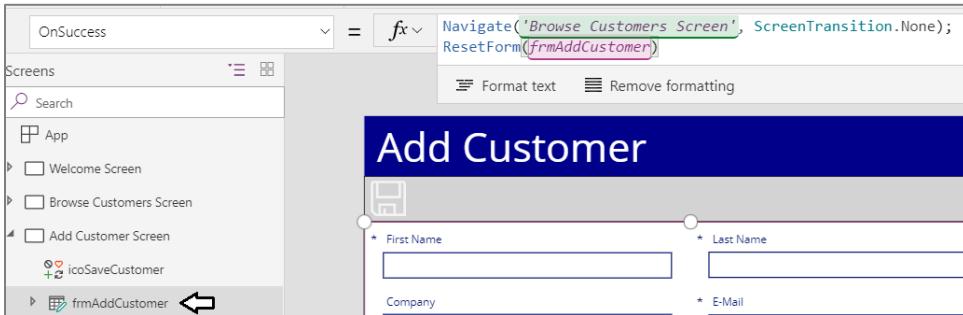
- The formula bar for the **DisplayMode** property should match the following screenshot.



- Set the **OnSuccess** property of **frmAddCustomer** using the following formula.

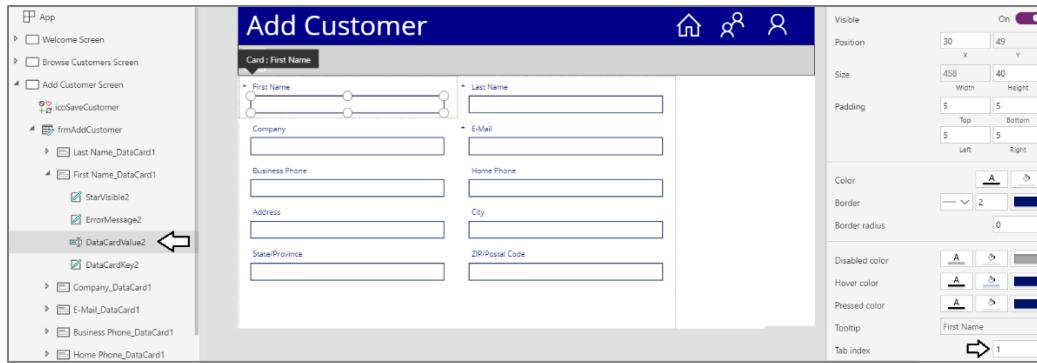
```
Navigate('Browse Customers Screen', ScreenTransition.None);  
ResetForm(frmAddCustomer)
```

- The formula bar for the **OnSuccess** property should match the following screenshot.



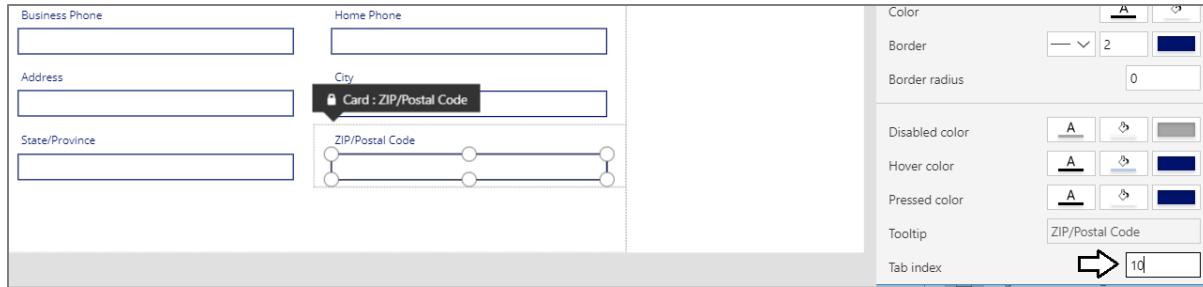
7. Set tab order for the textboxes on **frmAddCustomer**.

- a) Select the value textbox for **First Name** data card and update its **Tab index** to **1** at the bottom of the **Properties** pane.



You will now assign a Tab index value to each textbox in the edit form.

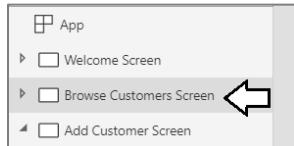
- b) Update the **Tab index** of value textbox of the **Last Name** data card to a value of **2**.
- c) Update the **Tab index** of value textbox of the **Company** data card to a value of **3**.
- d) Update the **Tab index** of value textbox of the **E-Mail** data card to a value of **4**.
- e) Update the **Tab index** of value textbox of the **Business Phone** data card to a value of **5**.
- f) Update the **Tab index** of value textbox of the **Home Phone** data card to a value of **6**.
- g) Update the **Tab index** of value textbox of the **Address** data card to a value of **7**.
- h) Update the **Tab index** of value textbox of the **City** data card to a value of **8**.
- i) Update the **Tab index** of value textbox of the **State/Province** data card to a value of **9**.
- j) Update the **Tab index** of value textbox of the **ZIP/Postal Code** data card to a value of **10**.



You should now have established a tab order to move through the form using the TAB key.

8. Test the **Add Customer Screen** to verify you can add new customer items to the SharePoint **Customers** list.

- a) Select the **Browser Customers Screen** in the left tree view.



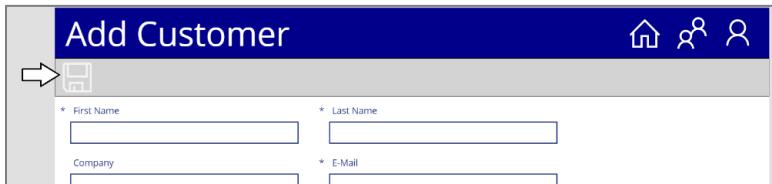
- b) Click the Play button in the upper, right corner to start the app.



- c) Click the **Person** icon in the Top Nav Banner to navigate to the **Add Customer Screen**.



- d) Note that the **Save** button should be initially disabled because the required fields are empty.



- e) Add values for the **First Name**, **Last Name** and **Company**.  
f) The **Save** button should still be disabled because the required field named E-Mail is still empty.



- g) Add a value for the **E-Mail** and notice that the **Save** button becomes enabled because all required fields have been added.



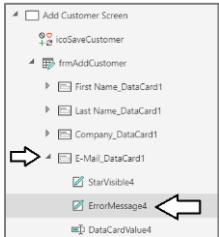
- h) Click the **Save** button to save the new customer and return to the **Browse Customers Screen**.



- i) You should be able to verify that a new customer item has been created.  
j) Stop the app from running and return to edit mode in PowerApps Studio.

While you have added validation to the edit form which requires the user to enter a value for the **E-Mail** field, the form does not provide any validation to ensure that the email address is properly formatted. You will now extend the edit form with additional validation constraints which ensure the user enters an email address that is well formed.

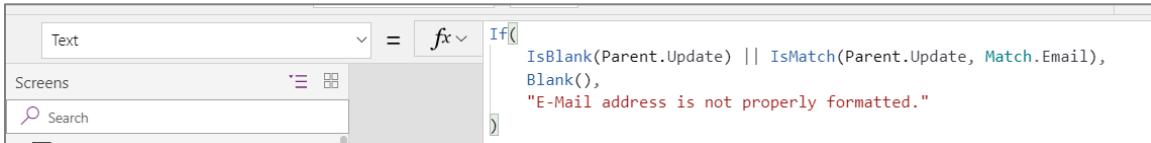
9. Add extra validation support to require a well-formed email address.
  - a) Expand the **Add Customer Screen** in the left tree view.
  - b) Expand the data card for the **E-Mail** field and locate the label with the error message as shown in the following screenshot.



- c) Set the **Text** property of the error message label to the following expression.

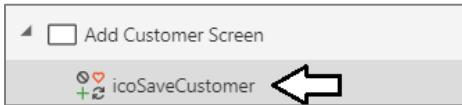
```
If(
    IsBlank(Parent.Update) || IsMatch(Parent.Update, Match.Email),
    Blank(),
    "E-Mail address is not properly formatted."
)
```

- d) The formula bar for the **Text** property should match the following screenshot.



In addition to setting the **Text** property of the error message label, you must also modify the expression which determines when to move the **Save** button between an enabled state and a disabled state.

- e) In the left tree view, select **icoSaveCustomer**.



- f) Modify the **DisplayStyle** property of **icoSaveCustomer** with the following expression to validate the email is well formed.

```
If(
    frmAddCustomer.Valid && IsMatch(frmAddCustomer.Updates.'E-Mail', Match.Email),
    DisplayMode.Edit,
    DisplayMode.Disabled
)
```

- g) The formula bar for the **Text** property should match the following screenshot.



10. Test the **Add Customer Screen** to verify that the edit form properly validates email addresses.

- Select the **Browser Customers Screen** in the left tree view.
- Click the Play button in the upper, left corner to start the app.



- Click the **Person** icon in the Top Nav Banner to navigate to the **Add Customer Screen**.
- Note that the **Save** button is initially disabled because the required fields are empty.

A screenshot of the 'Add Customer' screen. The title bar says 'Add Customer'. There is a large 'H' icon with an arrow pointing to it. Below the title are four input fields: 'First Name' (empty), 'Last Name' (empty), 'Company' (empty), and 'E-Mail' (empty).

First Name	Last Name	Company	E-Mail

- Add values for the **First Name**, **Last Name** and **Company**.
- The **Save** button should still be disabled because the required field named **E-Mail** is still empty.
- Enter an improperly-formatted email address. The error message should appear and the **Save** button should be disabled.

A screenshot of the 'Add Customer' screen. The title bar says 'Add Customer'. The 'First Name' field contains 'Bob', 'Last Name' contains 'Barker', and 'Company' contains 'The Price Is Right'. The 'E-Mail' field contains 'bob'. A red error message 'E-Mail address is not properly formatted.' is displayed next to the 'E-Mail' field.

First Name	Last Name	Company	E-Mail
Bob	Barker	The Price Is Right	bob

- Enter a valid email address. The error message should now disappear and the **Save** button should become enabled.

A screenshot of the 'Add Customer' screen. The title bar says 'Add Customer'. The 'First Name' field contains 'Bob', 'Last Name' contains 'Barker', and 'Company' contains 'The Price Is Right'. The 'E-Mail' field contains 'bob@bobbarker.com'. The error message from the previous step is no longer present.

First Name	Last Name	Company	E-Mail
Bob	Barker	The Price Is Right	bob@bobbarker.com

- Click the **Save** button to save the new customer and return to the **Browse Customers Screen**.
- Stop the app from running and return to edit mode in PowerApps Studio.

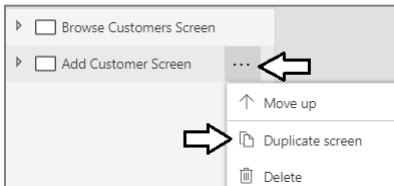
You have now fully implemented the functionality and validation for the edit form used in the **Add Customer Screen**. In the next exercise you will leverage this work by duplicating the **Add Customer Screen** to create the **Edit Customer Screen**.

## Exercise 5: Implement the Edit Customer Screen

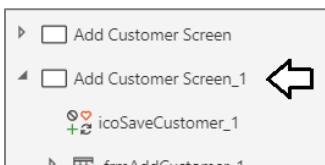
In this exercise, you will duplicate the **Add Customer Screen** to provide a starting point for the **Edit Customers Screen**.

- Duplicate the **Add Customer Screen** to create the **Edit Customer Screen**.

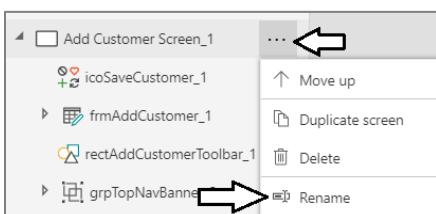
- Drop down the context menu for the **Add Customer Screen** and select the **Duplicate screen** command.



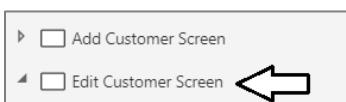
- You should see that the **Add Customer Screen** has been duplicated with a new of **Add Customer Screen\_1**.



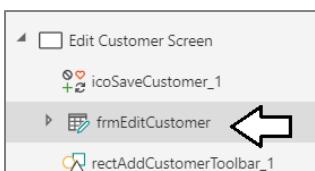
- Drop down the context menu for the new screen and select the **Rename** command.



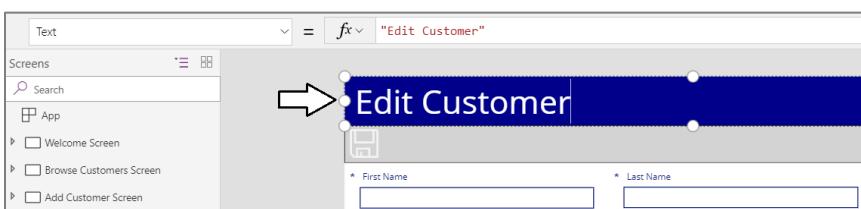
- Rename the new screen **Edit Customer Screen**.



- Rename the edit form inside the **Edit Customer Screen** to **frmEditCustomer**.



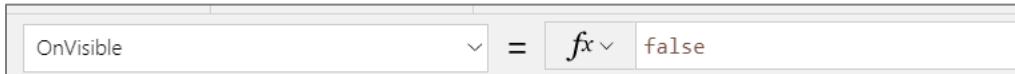
- Update the **Text** property for the label in the Top Nav Banner to read **Edit Customer** instead of **Add Customer**.



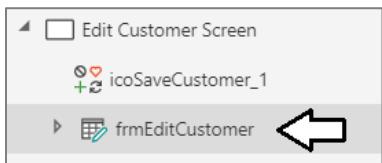
2. Update the expressions in the **Edit Customer Screen** to support editing existing items instead of creating new items.
- Examine the **OnVisible** property of the **Edit Customer Screen** which calls the **NewForm** function.



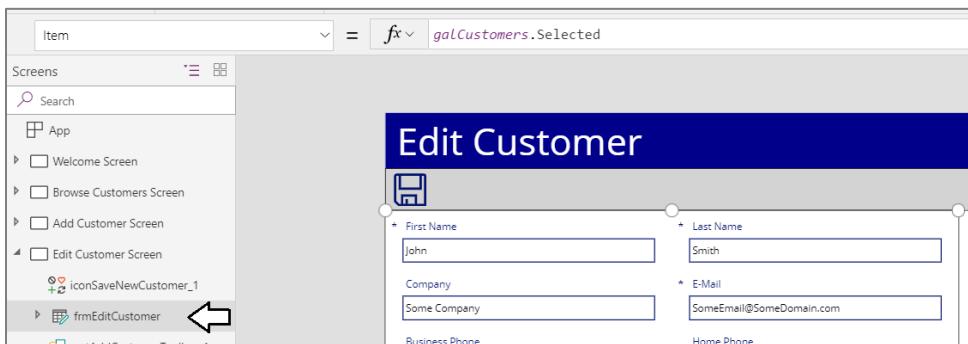
- Examine the **OnVisible** property of the **Edit Customer Screen** which calls the **NewForm** function.
- Remove the expression from the **OnVisible** property of the **Edit Customer Screen** by setting its value to **false**.



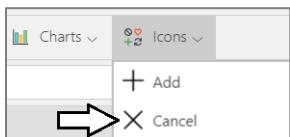
- Examine the **OnVisible** property of the **Edit Customer Screen** which calls the **NewForm** function.
- Remove the expression from the **OnVisible** property of the **Edit Customer Screen** by setting its value to **false**.
- Select **frmEditCustomer** in the left tree view.



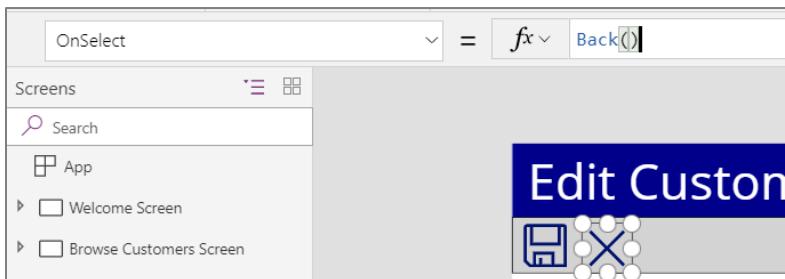
- Examine the **OnVisible** property of the **Edit Customer Screen** which calls the **NewForm** function.
- Remove the expression from the **OnVisible** property of the **Edit Customer Screen** by setting its value to **false**.
- Select **frmEditCustomer** in the left tree view.
- Update the **Item** property of **frmEditCustomer** with the expression **galCustomers.Selected**.



3. Add a **Cancel** icon button to the toolbar to which returns the user to the **Browse Customers Screen** without saving changes.
- Using the **Icons** menu, add a new **Cancel** icon to the **Edit Customer Screen** and place it on the toolbar.



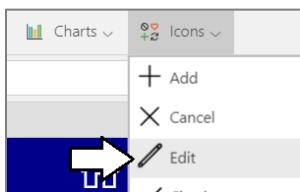
- Using the **Icons** menu, add a new **Cancel** icon to the **Edit Customer Screen** and place it on the toolbar.
- Rename the new icon to **icoCancel**
- Set the **OnSelect** property of **icoCancel** to **Back()**.



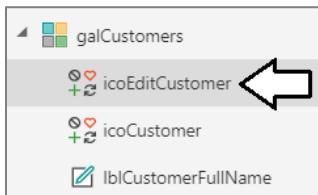
4. Add new icon buttons into the gallery template of **galCustomers** which allow users to edit and delete customers.
  - a) Select **galCustomers** in the left tree view and click the pen icon to enter template edit mode.



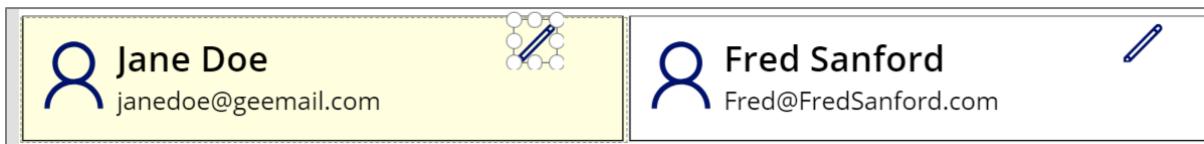
- b) Using the **Icons** menu, add a new **Edit** icon into the gallery template.



- c) Rename the new icon to **icoEditCustomer**.



- d) Set the **Height** property and the **Width** property of **icoEditCustomer** to **32**.
  - e) Position **icoEditCustomer** in the top right corner of the gallery template as shown in the following screenshot.



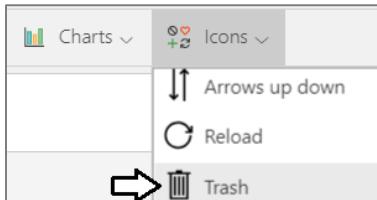
- f) Update the **OnSelect** property of **icoEditCustomer** using the following expression.

```
EditForm(frmEditCustomer);
Navigate('Edit Customer Screen',ScreenTransition.None)
```

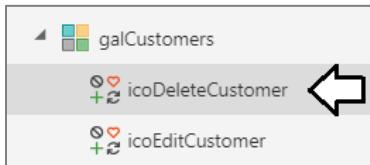
- g) The formula bar for the **OnSelect** property of **icoEditCustomer** should match the following screenshot.



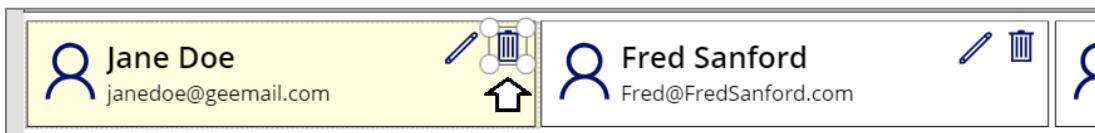
- h) Using the **Icons** menu, add a new **Trash** icon into the gallery template.



- i) Rename the new icon to **icoDeleteCustomer**.



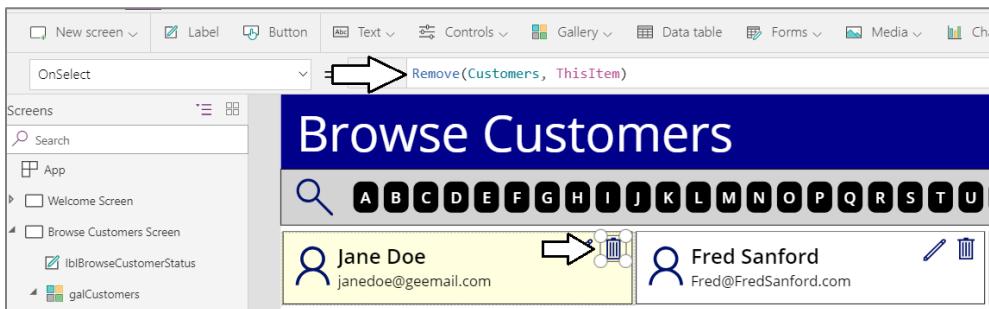
- j) Set the **Height** property and the **Width** property of **icoDeleteCustomer** to **32**.  
k) Position **icoDeleteCustomer** to the right of **icoEditCustomer** as shown in the following screenshot.



- l) Update the **OnSelect** property of **icoDeleteCustomer** using the following expression.

```
Remove(Customers, ThisItem)
```

- m) The layout and the **OnSelect** property for **icoDeleteCustomer** should match the following screenshot.



5. Test the new **Edit** button to verify it allows the user to edit a customer.

- a) Select the **Browser Customers Screen** in the left tree view.  
b) Click the Play button to start the app.



- c) Pick a customer to remove on the **Browser Customers Screen** and then click the edit button.



- d) You should be redirected to the **Edit Customer Screen** with the ability to update columns for the selected customer.

The screenshot shows a form titled "Edit Customer". It contains four input fields: "First Name" (John), "Last Name" (Smith), "Company" (Some Company), and "E-Mail" (SomeEmail@SomeDomain.com). Each field has a validation star (\*) and a placeholder text.

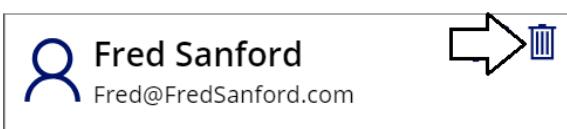
- e) Change the first name of the customer and click the **Save** button to test your work.

The screenshot shows the same "Edit Customer" form. The "First Name" field now contains "Zachary". Two arrows point to the "First Name" field and the "Save" button at the bottom right of the screen.

- f) You should be able to verify that your changes were saved for the current customer.



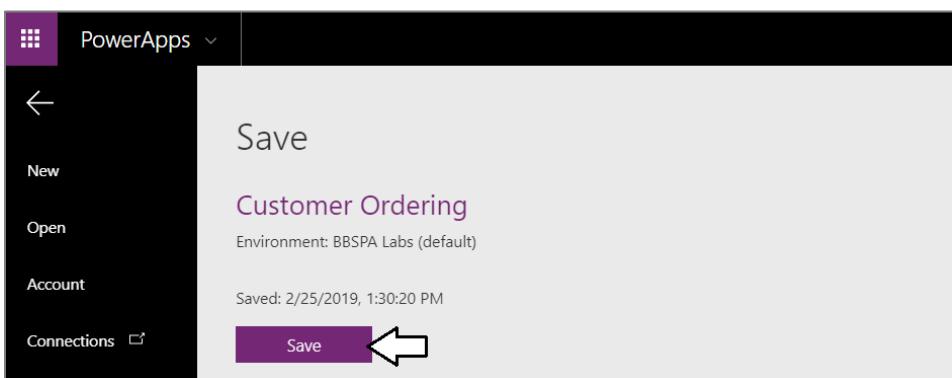
- g) Try to delete a customer using the **Delete** button.



- h) You should be able to verify that the customer has been deleted.

6. Save your work.

- a) Click the **File** menu to move to the backstage area of PowerApps Studio.  
b) Click **Save** in the left navigation and then click the **Save** button on the right to save your work.

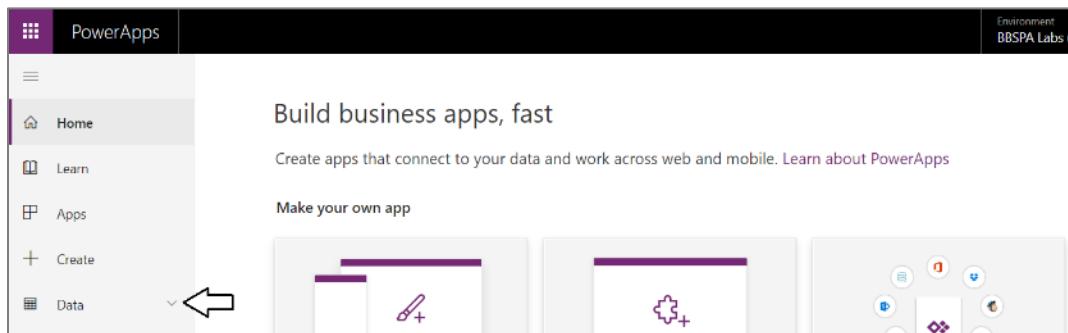


You have now implemented all the functionality required to create, view, edit and delete customer items in a SharePoint list.

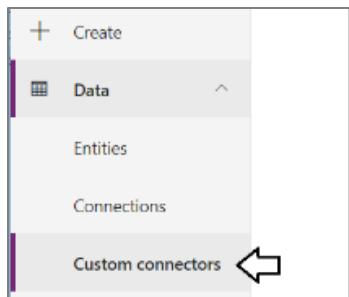
## Exercise 6: Integrate a Custom Connector to Provide New Customer Data

In this exercise, you will create a custom connector which makes it possible to retrieve new customer data from a custom web service on the Internet. After that, you will integrate the custom connector with the **Add Customer Screen** to streamline adding new customers to SharePoint

1. Import a Swagger file to create a new custom connector which provides access to an external web service.
  - a) Open a separate browser tab.
  - b) Navigate to the PowerApps portal at <https://web.powerapps.com>.
  - c) Expand the **Data** section in the left navigation.



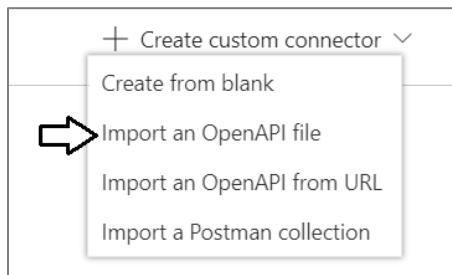
- d) Inside the **Data** section, Click the **Custom connectors** link to navigate to the **Custom connectors** page.



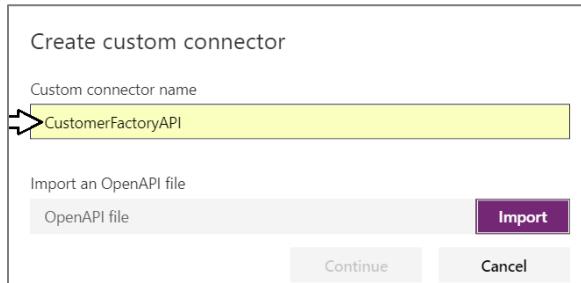
- e) On the **Custom Connectors** page, click **+ Create custom connector**.



- f) Select the **Import an OpenAPI File** command from the **Create customer connector** dropdown menu.



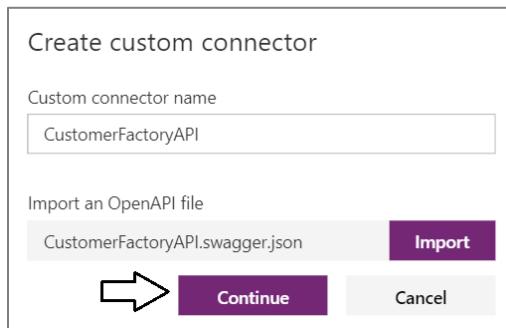
- g) On the **Create custom connector** dialog, enter a **Customer connector name** of **CustomerFactoryAPI**.



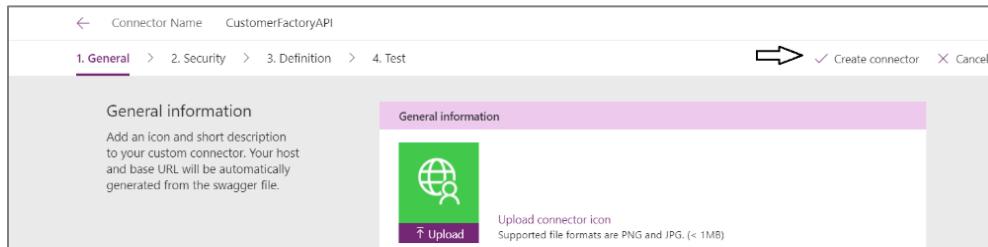
- h) Click the **Import** button and select the file named **CustomerFactoryAPI.swagger.json** which is located at the following path.

C:\Student\Extras\CustomerFactoryAPI.swagger.json

- i) Click the **Continue** button to complete the import process.



- j) Click the **Create connector** button on the left to create the new customer connector named **CustomerFactoryAPI**.

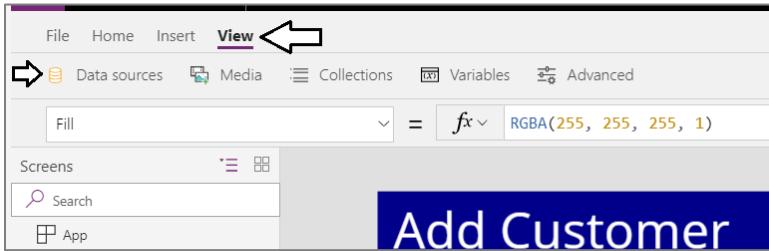


- k) Once the custom connector has been created, you should now see it in on the **Custom connectors** page.



You have now completed the work to create a custom connector. However, you must still create a new connection based on this custom connector. You will do that in an upcoming step while working on the **Customer Ordering** canvas app.

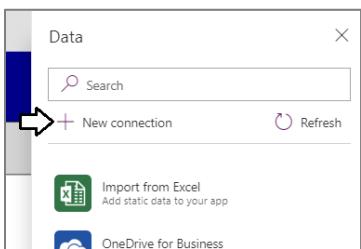
2. Create a new connection based on the **CustomerFactoryAPI** connector.
  - a) Return to the **Customer Ordering** canvas app in PowerApps Studio.
  - b) Select the **Add Customer Screen** in the left tree view.
  - c) Navigate to the **View** tab in the ribbon and click the **Data source** button to display the **Data** pane.



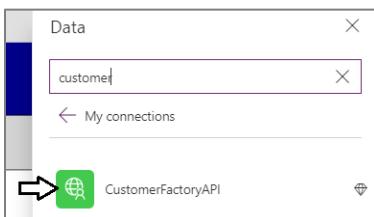
- d) When the **Data** pane appears, click the **+ Add data source** button.



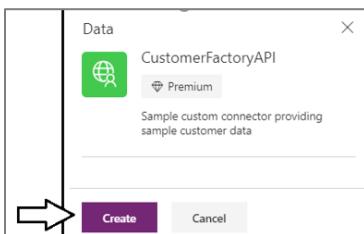
- e) Click the **+ New connection** button to create a new connection.



- f) Type "customer" into the connector search box to find the **CustomerFactoryAPI** connector.
- g) Click on the **CustomerFactoryAPI** connector to select it.



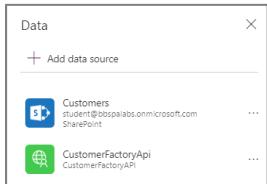
- h) Click the **Create** button to complete the process of creating a new connection for the **CustomerFactoryAPI** connector.



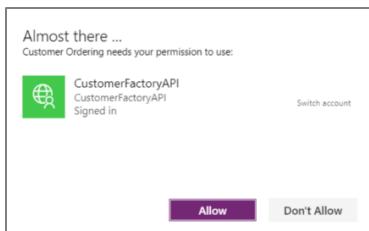
- i) When prompted by the Premium dialog, click **Got it** to continue.



- j) You should now be able to see the new **CustomerFactoryAPI** connection in the **Data** pane.



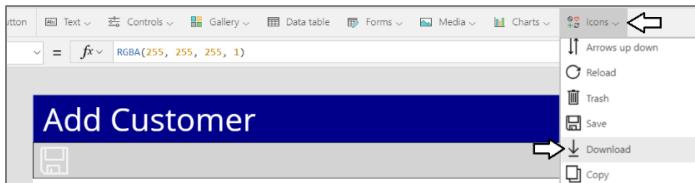
3. Close and reopen the **Customer Ordering** canvas app to authorize the connection for the **CustomerFactoryAPI** connector.
- Save your changes to the **Customer Ordering** canvas app.
  - Close the browse tab with the **Customer Ordering** canvas app.
  - Return to the PowerApps portal at <https://web.powerapps.com> and reopen the **Customer Ordering** canvas app in edit mode.
  - When prompted with the dialog to authorize the **CustomerFactoryAPI** connector, click **Allow**.



The need to close and reopen the **Customer Ordering** Canvas app project is due to a glitch in PowerApps Studio where it does not recognize the need to prompt the user to authorize the new connection. Hopefully this will be fixed soon and closing and then reopening a project in Edit mode will not be required.

4. Add a new button on the toolbar of the **Add Customer Screen** to retrieve new customer data.

- Select the **Add Customer Screen** in the left tree view.
- Use the **Icons** menu to insert a new **Download** icon on the screen.



- Rename the icon to **icoGetNextCustomer**.
- Position **icoGetNextCustomer** on the toolbar to the right of the **Save** icon.



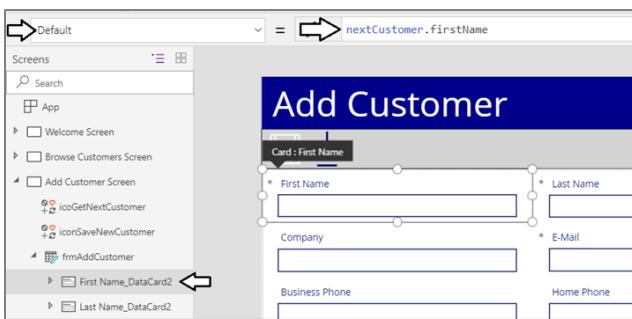
- e) Update the **OnSelect** property of **icoGetNextCustomer** using the following expression.

```
UpdateContext({ nextCustomer: CustomerFactoryApi.NextCustomer() })
```

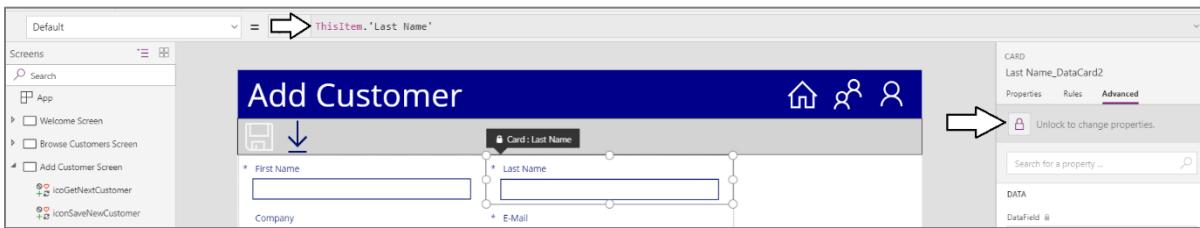
- f) The formula bar for the **OnSelect** property should match the following screenshot.



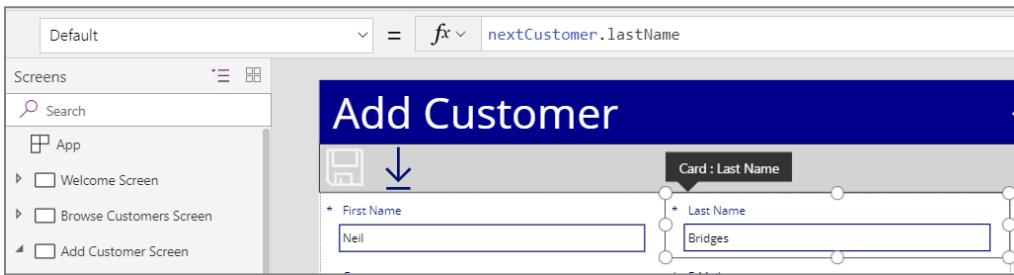
- g) Expand **frmAddCustomer** in the tree view on the left to view its child controls.  
 h) Select the data card for the **First Name** column and update its **Default** property to **nextCustomer.firstName**.



- i) Select the data card for the **Last Name** column and click the **Unlock to change properties** button in the **Advanced** tab.



- j) Update the **Default** property of the **Last Name** data card to **nextCustomer.lastName**.



5. Test the **Add Customer Screen** to make sure the Download button retrieve new customer data.

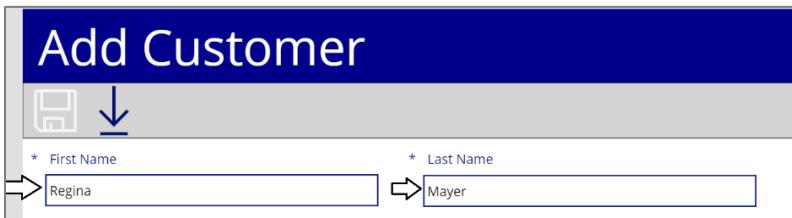
- a) Select the **Add Customer Screen** in the left tree view.  
 b) Click the **Play** button in the upper, left corner to start the app.



- c) Click the **Download** button in the ribbon.



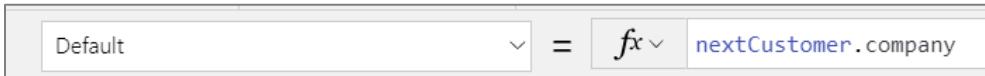
- d) You should see a new customer with a first name and last name each time you click the **Download** button.



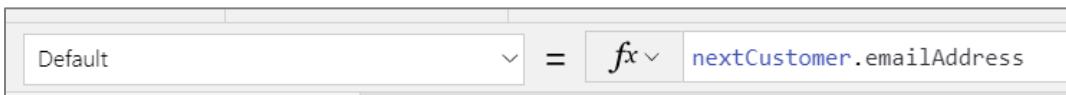
- e) Stop the app from running.

6. Complete the work on the **Add Customer Screen**.

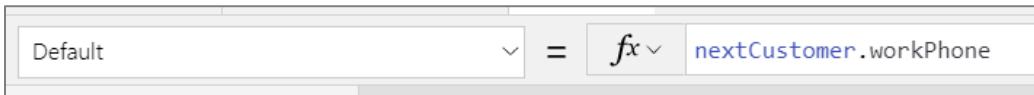
- a) Select the data card for the **Company** column and click the **Unlock to change properties** button in the **Advanced** tab.  
b) Update the **Default** property of the **Company** data card to `nextCustomer.company`.



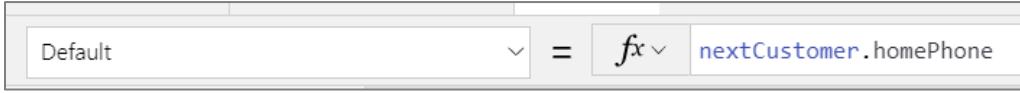
- c) Update the **Default** property of the **E-Mail** data card to `nextCustomer.emailAddress`.



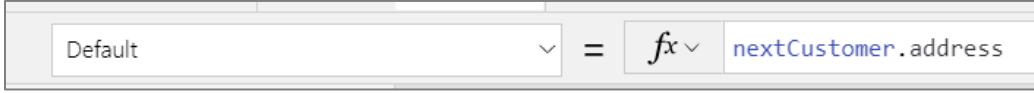
- d) Select the data card for the **Business Phone** column and click the **Unlock to change properties** button in the **Advanced** tab.  
e) Update the **Default** property of the **Business Phone** data card to `nextCustomer.workPhone`.



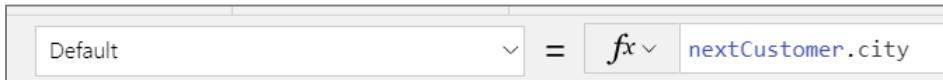
- f) Select the data card for the **Home Phone** column and click the **Unlock to change properties** button in the **Advanced** tab.  
g) Update the **Default** property of the **Home Phone** data card to `nextCustomer.homePhone`.



- h) Select the data card for the **Address** column and click the **Unlock to change properties** button in the **Advanced** tab.  
i) Update the **Default** property of the **Address** data card to `nextCustomer.address`.



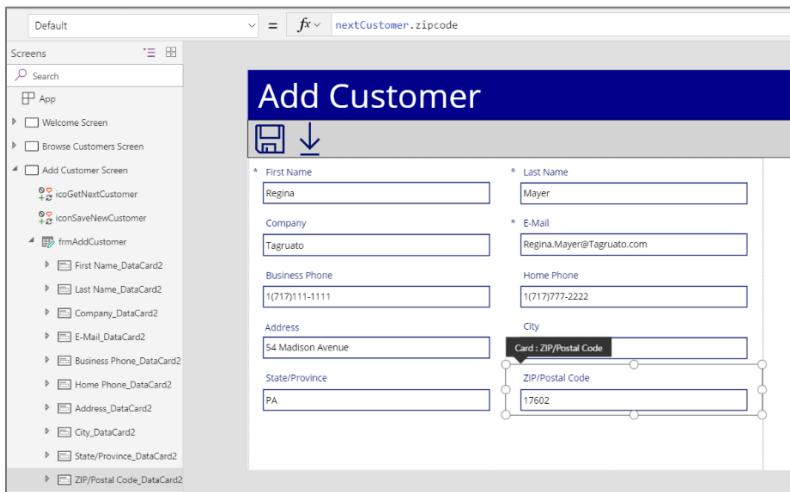
- j) Select the data card for the **City** column and click the **Unlock to change properties** button in the **Advanced** tab.
- k) Update the **Default** property of the **City** data card to **nextCustomer.city**.



- l) Select the data card for the **State/Province** column and click the **Unlock to change properties** button in the **Advanced** tab.
- m) Update the **Default** property of the **State/Province** data card to **nextCustomer.state**.



- n) Select the data card for the **ZIP/Postal Code** column and click the **Unlock to change properties** button in the **Advanced** tab.
- o) Update the **Default** property of the **ZIP/Postal Code** data card to **nextCustomer.zipcode**.
- p) You should now see new customer data for every column in the Customers list.



7. Update the **OnSuccess** property of **frmAddCustomer** to set the **nextCustomer** context variable to a blank value.
  - a) Select **frmAddCustomer** in the left tree view.
  - b) Update the **OnSuccess** property of **frmAddCustomer** using the following expression.

```
Navigate('Browse Customers Screen', ScreenTransition.None);  
ResetForm(frmAddCustomer);  
UpdateContext({nextCustomer: Blank()})
```

8. Update the **OnVisible** property of the **Add Customer Screen**.
  - a) Select the **Add Customer Screen** and update its **OnVisible** property using the following expression.
9. Test the **Add Customer Screen** to make sure the Download button retrieve new customer data.
  - a) Select the **Add Customer Screen** in the left tree view.
  - b) Click the **Play** button in the upper, left corner to start the app.
  - c) Click the **Download** button to retrieve data for a new customer.
  - d) Click the **Save** button to ensure you can save a new customer in SharePoint using customer data from **CustomerFactoryAPI**.

You have now completed this lab.