How to Create a Custom Form and Workflow using PowerApps and Flow

Due to heavy traffic. I am opening up a YouTube Page and will be posting content there too. Thanks so much for reaching out to me about the video and this post. More to come soon!

Youtube LINK

On first glance, PowerApps and Flow seem like excessive product enhancements for SharePoint that many SharePoint Power Users and Developers overlook for a variety of reasons.

There are many confusing tutorials and convoluted solutions out there and, in this series, we are going to work through a practical solution that any organization can utilize to enhance their lines of business with these tools.

THE BUSINESS PROBLEM

Human Resources has a manual checklist process for onboarding new employees.

There is a private HR site where the manual checklist is scanned and placed into a SharePoint Document Library with a folder for each employee to house all their important HR documentation.

The process is cumbersome and because it is a manual process, many steps are often forgotten leaving some new employees without vital information and no tracking details for Human Resources to view where the new employee is in the onboarding process.

THE SOLUTION

We will create a SharePoint list and use an Employee Onboarding Content Type with the site columns necessary to capture our required data.

We will then use Power Apps to create a custom form with custom screens for the Add, Edit, and View pages. These screens will have different views based on the appropriate information we are attempting to track.

Once our form is created, we will use Microsoft Flow to create a workflow to assign a Sequential Case Number for HR to use in tracking, create a new folder in the Document Library with the new Employees Name, create a Word Document in that folder based on a document template, and track the New Employee Onboarding process.

WHAT DO YOU NEED FOR THIS SOLUTION

- SharePoint Online or 2016
- Basic understanding of Content Types and Site Columns
- Power Apps
- Microsoft Flow

WHAT WILL YOU NOT USE FOR THIS SOLUTION

- SharePoint Designer
- Any Custom Code to create forms or workflows

STEP 1: Create a Content Type and its Associated Site Columns

There are many options and locations where you can create Content Types and Site Columns. I prefer to use a Content Type Hub so that if I make any changes to a Content Type or Site Column, I can publish those changes to any lists or libraries in all of my SharePoint tenant site collections (or farms for my on-prem environments).

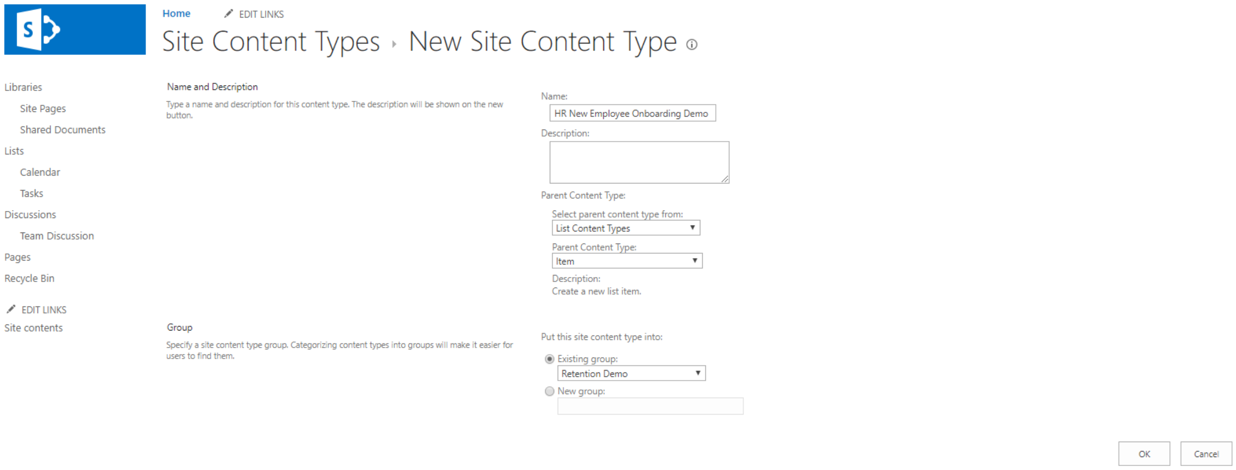
For this solution I strongly recommended creating the Content Types and Site Columns at least at the Site Collection level but the best practice for this solution is to use the Content Type Hub.

If you've never seen your Content Type Hub and use SharePoint Online, visit the following address to find it: https://yourdomain/sites/contenttypehub.

Once on the Content Type Hub we will go to Site Settings and create a new Content Type.

For this project we will call it "HR New Employee Onboarding Demo."

For the Content Type Settings, we will make the Parent Content Type "List Content Types" and ensure we use "Item" to ensure we deploy with only the default columns. We will also place it in a custom group so that it will be easily found.



Site Content Types › New Site Content Type

Name and Description

Type a name and description for this content type. The description will be shown on the new button.

Name: HR New Employee Onboarding Demo

Description:

Parent Content Type:

Select parent content type from: List Content Types

Parent Content Type: Item

Description: Create a new list item.

Group

Specify a site content type group. Categorizing content types into groups will make it easier for users to find them.

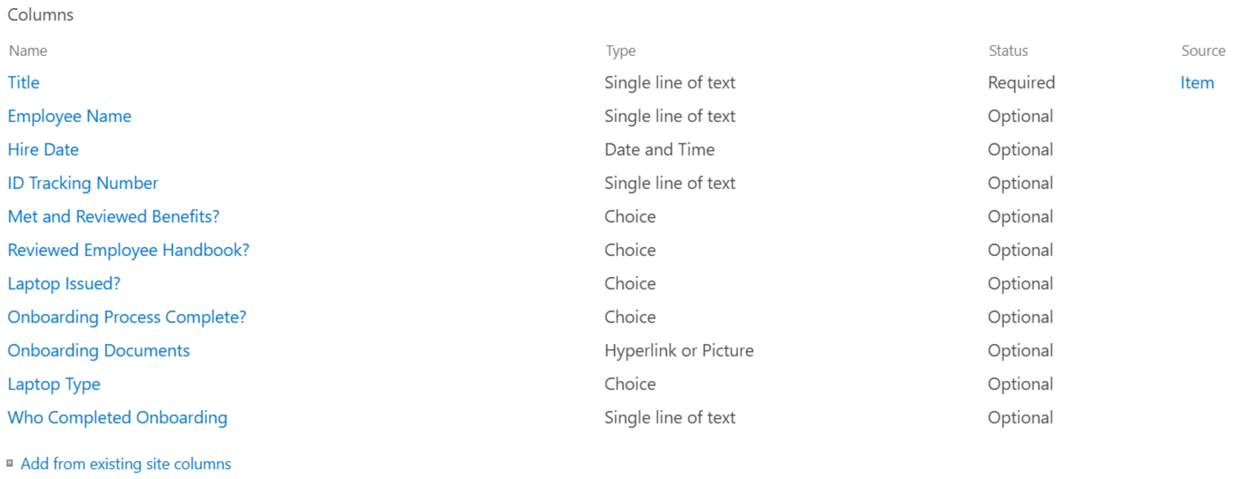
Put this site content type into:

● Existing group: Retention Demo

○ New group:

Now that our Content type is created, we will add the following Site Columns:

- Employee Name(Single line of text column)
- Hire Date(Date and Time column)
- ID Tracking Number(Single line of text column)
- Met and Reviewed Benefits?(Choice column; Yes, No are the choices and default the column to display “No”)
- Reviewed Employee Handbook?(Choice column; Yes, No are the choices and default the column to display “No”)
- Laptop Issued? (Choice column; Yes, No are the choices and default the column to display “No”)
  Laptop Type (Choice column; Dell Laptop, Microsoft Surface, other)
- Onboarding Process Complete? (Choice Column; Yes, No are the choices and default the column to display “No”)
- Who Completed Onboarding (People or Group Column; in this example I have a tenant with only one user but for you, you will want to use a People Picker Column)



Columns

Name	Type	Status	Source
Title	Single line of text	Required	Item
Employee Name	Single line of text	Optional	
Hire Date	Date and Time	Optional	
ID Tracking Number	Single line of text	Optional	
Met and Reviewed Benefits?	Choice	Optional	
Reviewed Employee Handbook?	Choice	Optional	
Laptop Issued?	Choice	Optional	
Onboarding Process Complete?	Choice	Optional	
Onboarding Documents	Hyperlink or Picture	Optional	
Laptop Type	Choice	Optional	
Who Completed Onboarding	Single line of text	Optional	

Add from existing site columns

It will take an hour or two for the Content Type publishes out because you are waiting for a Timer Job to run in SharePoint Administration you have no control over.

To check its status, visit any site collection, navigate to site settings, click Site Content Types and scroll down to your newly created Content Type Group and see if your new Content Type published. If it has, we are ready to proceed to our next phase.

STEP 2: Create a Custom List and Library

For this solution, we need both a custom list and library to fulfill the requirements.

First create your library. We will name it “Employee Onboarding Documents.”

For this solution we do not need a content type or any information but in a live solution you can attach a Content Type, site columns, etc. to it if you want to capture other data associated with your documents.

Add a Word Document as a template to the top level of your library. We will use it to help create our folder in the Microsoft Flow in Step 4. For our demo, I'm calling the .docx file "New Employee Template." For your organization you can add whatever blank document template you need here.

Now, we must create a list to create the custom forms.

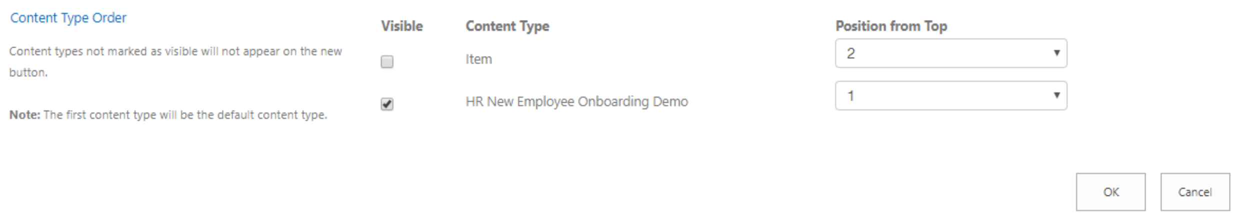
Create your list and name it "Employee Onboarding Form."

Go to your List Settings Menu and ensure the "Allow Management of Content Types" radio button is checked "Yes" in Advanced Settings then click "OK."

Once back on the List Settings screen, scroll down to the Content Type area and click "Add from existing site content types."

Add the new Content Type and click "OK."

Back on the List Settings screen click "Change new button order and default content type." Uncheck item and change your new Content Type to 1 on the "Position from Top" selection and click "OK." This will ensure that any new list item created will default to our new Content Type and all the columns will sync up properly.



Content Type Order

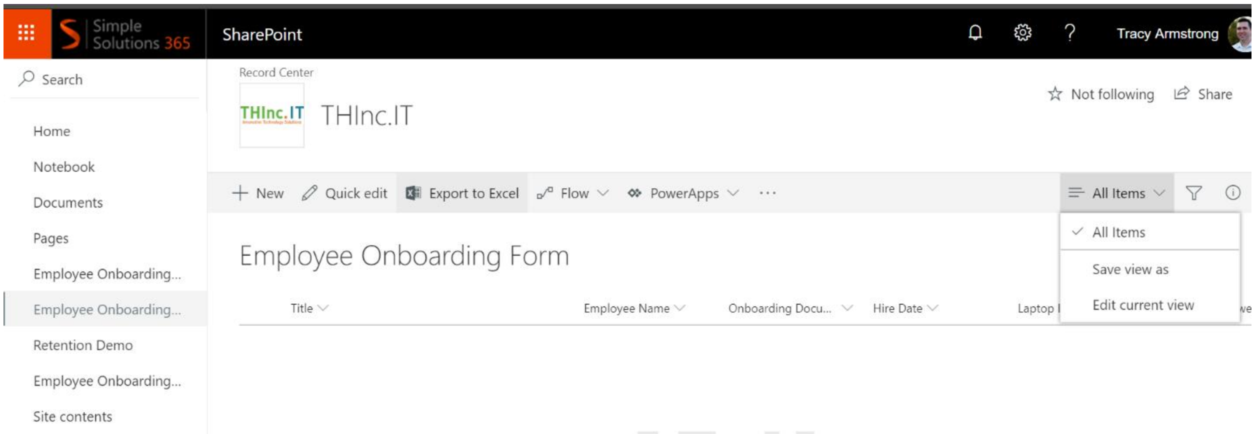
Content types not marked as visible will not appear on the new button.

Note: The first content type will be the default content type.

Visible	Content Type	Position from Top
☐	Item	2
☑	HR New Employee Onboarding Demo	1

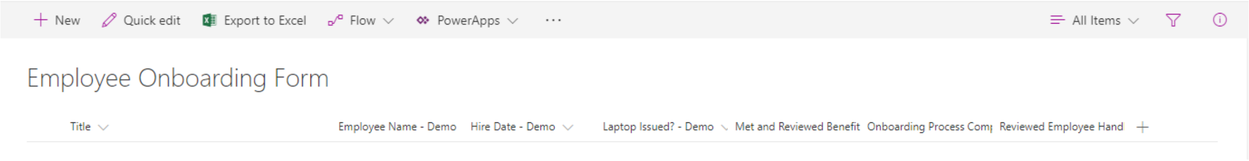
Click back to the main area of the list using the Breadcrumb. You will notice in this view none of your new site columns are showing. We need to edit the view and add the columns so that we can see them easily in our default view.

Click "All Items" and then click "Edit Current View."



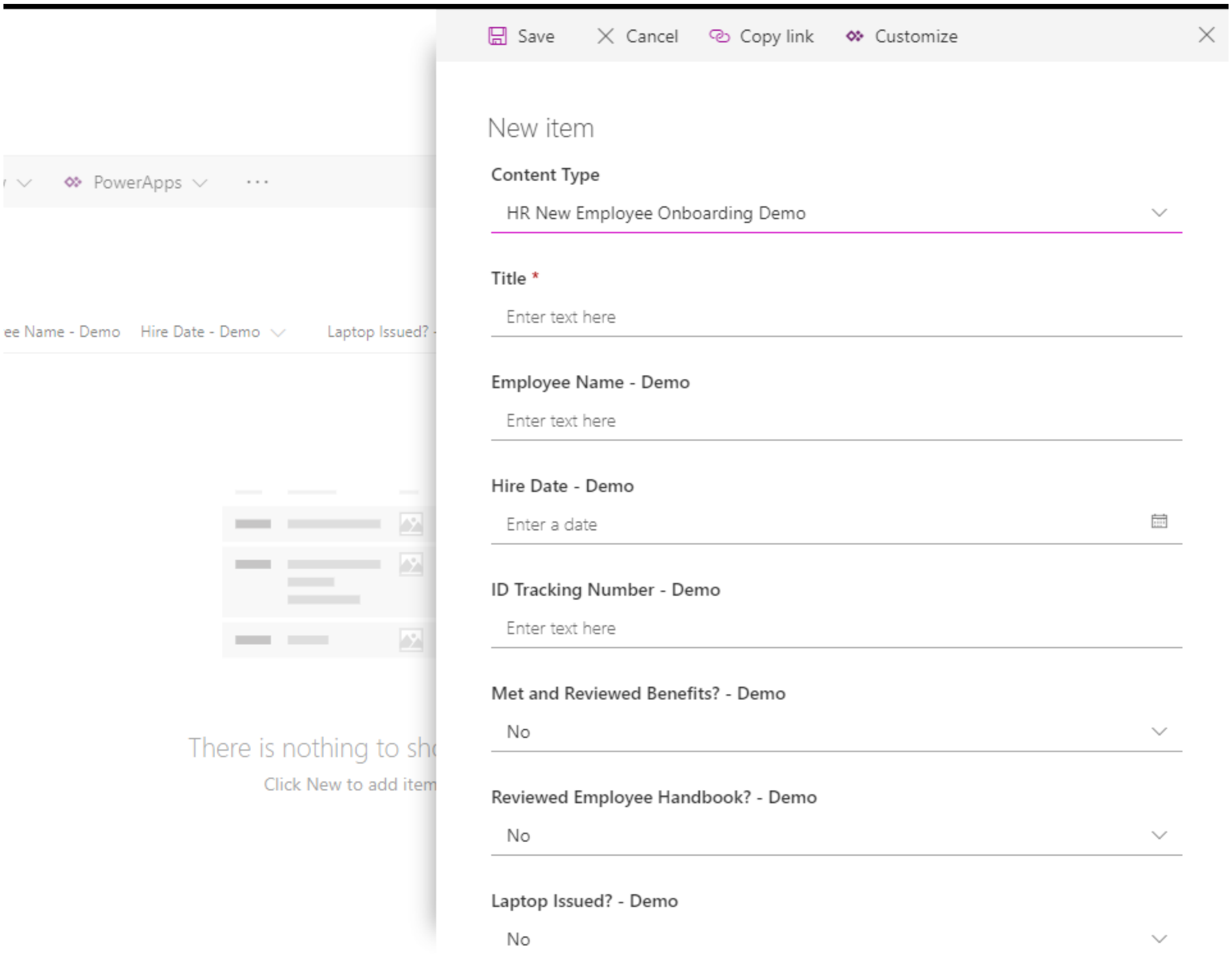
Check the Display boxes and arrange the order of the fields as you see fit. Then click “OK” to save your changes.

You should now see all of your custom columns.



STEP 3: Create Custom Forms Using PowerApps

Normally when we create a new list item it appears like this:



New item

Content Type: HR New Employee Onboarding Demo

Title *: Enter text here

Employee Name - Demo: Enter text here

Hire Date - Demo: Enter a date

ID Tracking Number - Demo: Enter text here

Met and Reviewed Benefits? - Demo: No

Reviewed Employee Handbook? - Demo: No

Laptop Issued? - Demo: No

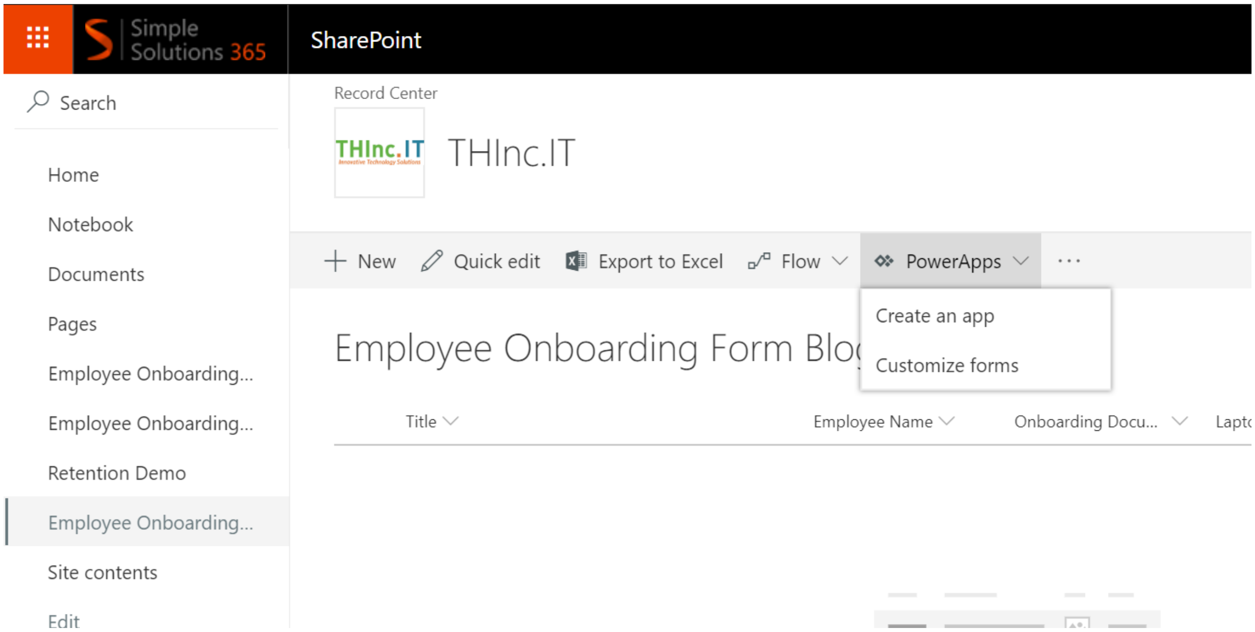
It’s very plain, not custom and shows fields we aren’t necessarily ready to fill out, especially since we need a sequential ID Tracking number!

To create a solution that fits our requirements best, we want to create custom Add, Edit, and View Screens for list items.

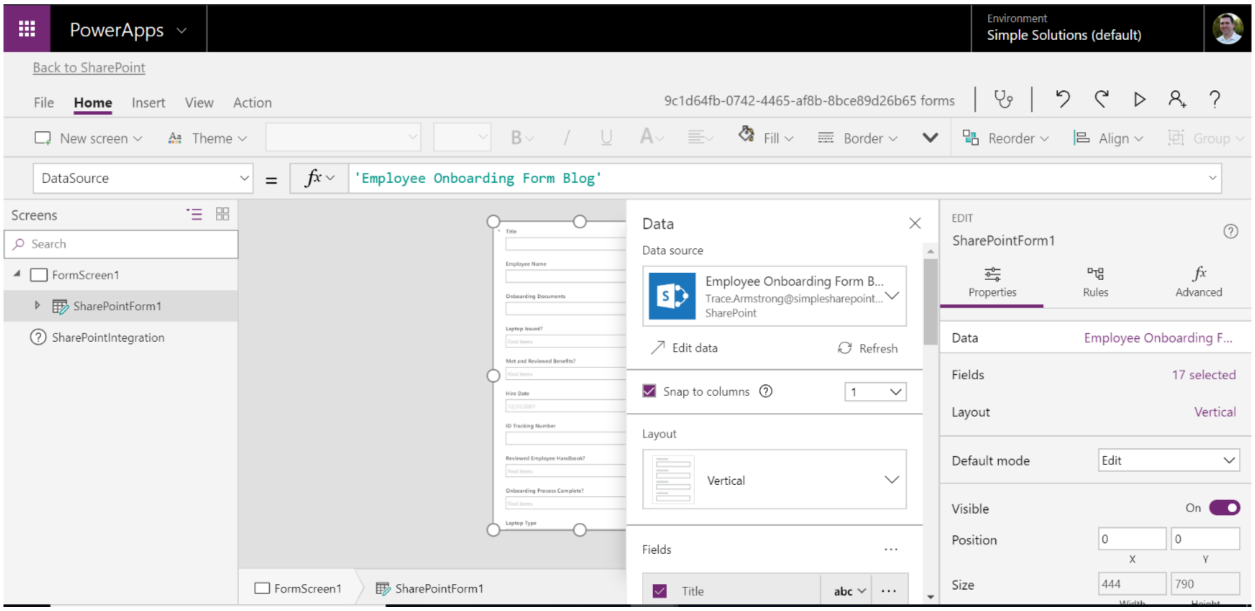
In the past you were forced to use InfoPath (cumbersome) or create a custom form using HTML and JavaScript (great solution but not easily edited in the future).

With PowerApps, we can create a custom form with no custom code to design the form and a little bit of code to tell the list which form to use in what situation.

The first step in creating these screens is to click the PowerApps link on the main list page and choose the “Customize forms” option.



After a few moments PowerApps will build your app and redirect you to the customization screen. It should look like this when the process is complete.

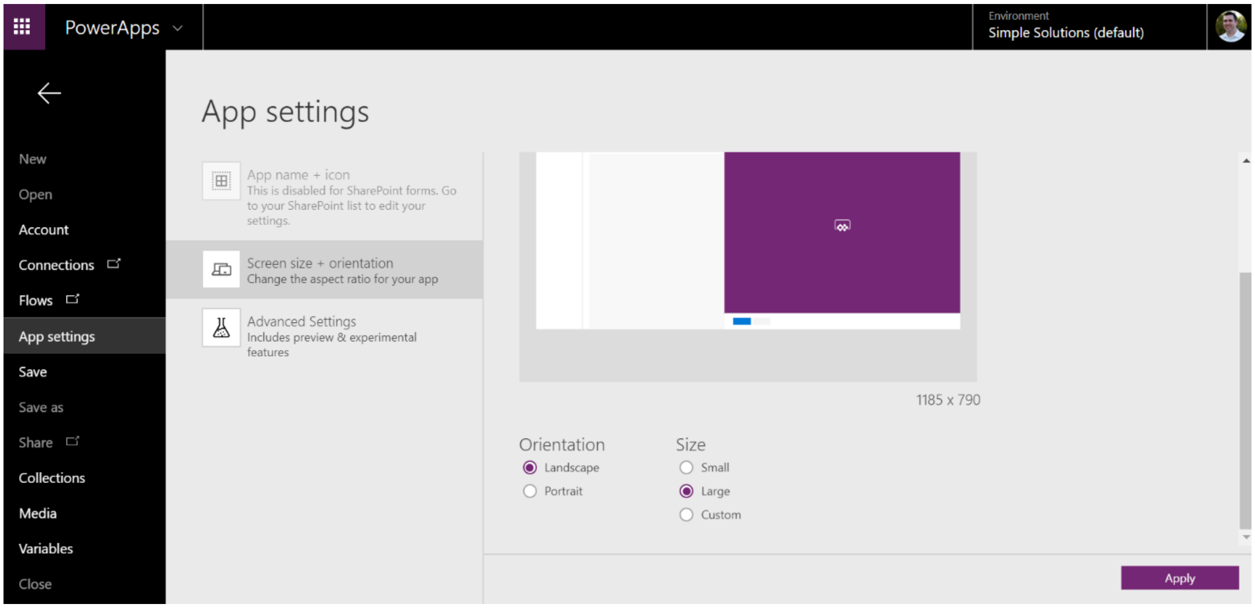


DataSource = 'Employee Onboarding Form Blog'

Data	Employee Onboarding F...
Fields	17 selected
Layout	Vertical
Default mode	Edit
Visible	On
Position	0 X, 0 Y
Size	444 Width, 790 Height

By default, PowerApps will create the form in a Mobile View. However, you are not forced to use a small mobile view. You can create a tablet view that also looks good on Laptop or Desktop displays. For the purposes of this demo we will create a tablet view.

To change the app to a tablet view, click “File”, “App Settings”, and on the “Screen size + orientation” option, change the Orientation to “Landscape” and the Size to “Large.” Then click “Apply.”


PowerApps
Environment
Simple Solutions (default)
New
Open
Account
Connections
Flows
App settings
Save
Save as
Share
Collections
Media
Variables
Close
App settings
App name + icon
This is disabled for SharePoint forms. Go to your SharePoint list to edit your settings.
Screen size + orientation
Change the aspect ratio for your app
Advanced Settings
Includes preview & experimental features
Apply

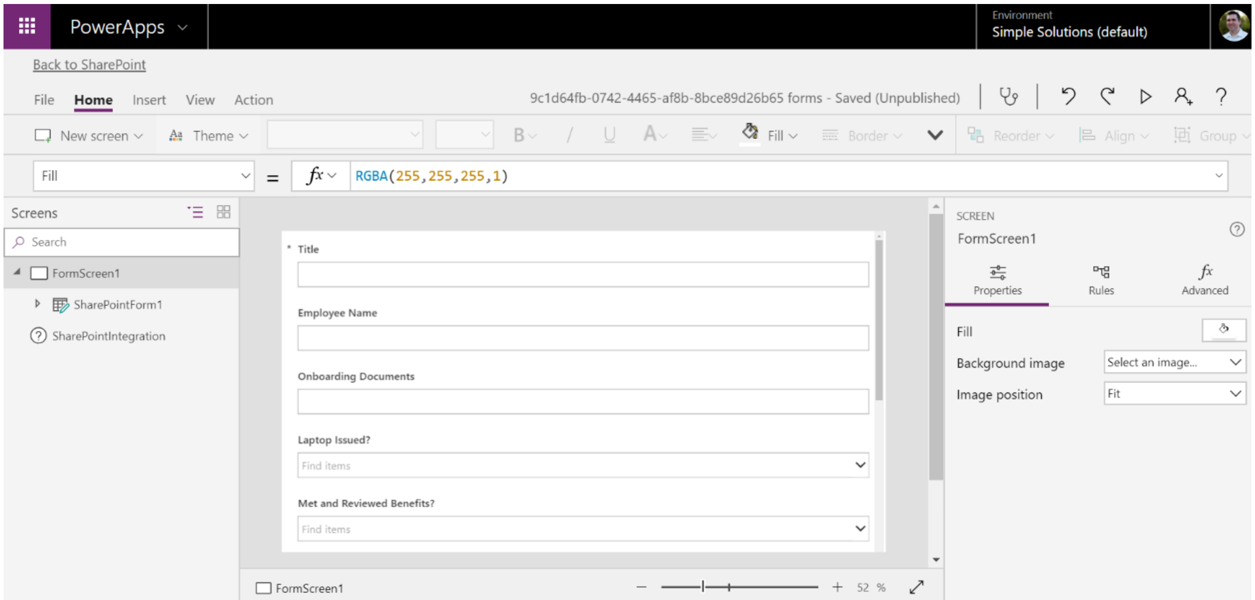
1185 x 790

Orientation
● Landscape
○ Portrait

Size
○ Small
● Large
○ Custom

Once the process completes, click the arrow in the top left of your screen to return to the customization page.

You will now see the form in a landscape view.

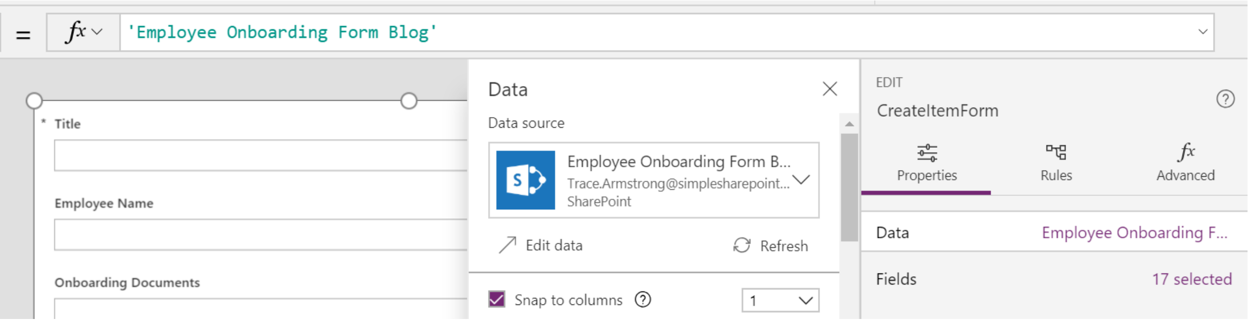


Fill = RGBA(255,255,255,1)

From here we can start customizing the look and feel.

The first thing we want to do is create a form header and arrange the fields because it is much easier to do these before creating multiple screens.

To create a form header, select SharePointForm1 and in the Properties menu click “Data” to get the Data Source Screen.



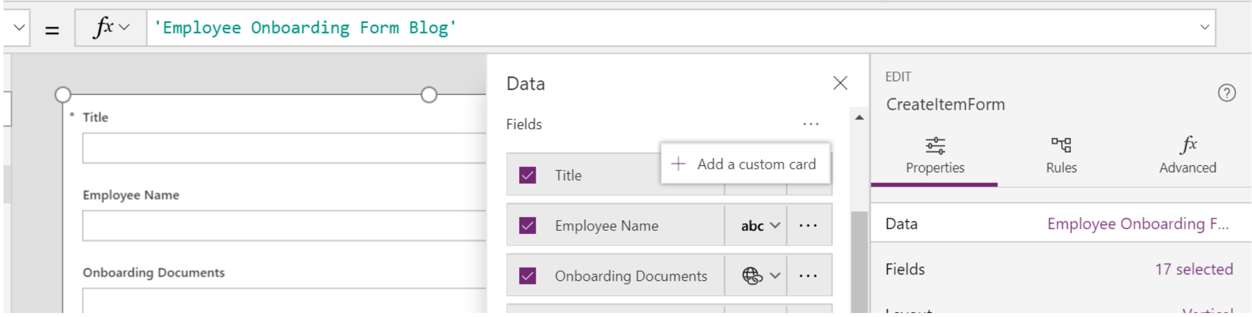
```
'Employee Onboarding Form Blog'
```

☑ Snap to columns: 1

Data: Employee Onboarding F...

Fields: 17 selected

Now scroll down in the Data Menu to the Fields area and click the ellipses to enable the ability to create a custom DataCard.



```
'Employee Onboarding Form Blog'
```

☑ Title

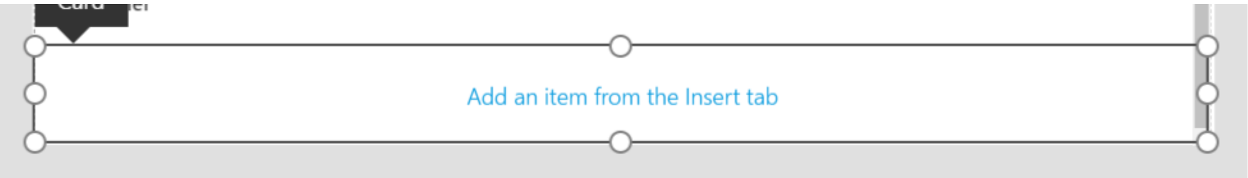
☑ Employee Name

☑ Onboarding Documents

Data: Employee Onboarding F...

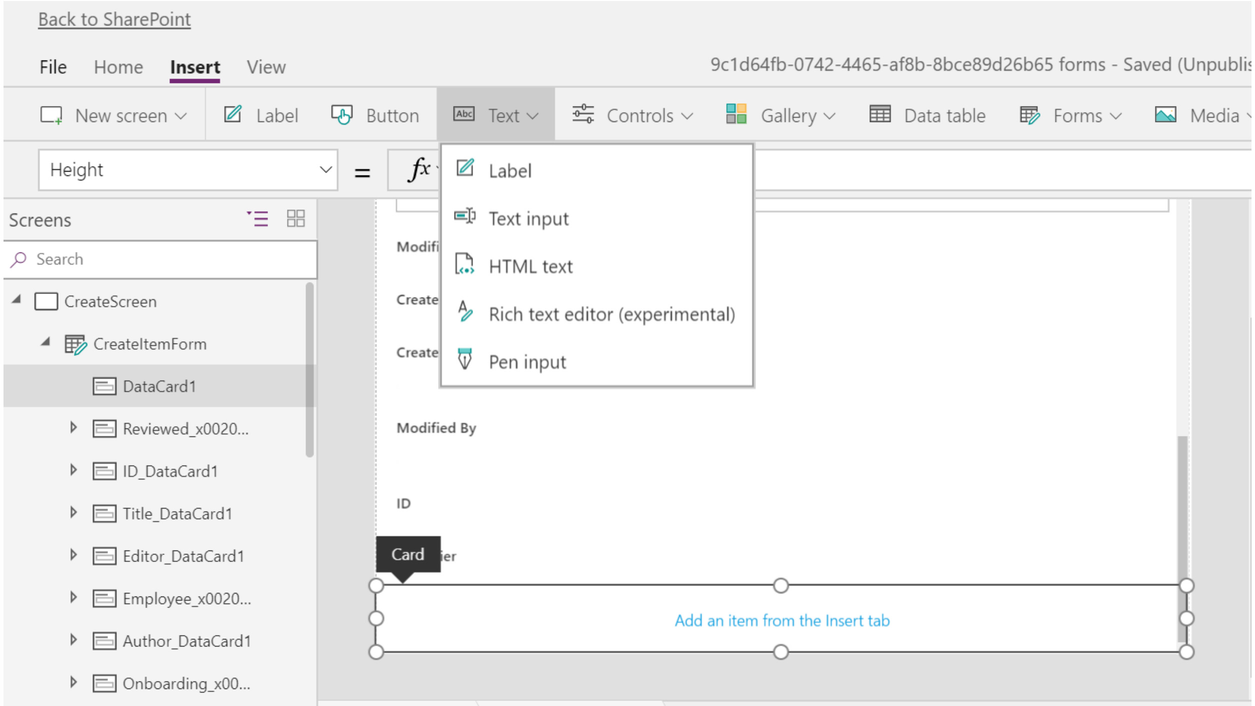
Fields: 17 selected

Close the Data Menu and scroll down on the form screen to the newly created field and select it.

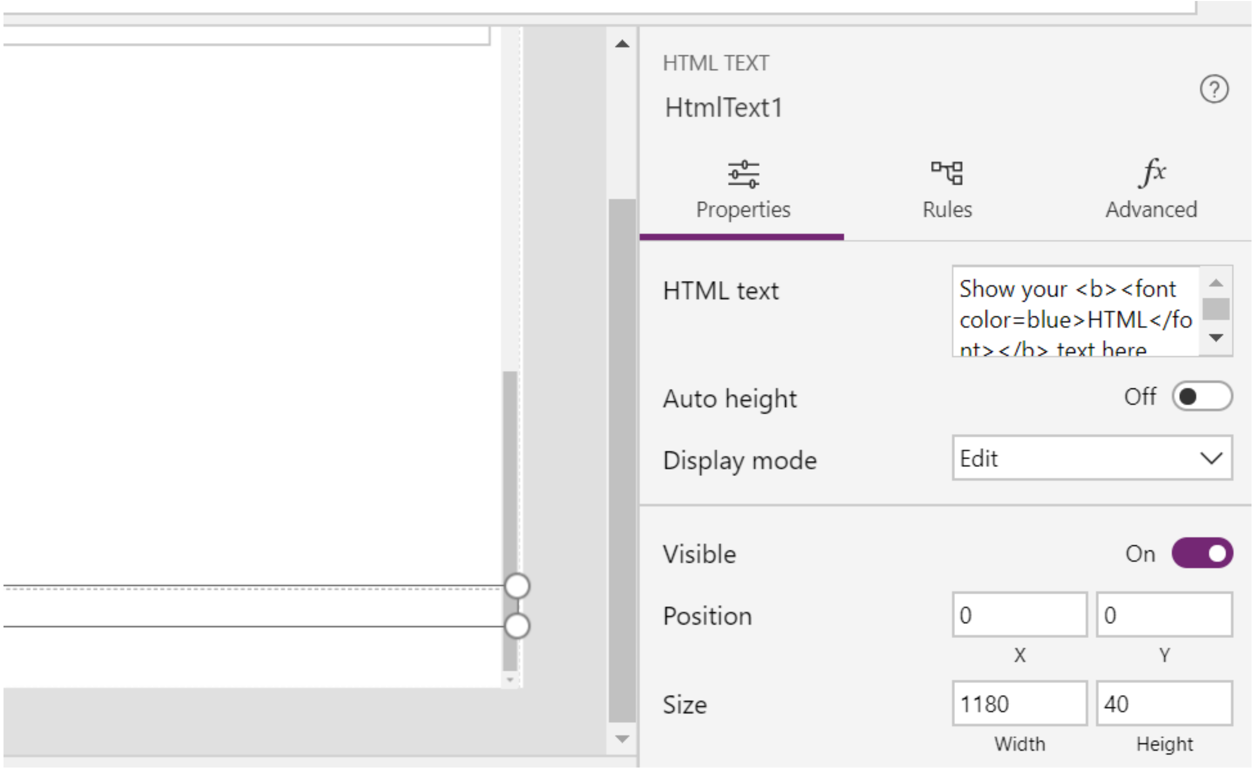


Add an item from the Insert tab

Once selected, click the insert tab and insert an HTML Text box.



The HTML Text box will be automatically selected. Change the width in properties menu to 1180 and the x and y locations to 0.



HTML text: Show your <b><font color=blue>HTML</font></b> text here

Auto height: Off

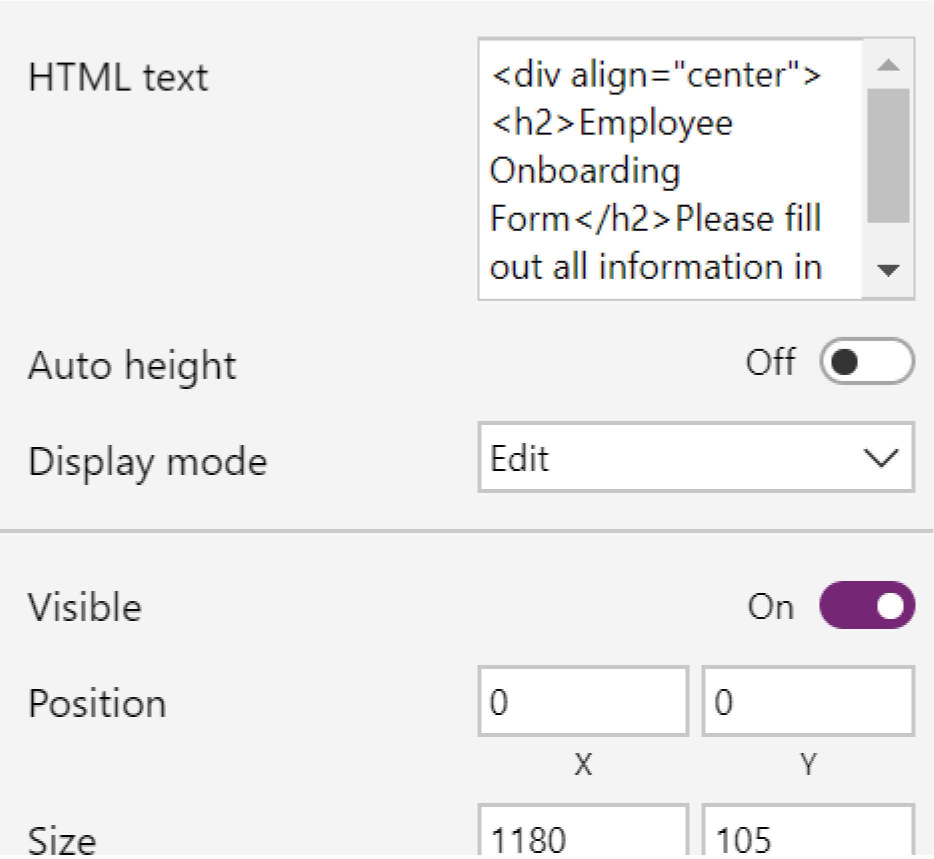
Display mode: Edit

Visible: On

Position: 0 (X) 0 (Y)

Size: 1180 (Width) 40 (Height)

At this point click into HTML Text then add your HTML to the HTML text box.



HTML text: <div align="center"><h2>Employee Onboarding Form</h2>Please fill out all information in

Auto height: Off

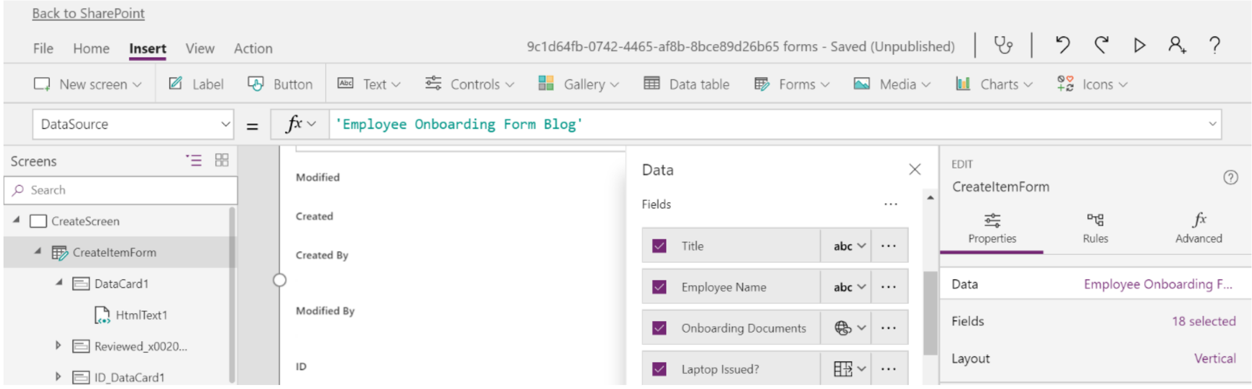
Display mode: Edit

Visible: On

Position: 0 (X) 0 (Y)

Size: 1180 105

Now select SharePointForm1 in the Screens menu and reopen the Data Screen.



DataSource = 'Employee Onboarding Form Blog'

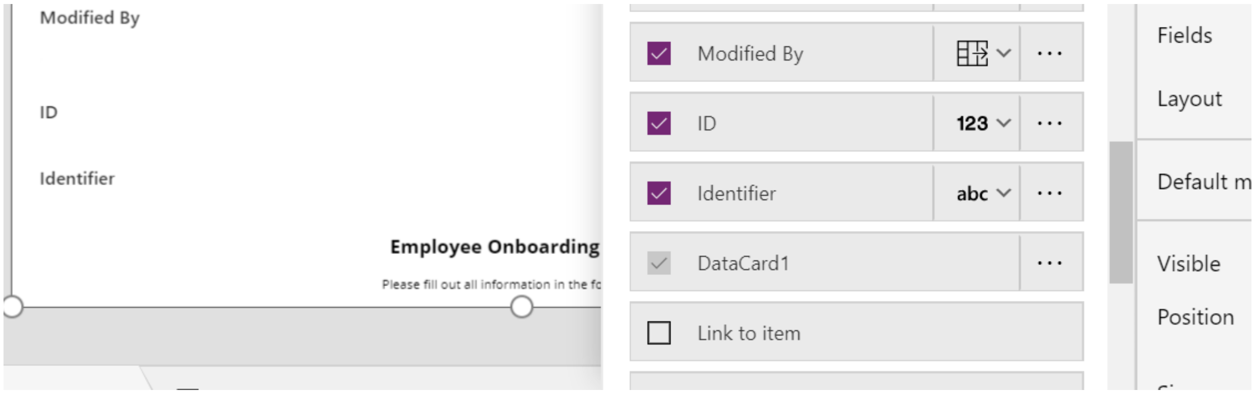
Data

Fields	
☑ Title	abc
☑ Employee Name	abc
☑ Onboarding Documents	
☑ Laptop Issued?	

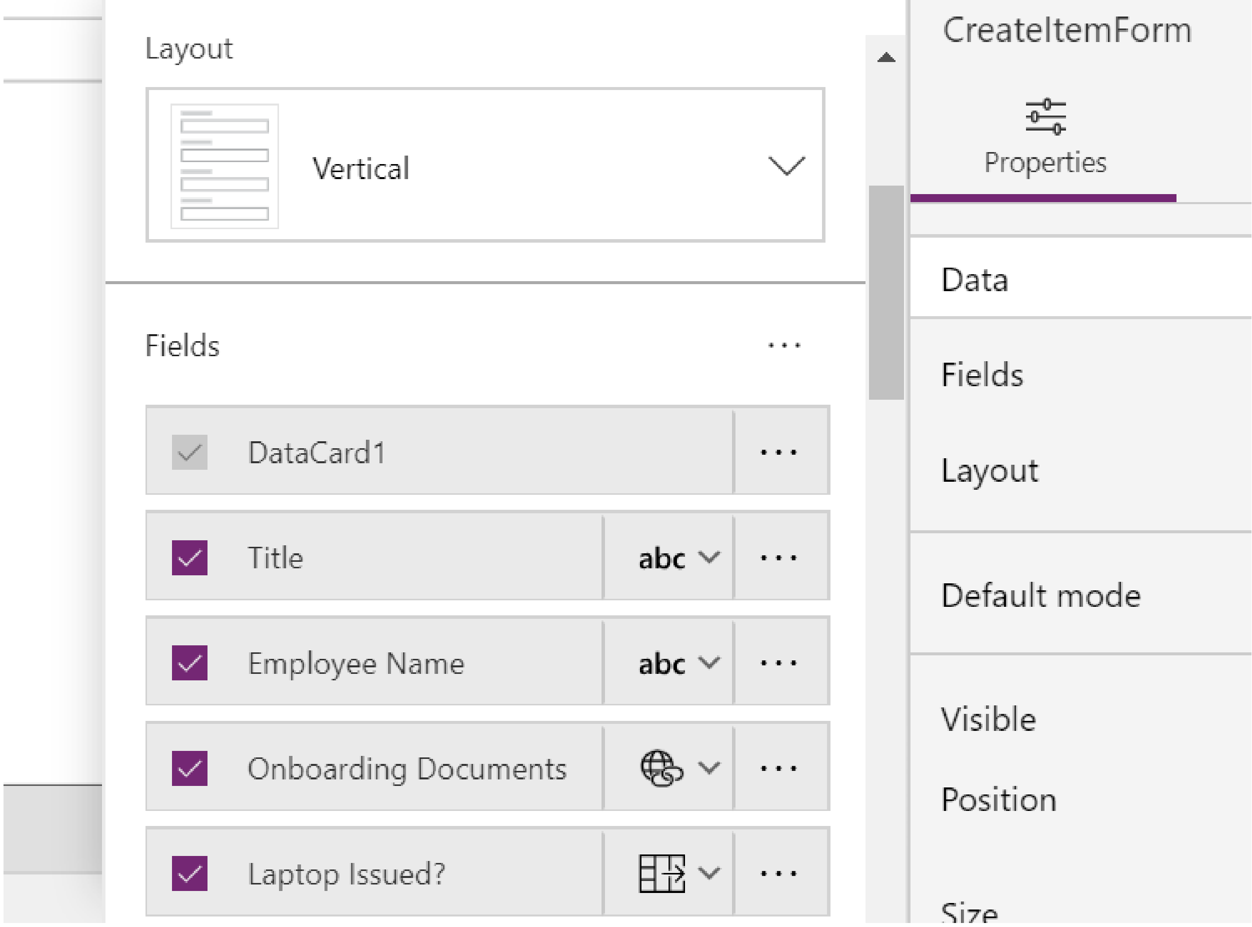
CreateItemForm

Data	Employee Onboarding F...
Fields	18 selected
Layout	Vertical

In the fields area scroll down until you see your new DataCard and drag it to the top of the fields screen.



- ☑ Modified By
- ☑ ID 123
- ☑ Identifier abc
- ✓ DataCard1
- □ Link to item

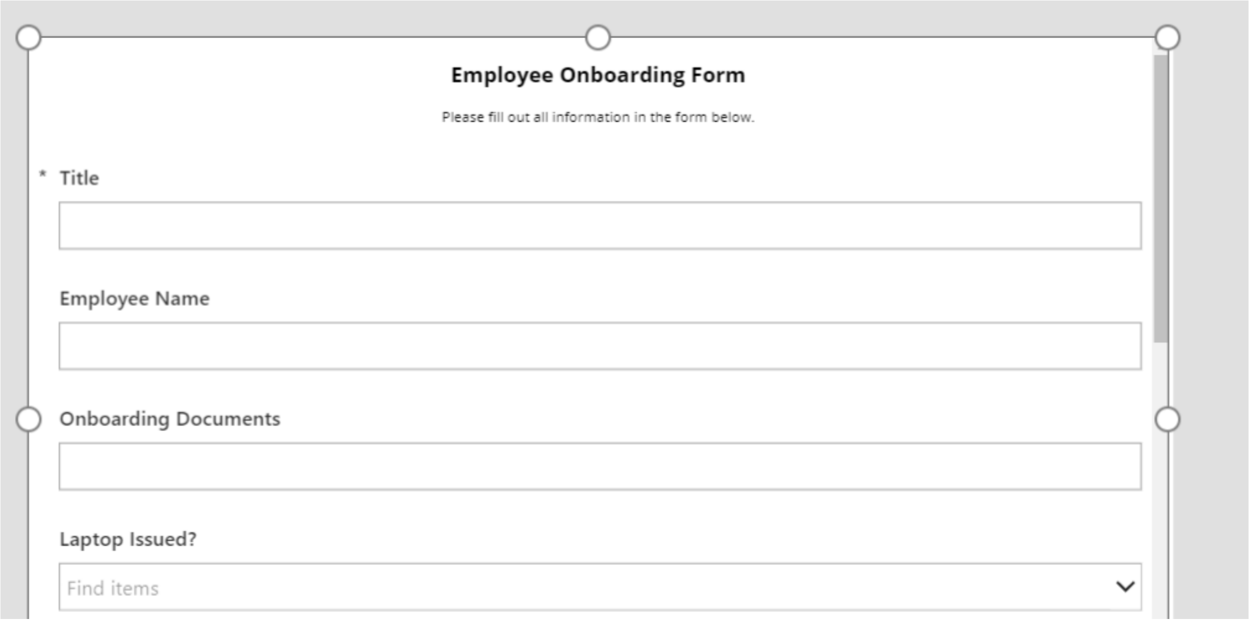


Layout: Vertical

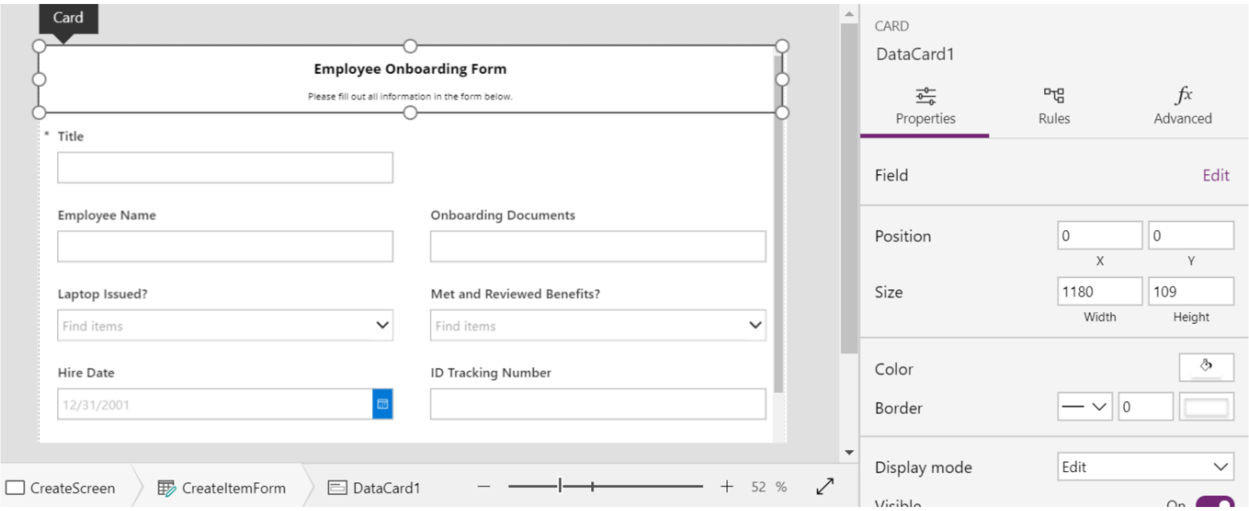
Fields

- ✓ DataCard1
- ☑ Title abc
- ☑ Employee Name abc
- ☑ Onboarding Documents
- ☑ Laptop Issued?

In the form screen you should see your form header at the top of the form.



Click back into the Data menu for the form and change the form Snap Columns from 1 to 2. You will see all columns “snap” to two columns which provides a problem for our header. We have to manually set the width for it. Select your DataCard and in the width type 1180.



Position: 0 (X), 0 (Y)

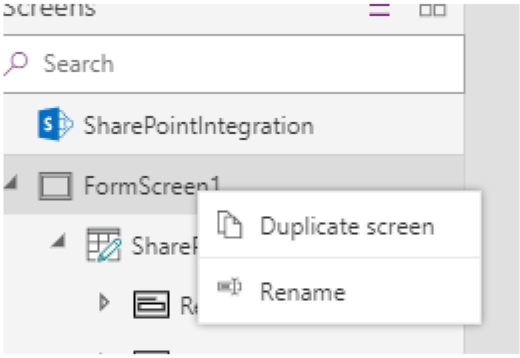
Size: 1180 (Width), 109 (Height)

Border: 0

Display mode: Edit

At this point, we are ready to create custom screens because it is easier to remove fields and customize screens than it is to add them in later.

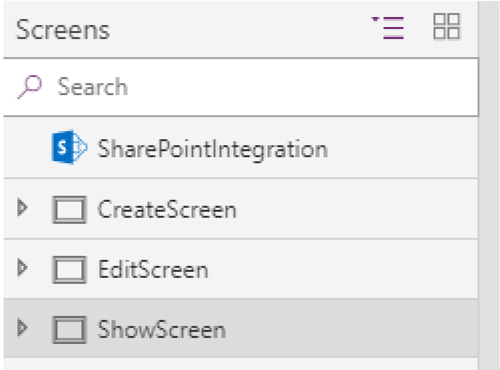
To do this, right click on FormScreen1 and click “Duplicate Screen.”



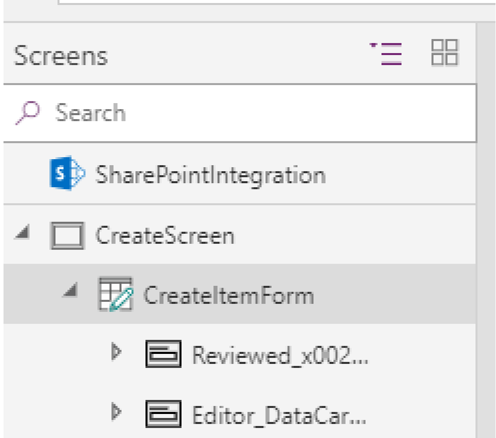
Repeat the process to create a third screen.

Now we want to rename the screens to make customizations easier.

Right click on each screen and rename to the names in the screenshot below.

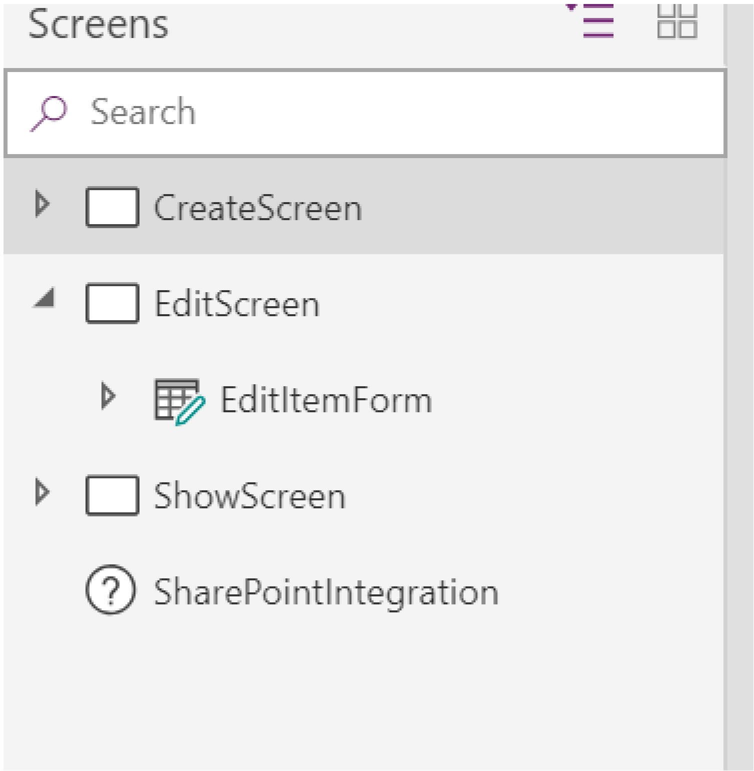


Next expand “CreateScreen.” Rename “SharePointForm1” to “CreateItemForm.”



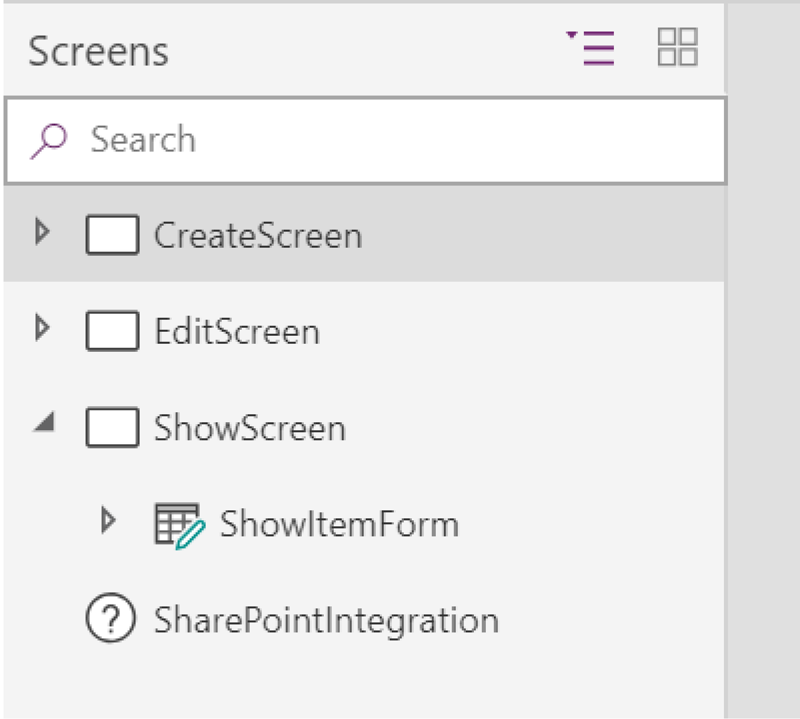
Collapse “CreateScreen” and expand “EditScreen.”

Rename “SharePointForm1_2” to “EditItemForm.”



Collapse “Edit Screen” and expand “Show Screen.”

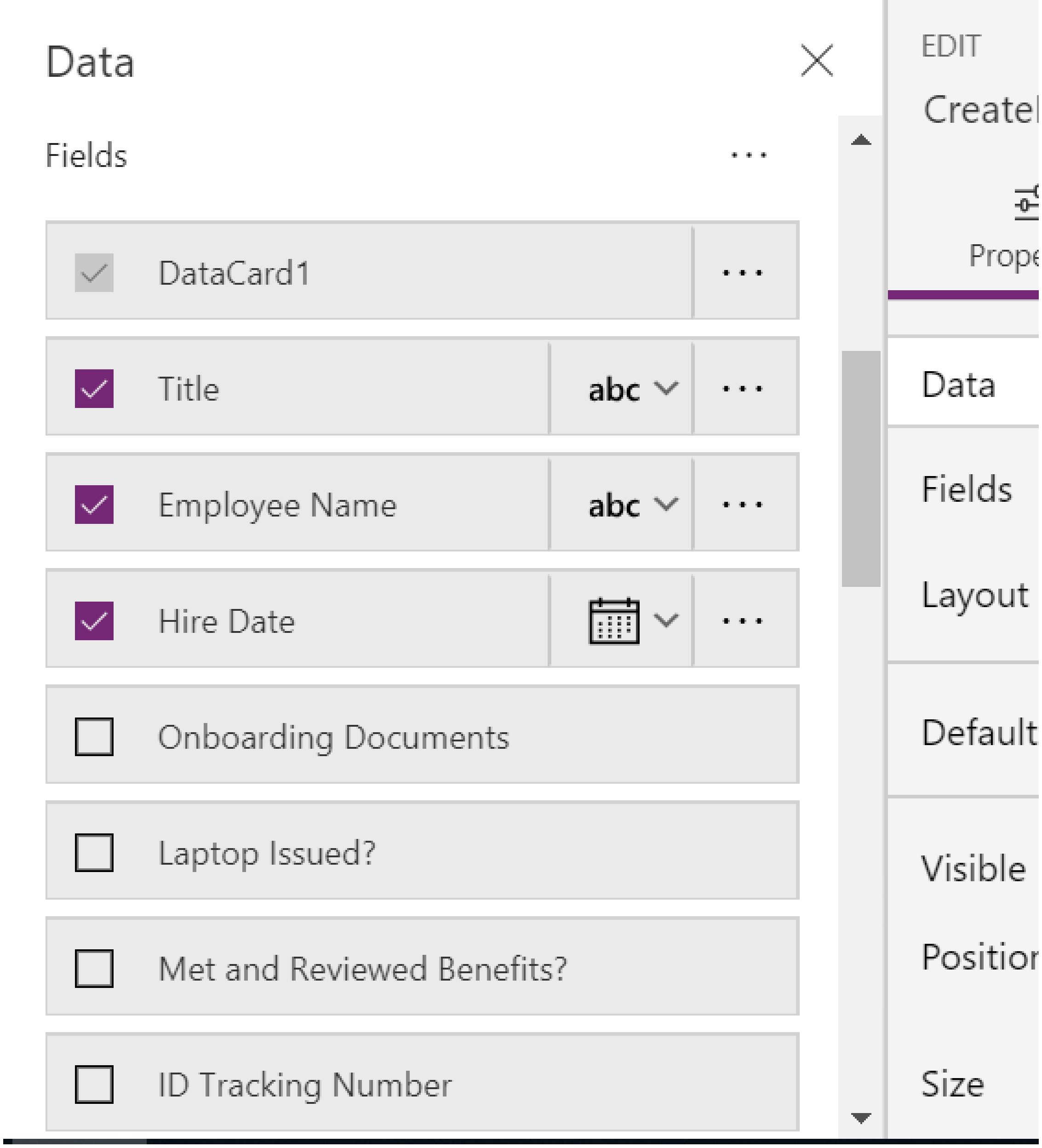
Rename “SharePointItemForm1_3” to “ShowItemForm.”



For our form, we are going to create a unique experience for the CreateScreen because when we create a new user we will only have information for 3 fields: Title, Employee Name, and Hire Date.

Go to CreateItemForm and open the Data menu.

All the visible fields are checked so in order to meet our requirements we will uncheck everything but DataCard1, Title, Employee Name, and Hire Date.



Data

Fields

☑ DataCard1

☑ Title abc

☑ Employee Name abc

☑ Hire Date

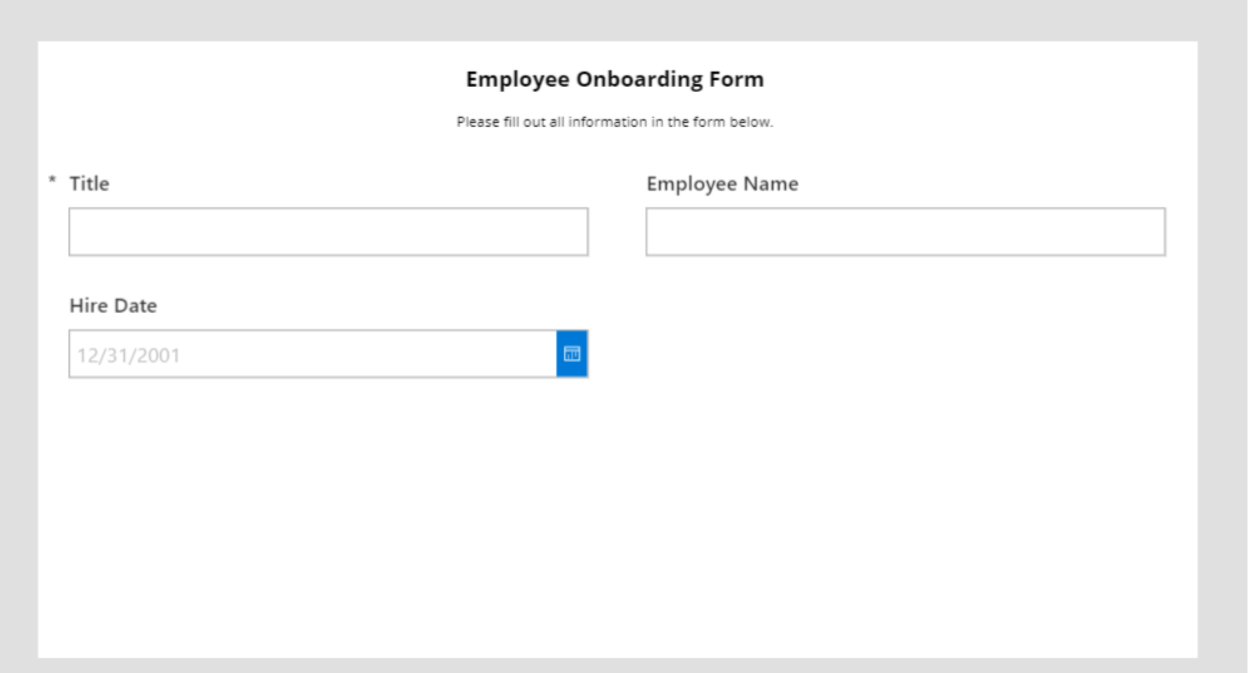
☐ Onboarding Documents

☐ Laptop Issued?

☐ Met and Reviewed Benefits?

☐ ID Tracking Number

Our form will now look like this:

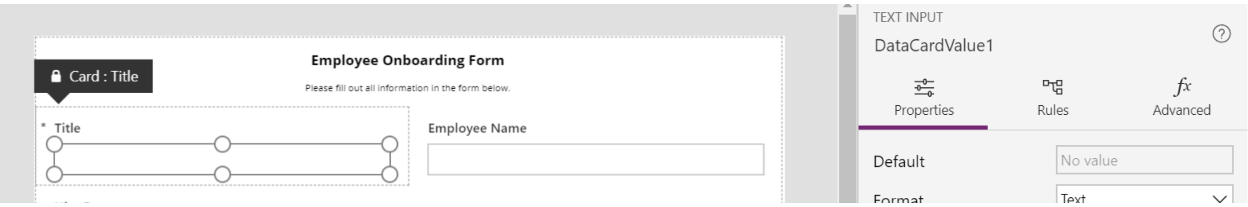


Employee Onboarding Form

Please fill out all information in the form below.

To make this a nicer looking form, we are going to default the Title field and resize it because we will use the Title field later in our Microsoft Flow to create a custom Employee ID.

Select the Title DataCard and select the input box. This will allow us to edit the properties not for the DataCard but the input box itself to set some default properties. To confirm you have the field selected, you should see DataCardValue1 in your right properties menu.



Employee Onboarding Form

Please fill out all information in the form below.

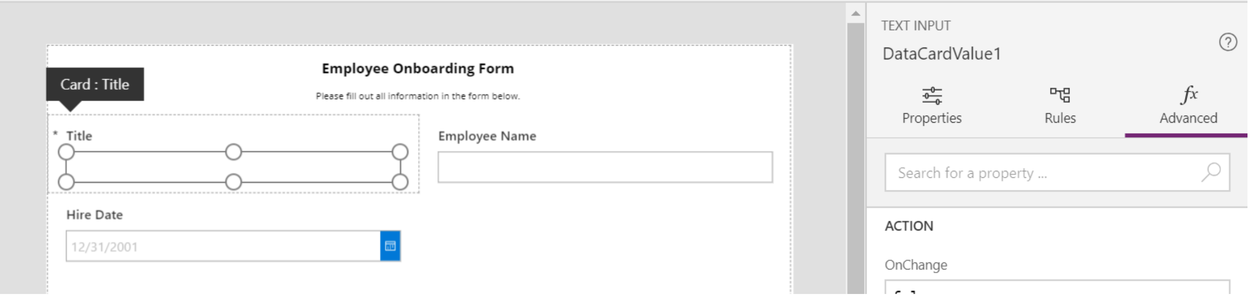
TEXT INPUT

DataCardValue1

Default: No value

Format: Text

Select the Advanced tab and click Unlock to change properties.



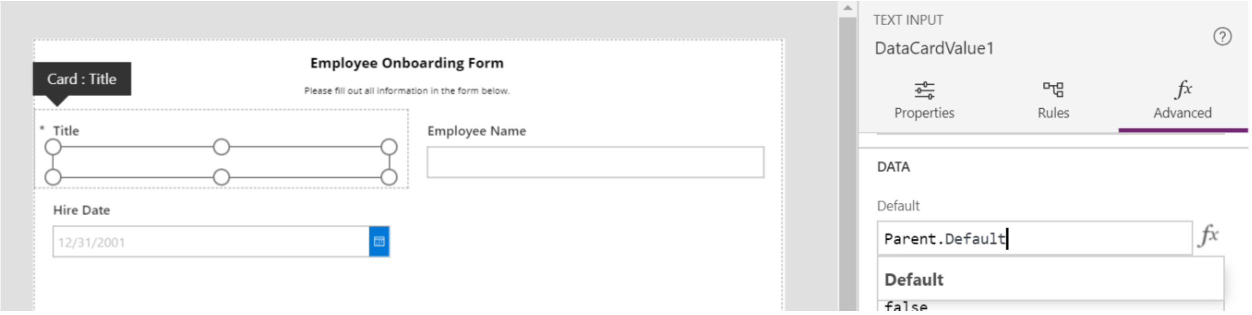
Employee Onboarding Form

Please fill out all information in the form below.

TEXT INPUT

DataCardValue1

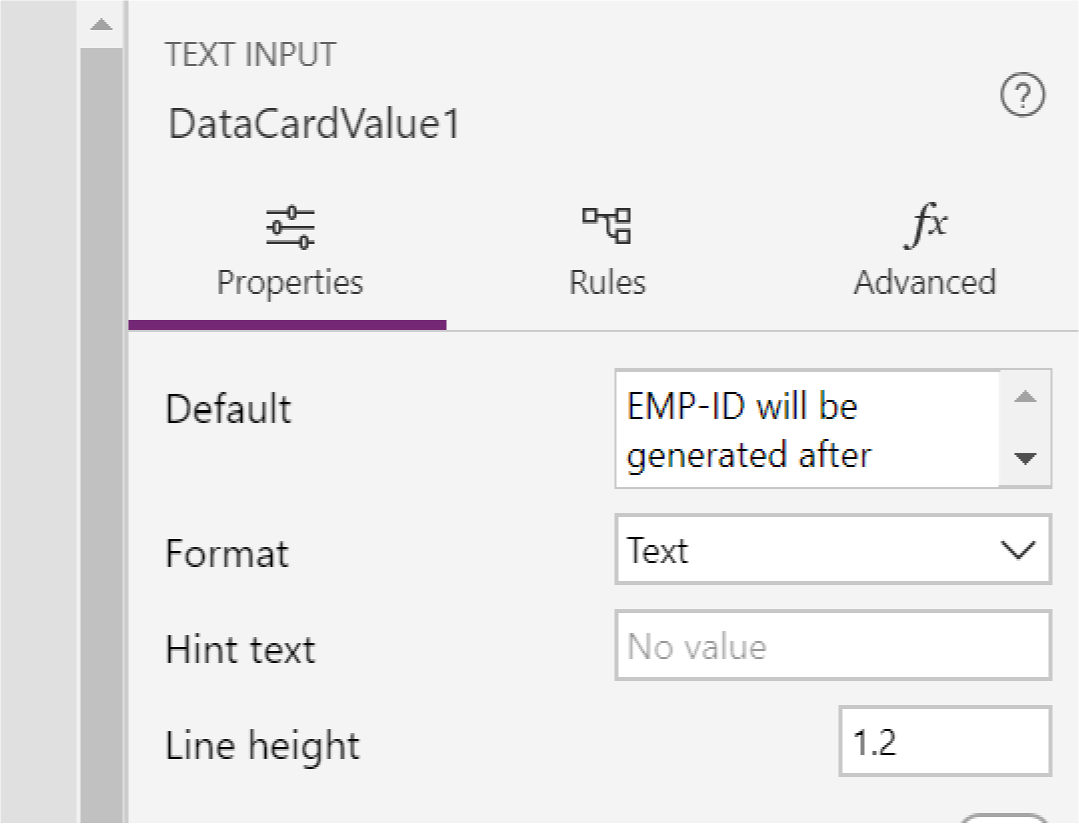
Under the DATA area, click into the Default input and delete Parent.Default.



Default: `Parent.Default`

Default
false

Click back to Properties and in the Default box type: “EMP-ID will be generated after submitting form”.



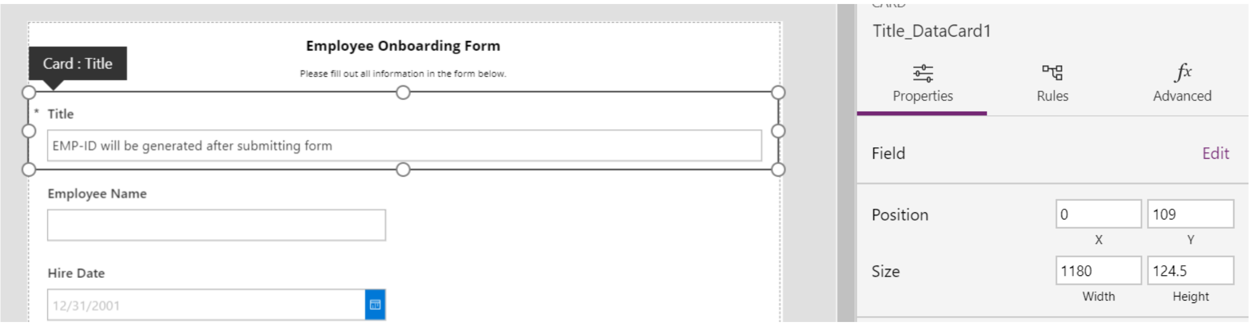
Default: EMP-ID will be generated after

Format: Text

Hint text: No value

Line height: 1.2

Now, select the entire DataCard and change the width to 1180.



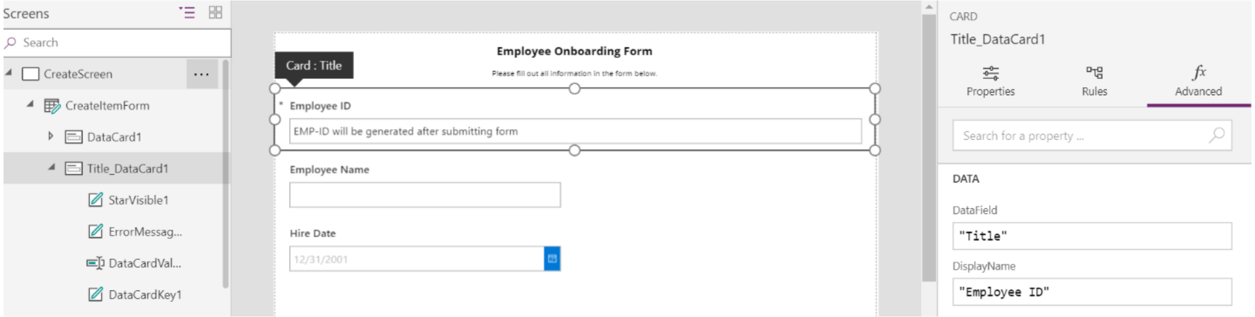
Title: EMP-ID will be generated after submitting form

Field: Edit

Position: 0 (X), 109 (Y)

Size: 1180 (Width), 124.5 (Height)

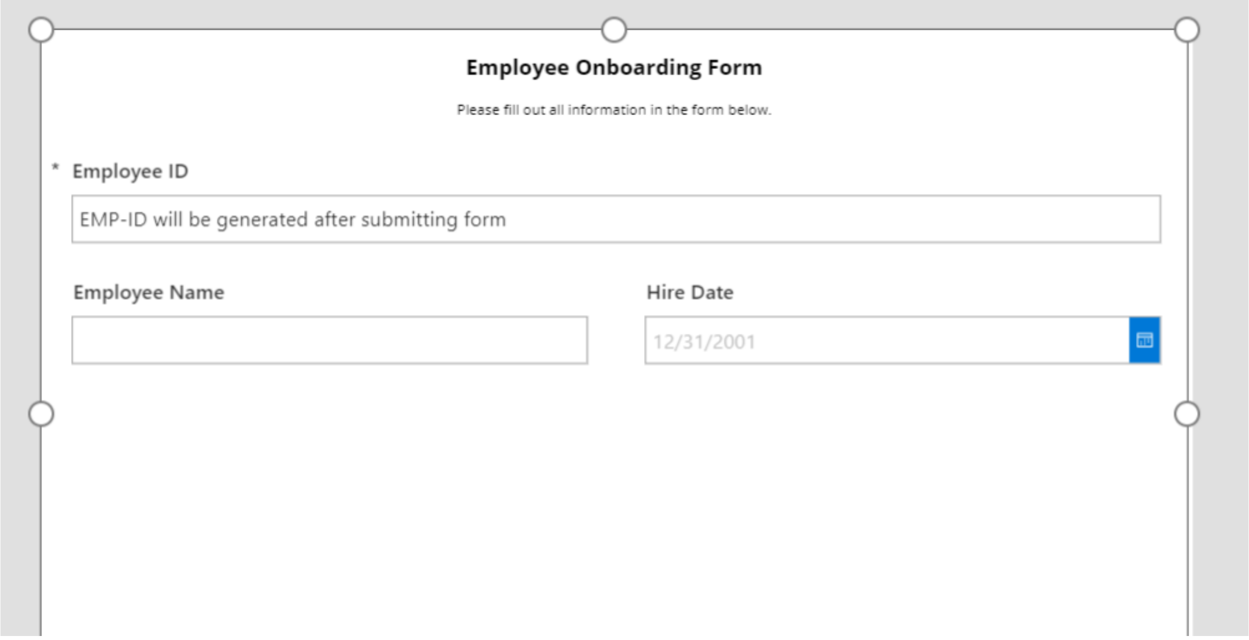
Now click the Advanced tab and in the DisplayName field, change Title to “Employee ID”.



DataField
"Title"

DisplayName
"Employee ID"

Next, we need to put Hire Date on the same line as Employee Name. Select the Hire Date DataCard and put your cursor in the top right of the box. You should see the expand arrow appear diagonally. Drag your cursor up to the right and it should snap beside Employee Name.



Employee Onboarding Form

Please fill out all information in the form below.

* Employee ID

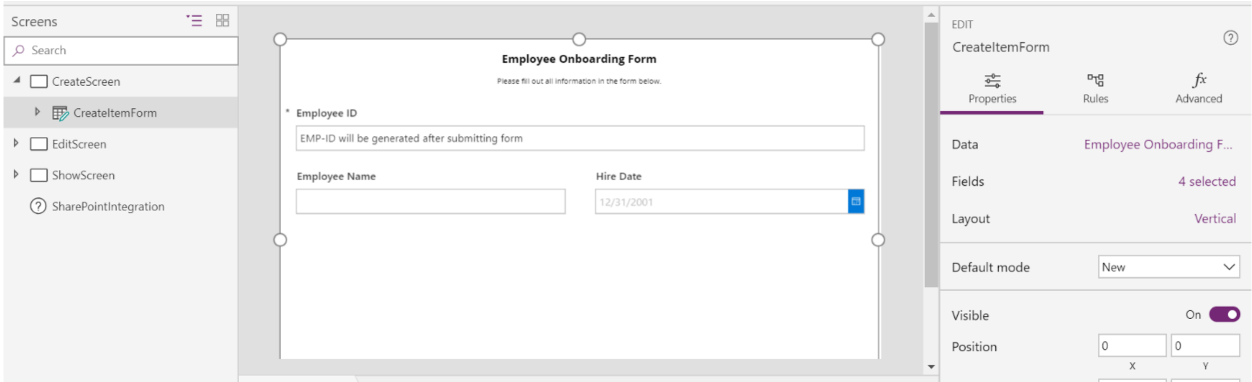
EMP-ID will be generated after submitting form

Employee Name

Hire Date

12/31/2001

The last thing we want to do now is select our entire CreateItemForm and in the properties menu change Default Mode from Edit to New.



Employee Onboarding Form

Please fill out all information in the form below.

* Employee ID

EMP-ID will be generated after submitting form

Employee Name

Hire Date

12/31/2001

EDIT

CreateItemForm

Data: Employee Onboarding F...

Fields: 4 selected

Layout: Vertical

Default mode: New

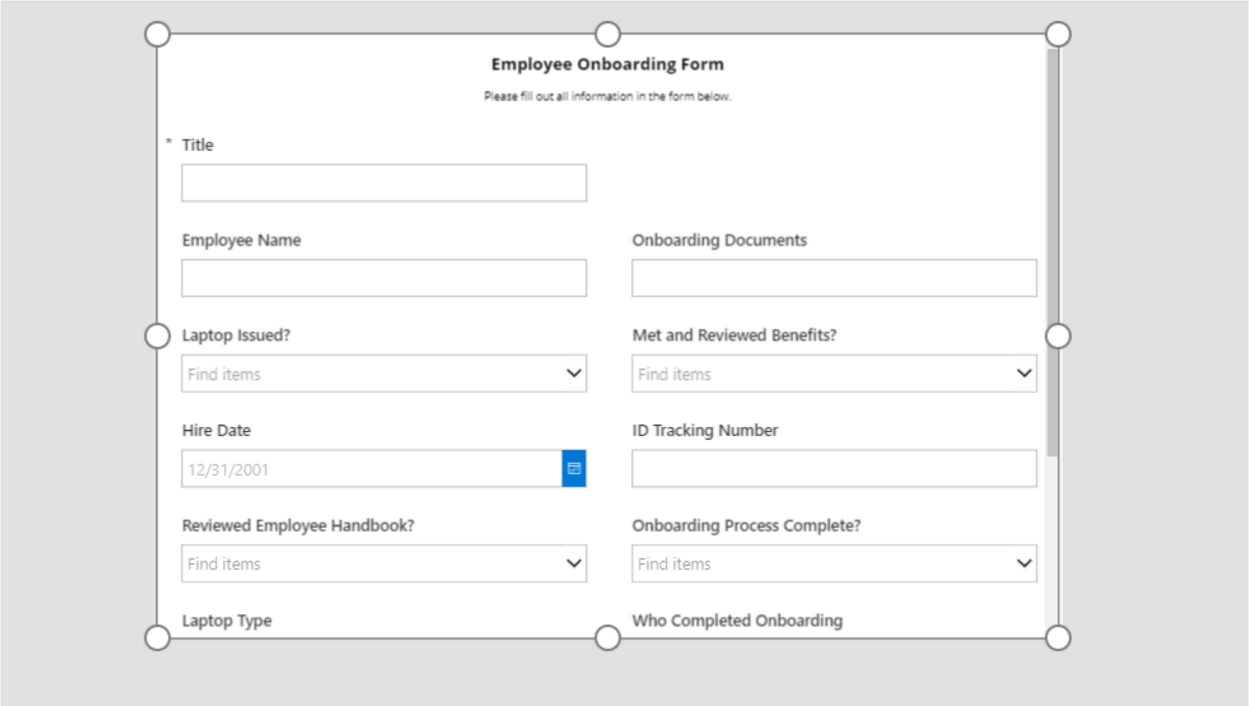
Visible: On

Position: 0 X, 0 Y

We are finished with our CreateScreen and now need to work on our EditScreen.

Collapse CreateScreen and expand EditScreen.

Select EditItemForm to ensure you’re working in the correct screen. It should look like this:



Employee Onboarding Form

Please fill out all information in the form below.

* Title

Employee Name

Onboarding Documents

Laptop Issued?

Find items

Met and Reviewed Benefits?

Find items

Hire Date

12/31/2001

ID Tracking Number

Reviewed Employee Handbook?

Find items

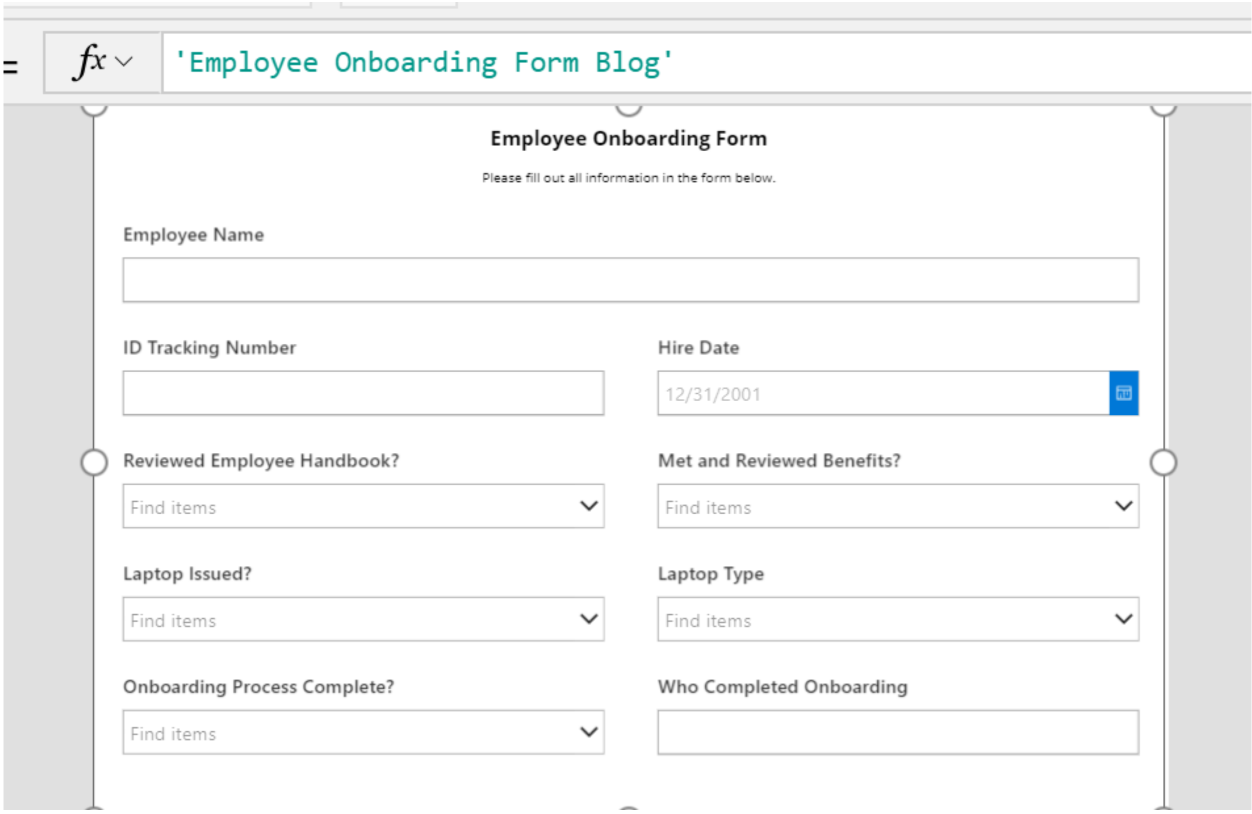
Onboarding Process Complete?

Find items

Laptop Type

Who Completed Onboarding

Open the Data Menu and hide the fields you do not need and rearrange the fields until they look like this:



```
'Employee Onboarding Form Blog'
```

Employee Onboarding Form

Please fill out all information in the form below.

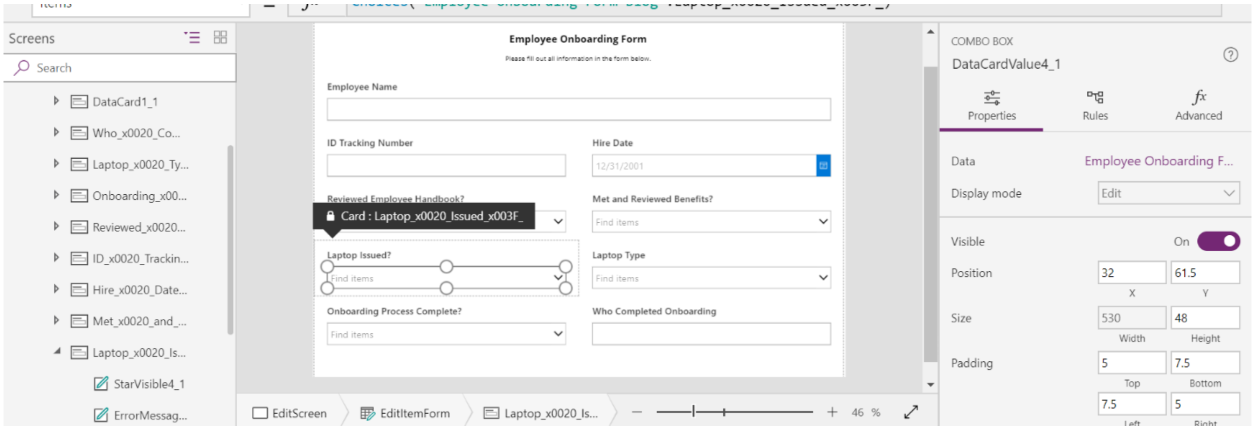
Employee Name	
ID Tracking Number	Hire Date: 12/31/2001
Reviewed Employee Handbook?: Find items	Met and Reviewed Benefits?: Find items
Laptop Issued?: Find items	Laptop Type: Find items
Onboarding Process Complete?: Find items	Who Completed Onboarding

Now, we will create conditional fields to show/hide based on a dropdown's value for Laptop Type and Who Completed Onboarding.

For Laptop Type, we want to only show it if Laptop Issued?'s value is Yes.

For Who Completed Onboarding, we want to show it if Onboarding Process Complete?'s value is also Yes.

To do this we need to select the Laptop Issued DataCard then select the dropdown control itself.



```
Choices('Employee Onboarding Form Blog'.Laptop_x0020_Issued_x003F_)
```

Employee Onboarding Form

Please fill out all information in the form below.

Employee Name	
ID Tracking Number	Hire Date: 12/31/2001
Reviewed Employee Handbook?	Met and Reviewed Benefits?: Find items
Laptop Issued?: Find items	Laptop Type: Find items
Onboarding Process Complete?: Find items	Who Completed Onboarding

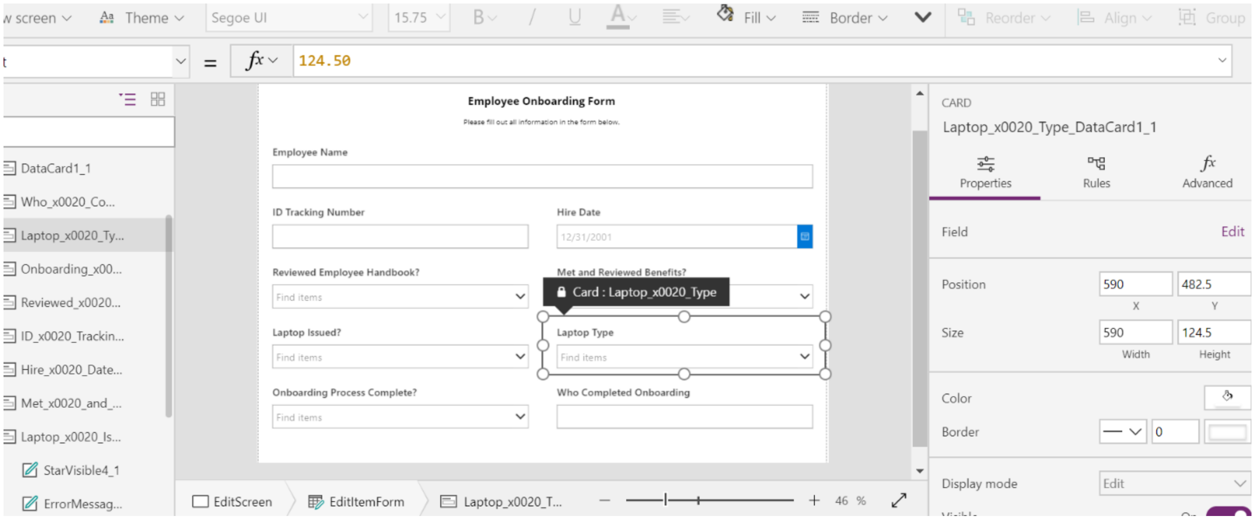
COMBO BOX

DataCardValue4_1

Data	Employee Onboarding F...	
Display mode	Edit	
Visible	On	
Position	32 (X)	61.5 (Y)
Size	530 (Width)	48 (Height)
Padding	5 (Top)	7.5 (Bottom)
	7.5 (Left)	5 (Right)

You should notice that PowerApps refers to this as a COMBO BOX. Copy the name of the COMBO BOX (in this case DataCardValue4_1) and paste into notepad or some other file.

Next select the entire Laptop Type DataCard.



fx 124.50

CARD
Laptop_x0020_Type_DataCard1_1

Field: Edit

Position: 590 (X), 482.5 (Y)

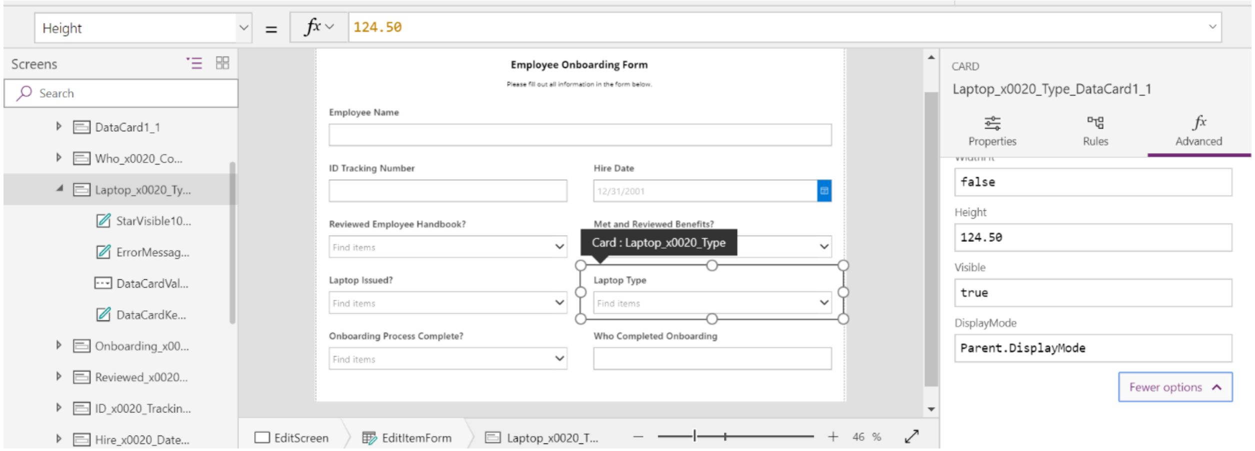
Size: 590 (Width), 124.5 (Height)

Color

Border: 0

Display mode: Edit

From here, we click Advanced and Unlock the ability to edit the Advanced Properties and scroll until we see the Visible Property.



Height = fx 124.50

CARD
Laptop_x0020_Type_DataCard1_1

```
false
```

Height
```
124.50
```

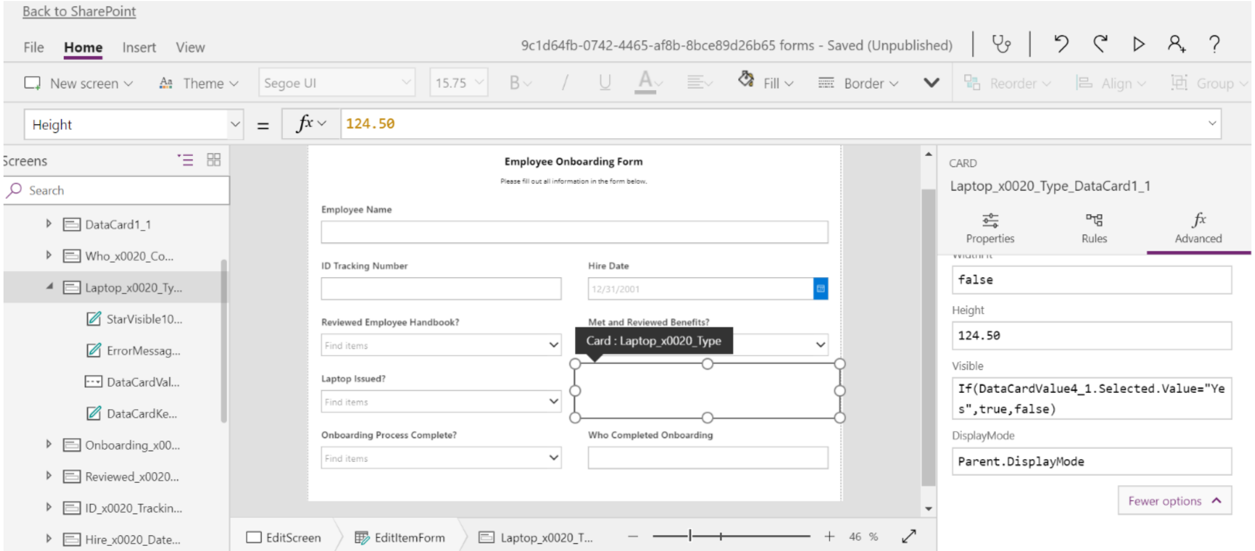
Visible
```
true
```

DisplayMode
```
Parent.DisplayMode
```

Delete “true” and add this formula to the Visible input box:

If(DataCardValue4_1.Selected.Value=”Yes”,true,false)

Once you do this, your Laptop Type DataCard will disappear from the view.



Height = 124.50

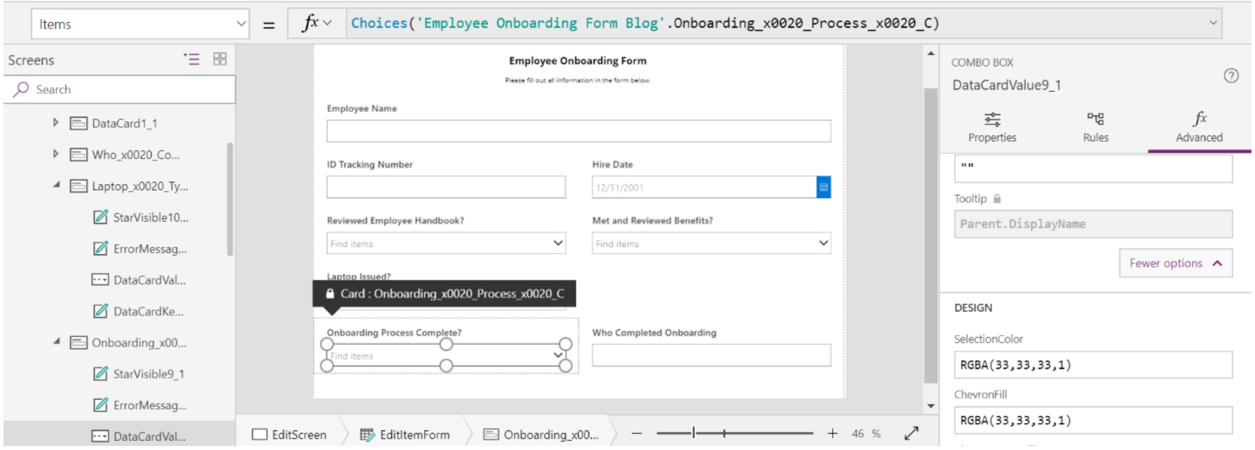
false

Height: 124.50

Visible: If(DataCardValue4_1.Selected.Value="Yes",true,false)

DisplayMode: Parent.DisplayMode

Now, let’s set the same conditions for the Onboarding fields. Select the Onboarding Process Complete? DataCard and then select the dropdown box.



Items = Choices('Employee Onboarding Form Blog'.Onboarding_x0020_Process_x0020_C)

""

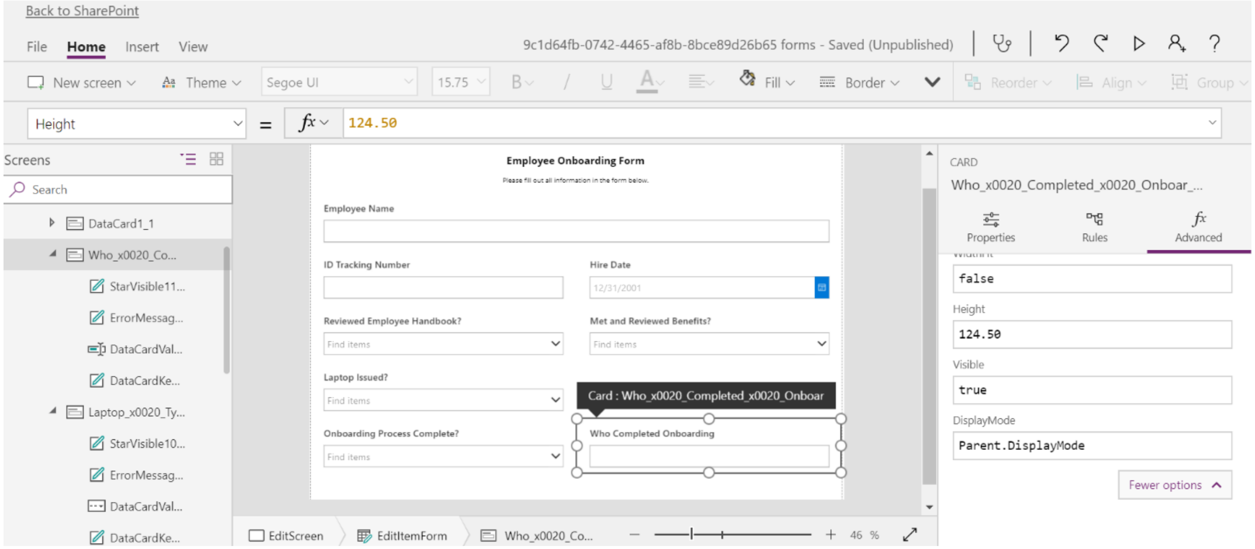
Tooltip: Parent.DisplayName

SelectionColor: RGBA(33,33,33,1)

ChevronFill: RGBA(33,33,33,1)

Copy the DataCardValue9_1 from the COMBO BOX name and select the “Who Completed Onboarding” DataCard.

From here, we click Advanced and Unlock the ability to edit the Advanced Properties and scroll until we see the Visible Property.



Height = 124.50

Width fit: false

Height: 124.50

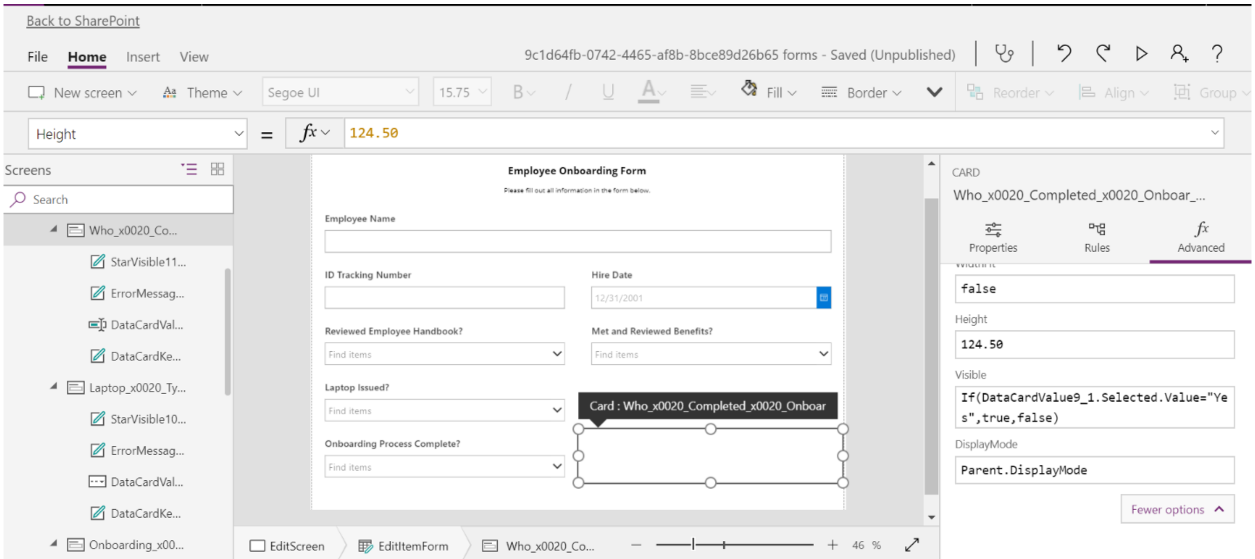
Visible: true

DisplayMode: Parent.DisplayMode

Delete “true” and add this formula to the Visible input box:

If(DataCardValue9_1.Selected.Value=”Yes”,true,false)

Once you do this, your Who Completed Onboarding DataCard will disappear from the view.



Height = 124.50

Width fit: false

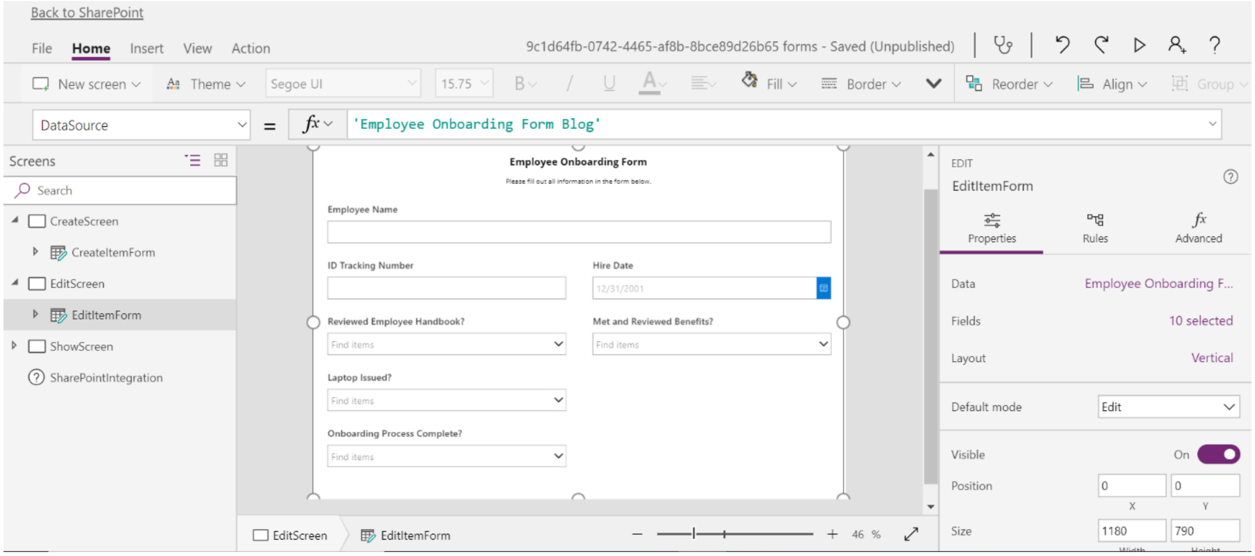
Height: 124.50

Visible:

```
If(DataCardValue9_1.Selected.Value="Yes",true,false)
```

DisplayMode: Parent.DisplayMode

The last thing we want to do on the EditScreen is select EditItemForm again and ensure the Default Mode field is set to Edit.

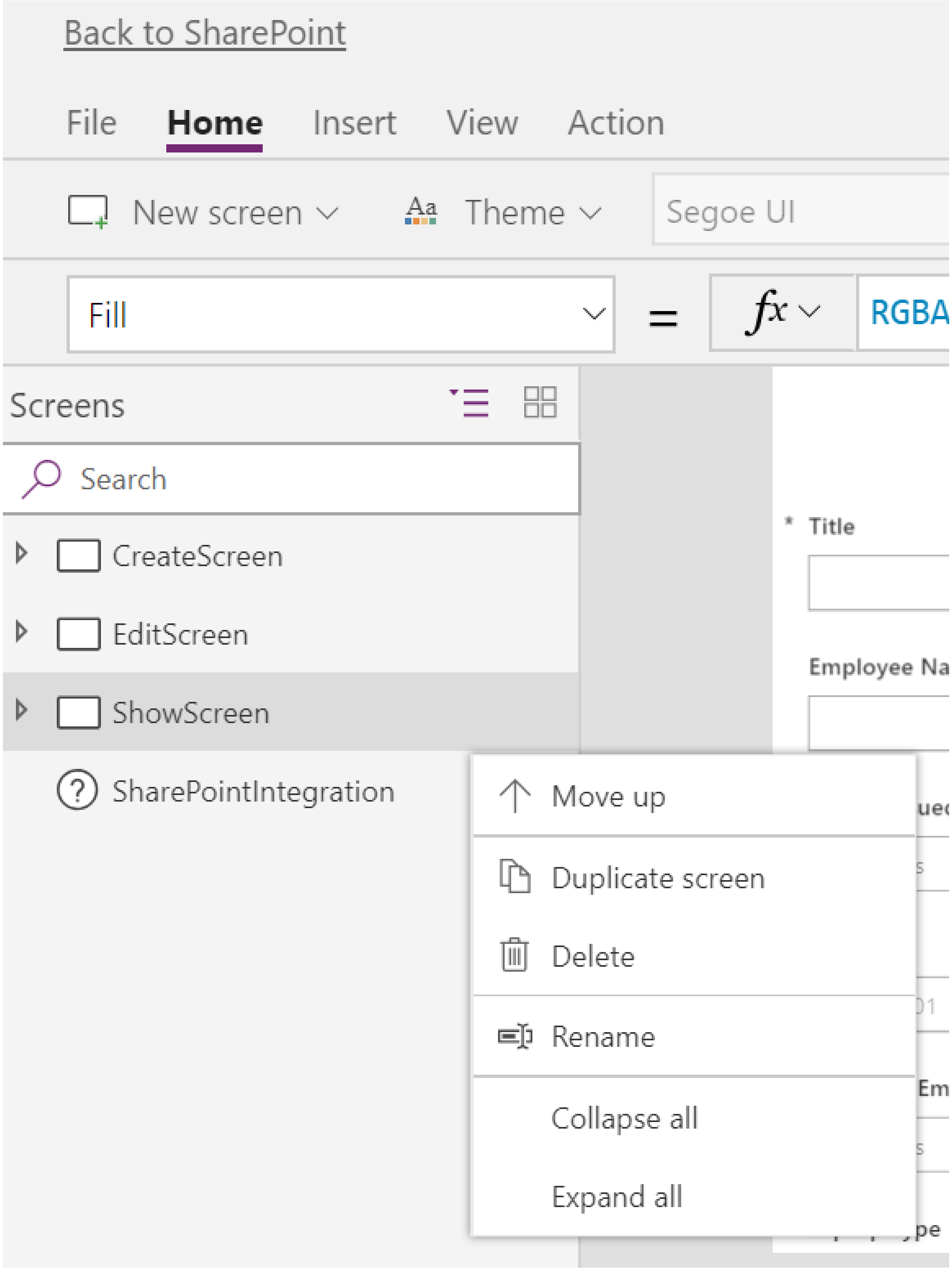


DataSource = 'Employee Onboarding Form Blog'

Data	Employee Onboarding F...
Fields	10 selected
Layout	Vertical
Default mode	Edit
Visible	On
Position	0 X, 0 Y
Size	1180 Width, 790 Height

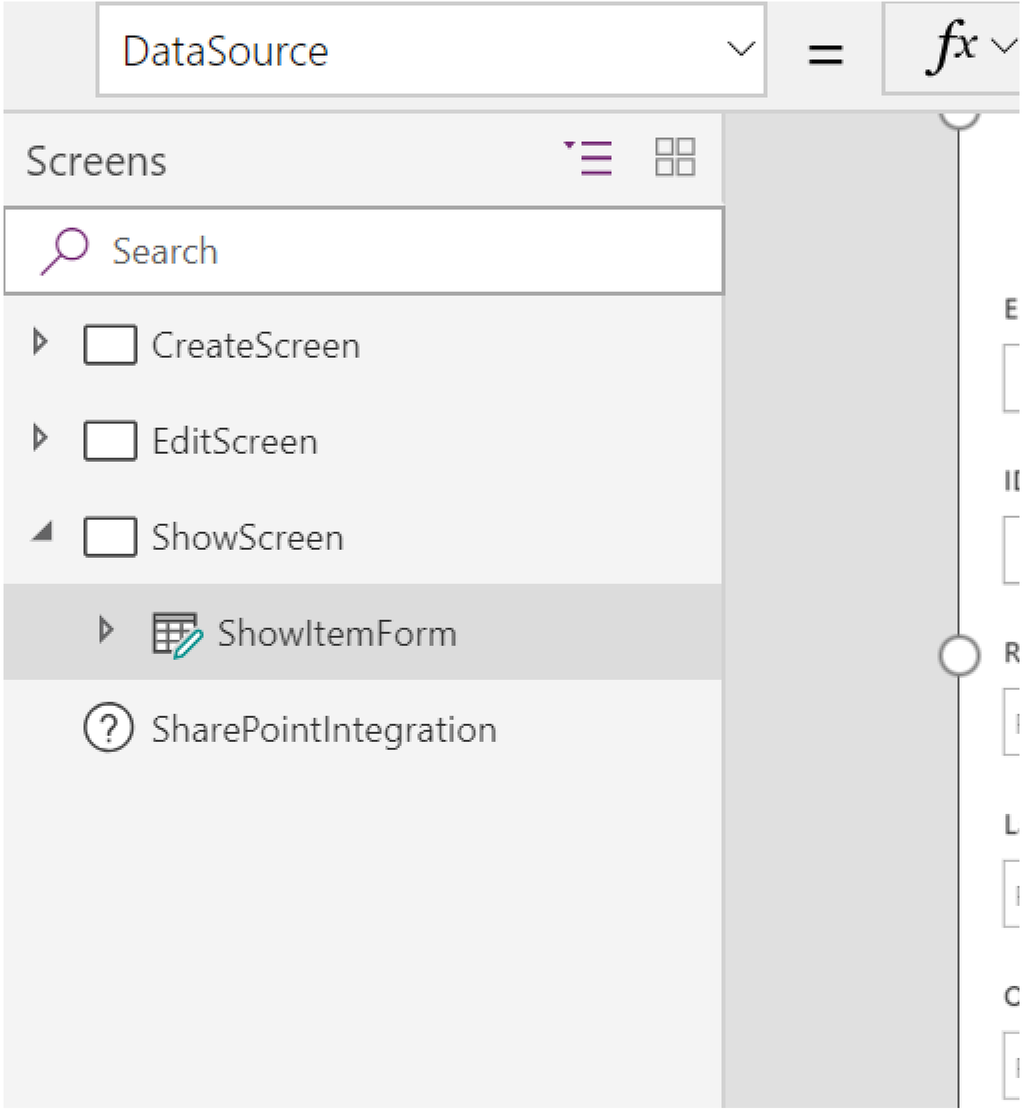
Here we have two options for ShowScreen. In the steps above we created ShowScreen but in this instance we want the exact same look and feel for the ShowScreen as we do on the Edit Screen. So, for us it is easier to deleted ShowScreen, and duplicate EditScreen and rename it appropriately.

Delete ShowScreen.

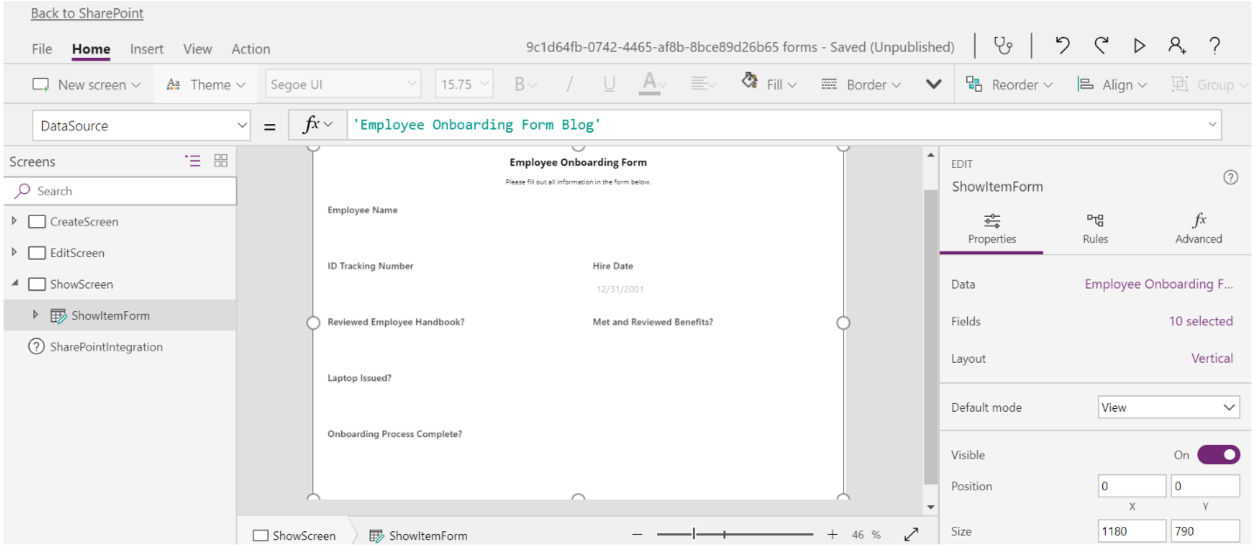


Duplicate EditScreen.

Change the name of the duplication to ShowScreen and ShowItemForm.



Change the Default Mode to View for ShowItemForm and it will now look like this:

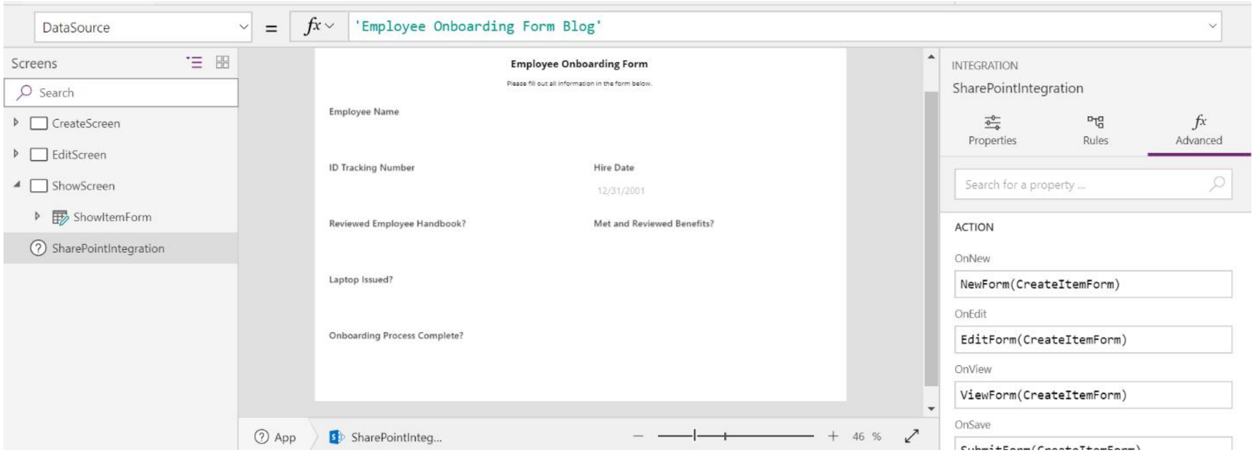


DataSource = 'Employee Onboarding Form Blog'

Data	Employee Onboarding F...
Fields	10 selected
Layout	Vertical
Default mode	View
Visible	On
Position	0 X, 0 Y
Size	1180, 790

Now that we have customized our form screens, we must create formulas to show the different forms depending on whether you are creating, editing, or viewing a list item.

Click “SharePoint Integration” on your left navigation and then click “Advanced” on the right navigation. Be sure to expand More Options.



DataSource = 'Employee Onboarding Form Blog'

OnNew
`NewForm(CreateItemForm)`

OnEdit
`EditForm(CreateItemForm)`

OnView
`ViewForm(CreateItemForm)`

OnSave
`SubmitForm(CreateItemForm)`

NOTE: Thanks to Christopher below for struggling with this. DO NOT directly copy-paste the code for the following SharePoint Integration piece directly in to PowerApps. The font WordPress uses does not play well for code functions in PowerApps. Type this code by hand and it will work fine! Thanks Christopher for identifying an issue and being patient as I struggled to figure out why it would not work.

In the “OnNew” field put the following formula:

Set(SharePointFormMode, “CreateForm”); NewForm(CreateItemForm); Navigate(CreateScreen, ScreenTransition.None)

In the “OnEdit” field put the following formula:

```
Set(SharePointFormMode, “EditForm”); EditForm(EditItemForm); Navigate(EditScreen,
ScreenTransition.None)
```

In the “OnView” field put the following formula:

```
Set(SharePointFormMode, “ShowForm”); ViewForm(ShowItemForm);
Navigate(ShowScreen, ScreenTransition.None)
```

Now, to ensure the submission and cancellation process works correctly we must expand the more options button so that there are OnSave and OnCancel fields.

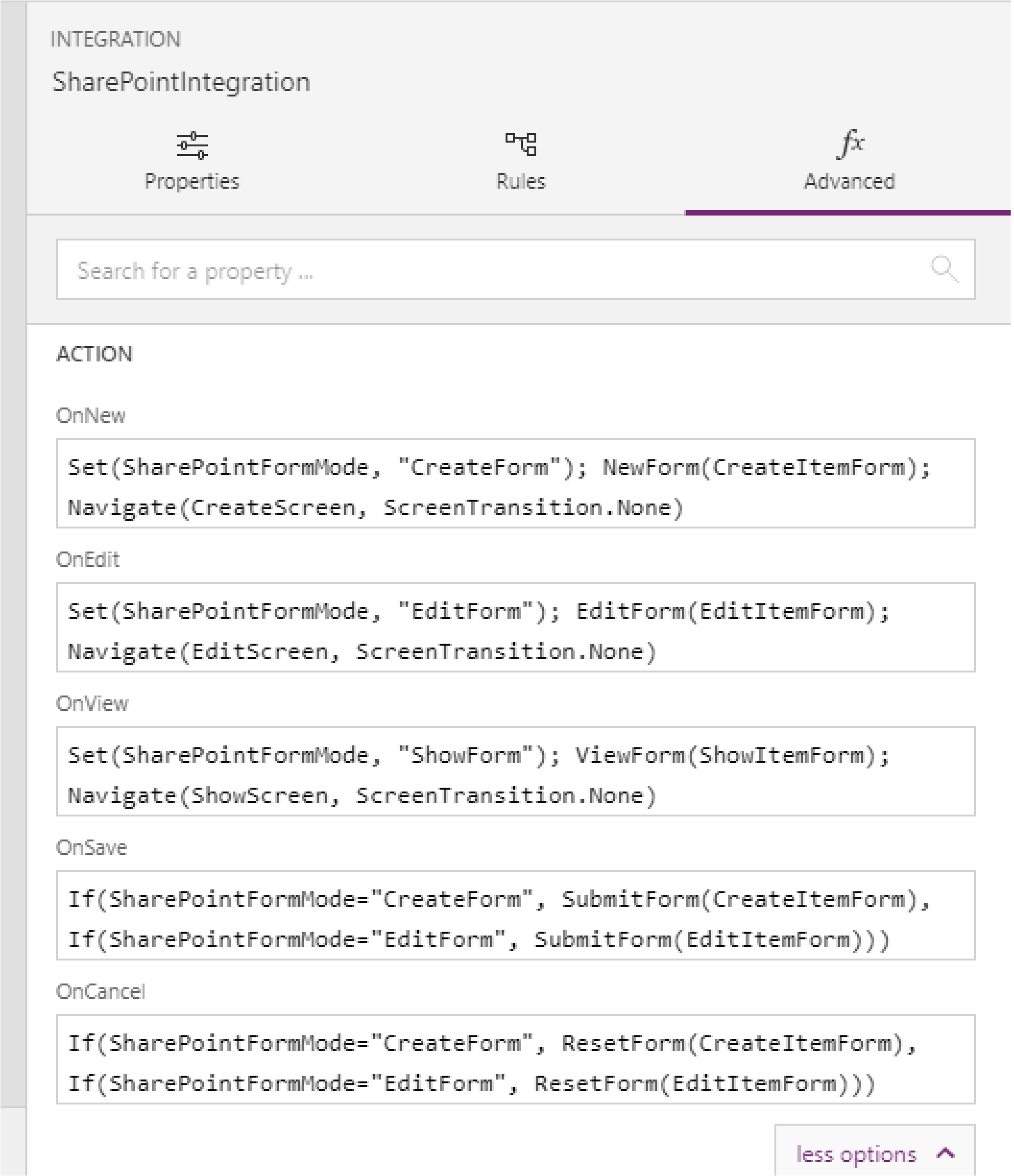
In the “OnSave” field put the following formula:

```
If(SharePointFormMode=”CreateForm”, SubmitForm(CreateItemForm),
If(SharePointFormMode=”EditForm”, SubmitForm(EditItemForm)))
```

In the “OnCancel” field put the following formula:

```
If(SharePointFormMode=”CreateForm”, ResetForm(CreateItemForm),
If(SharePointFormMode=”EditForm”, ResetForm(EditItemForm)))
```

The fields will look like this when filled out:



ACTION

OnNew

```
Set(SharePointFormMode, "CreateForm"); NewForm(CreateItemForm);
Navigate(CreateScreen, ScreenTransition.None)
```

OnEdit

```
Set(SharePointFormMode, "EditForm"); EditForm(EditItemForm);
Navigate(EditScreen, ScreenTransition.None)
```

OnView

```
Set(SharePointFormMode, "ShowForm"); ViewForm(ShowItemForm);
Navigate(ShowScreen, ScreenTransition.None)
```

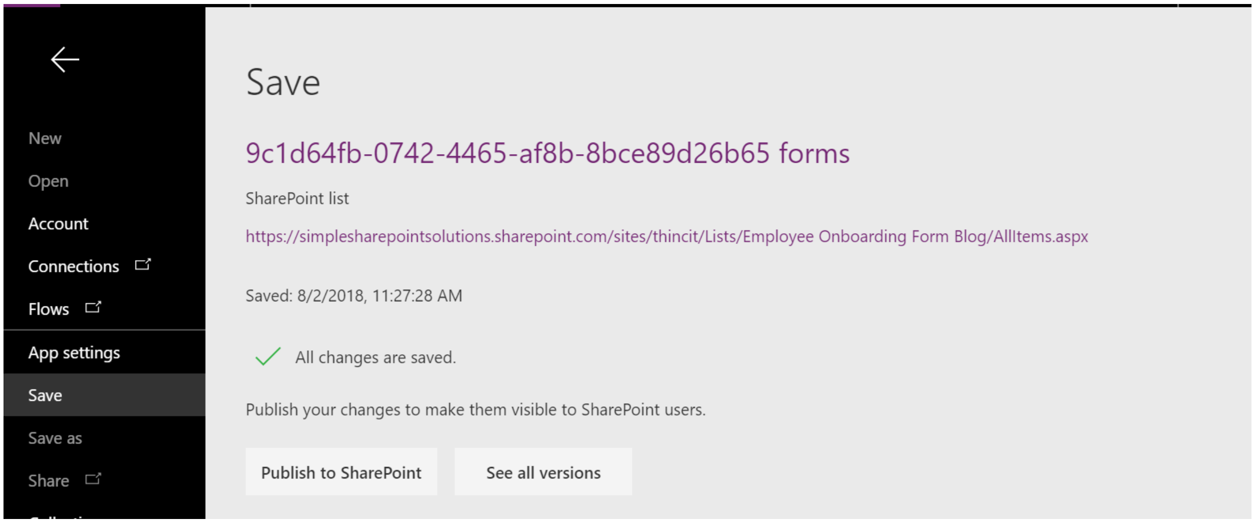
OnSave

```
If(SharePointFormMode="CreateForm", SubmitForm(CreateItemForm),
If(SharePointFormMode="EditForm", SubmitForm(EditItemForm)))
```

OnCancel

```
If(SharePointFormMode="CreateForm", ResetForm(CreateItemForm),
If(SharePointFormMode="EditForm", ResetForm(EditItemForm)))
```

Now we can save and publish our form by clicking the “File” tab, saving, and publishing.



Save

9c1d64fb-0742-4465-af8b-8bce89d26b65 forms

SharePoint list

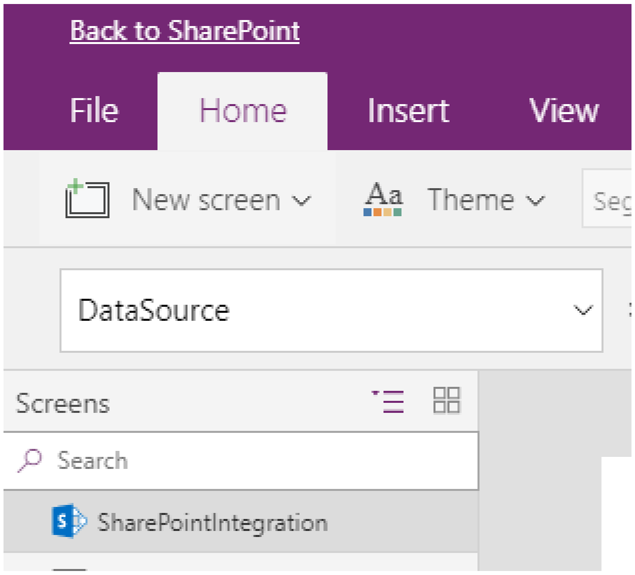
https://simplesharepointsolutions.sharepoint.com/sites/thincit/Lists/Employee Onboarding Form Blog/AllItems.aspx

Saved: 8/2/2018, 11:27:28 AM

✓ All changes are saved.

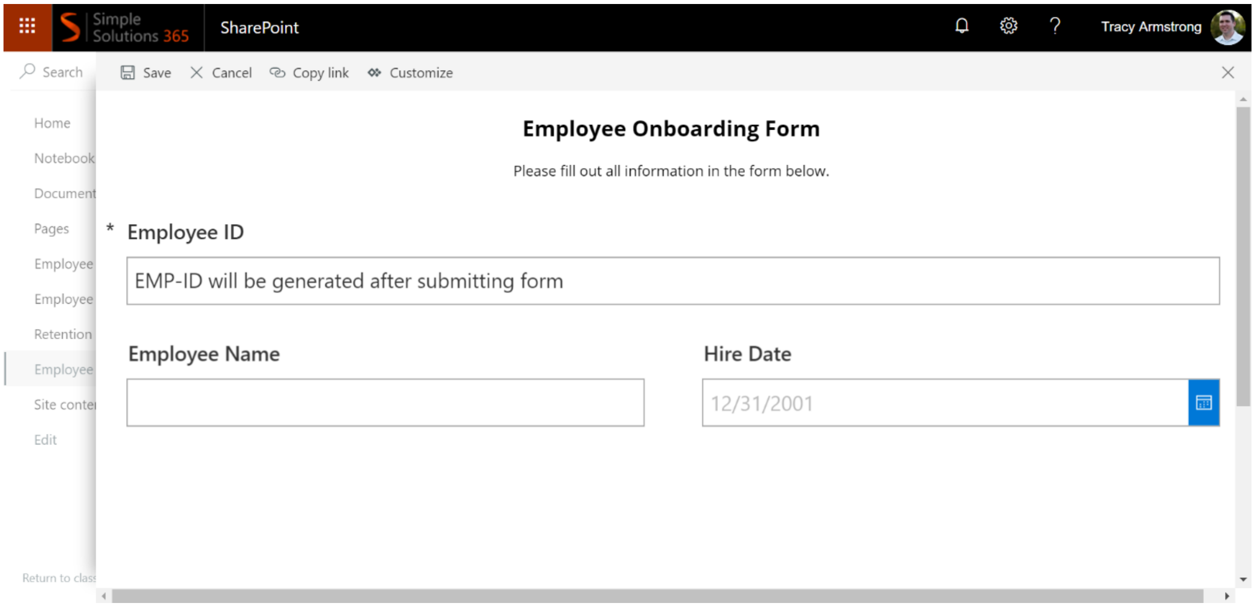
Publish your changes to make them visible to SharePoint users.

Click “Back to SharePoint.”



Now we can test our solution.

Click “New” to create a new list item. If we are successful, our form should show up like this:



Employee Onboarding Form

Please fill out all information in the form below.

* Employee ID

EMP-ID will be generated after submitting form

Employee Name

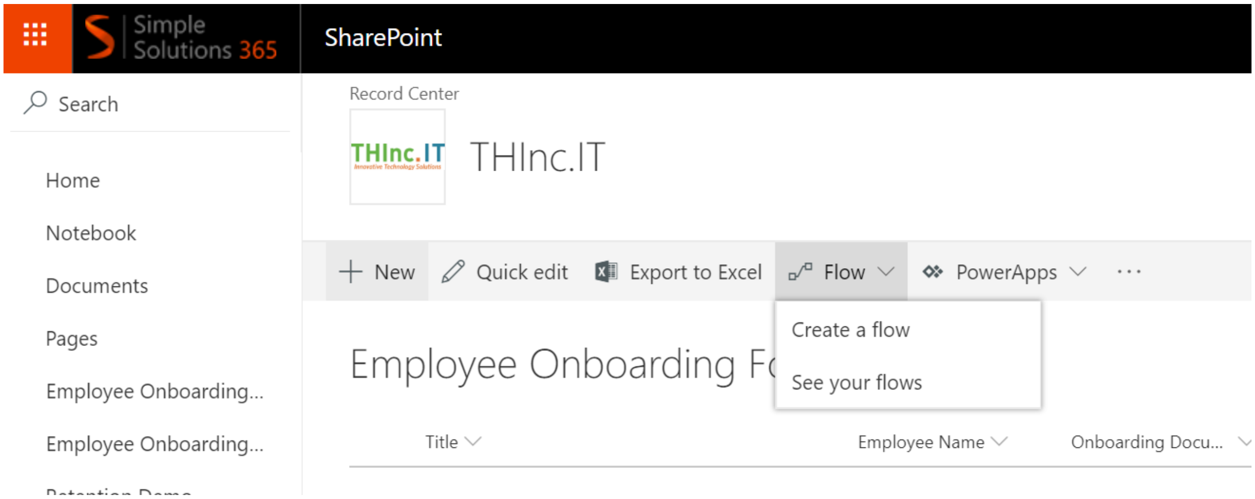
Hire Date

12/31/2001

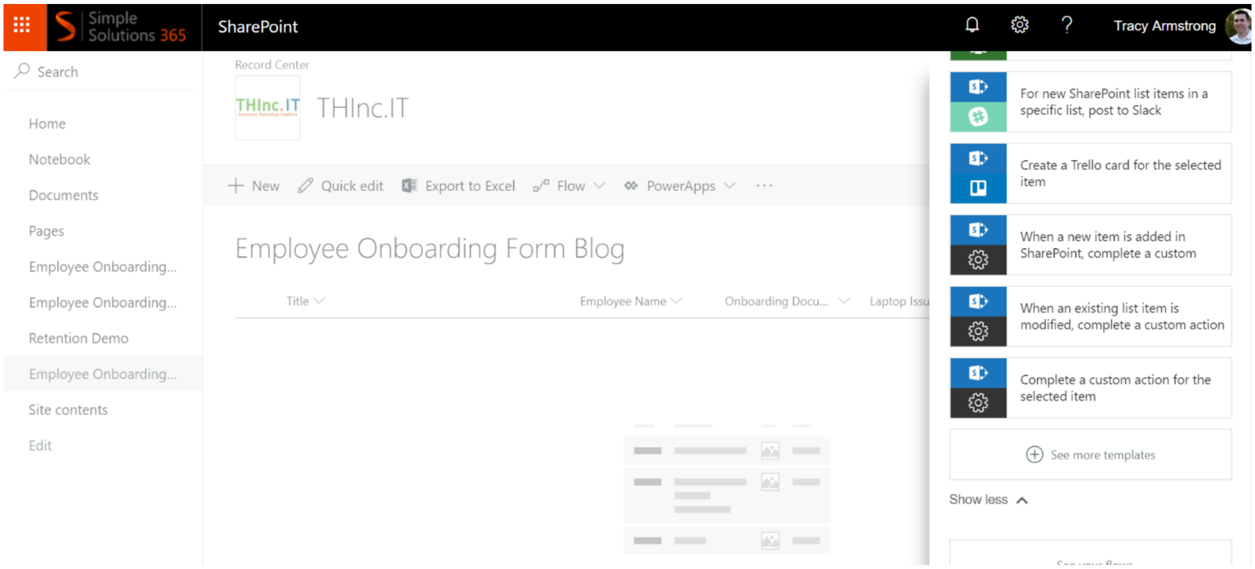
This confirms that our custom form was created properly.

STEP 4: Create Microsoft Flow for Automation

On the main list screen, click “Flow” to “Create a flow.”



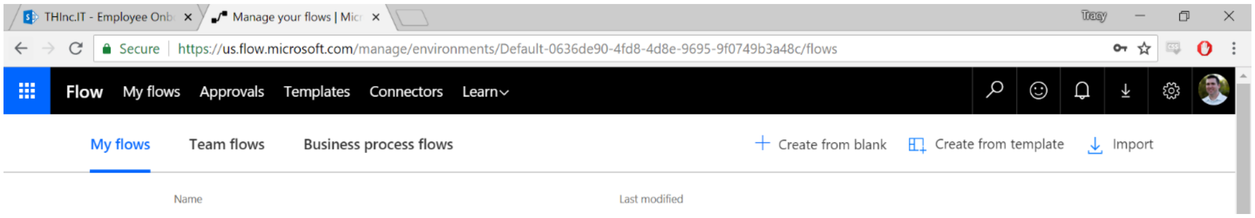
On the pop-out menu to the right we want to expand the option to see more templates and choose “more templates”.



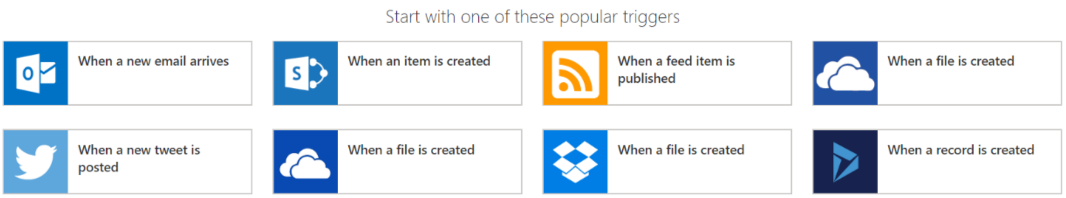
Flow will open a new tab or window.

Hit the “My Flows” link in the top nav bar.

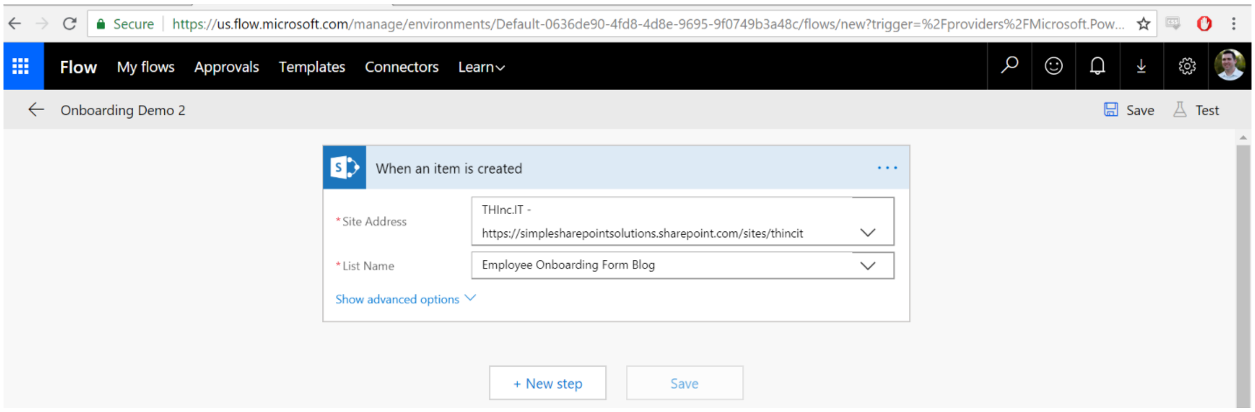
On the next screen click “Create from blank”.



Start with one of the popular triggers for SharePoint “When an item is created”.



On the Flow screen, change the Flow name to something you can easily remember and click “Edit” under the “When a new item is created” box to verify it is pulling from the correct list.

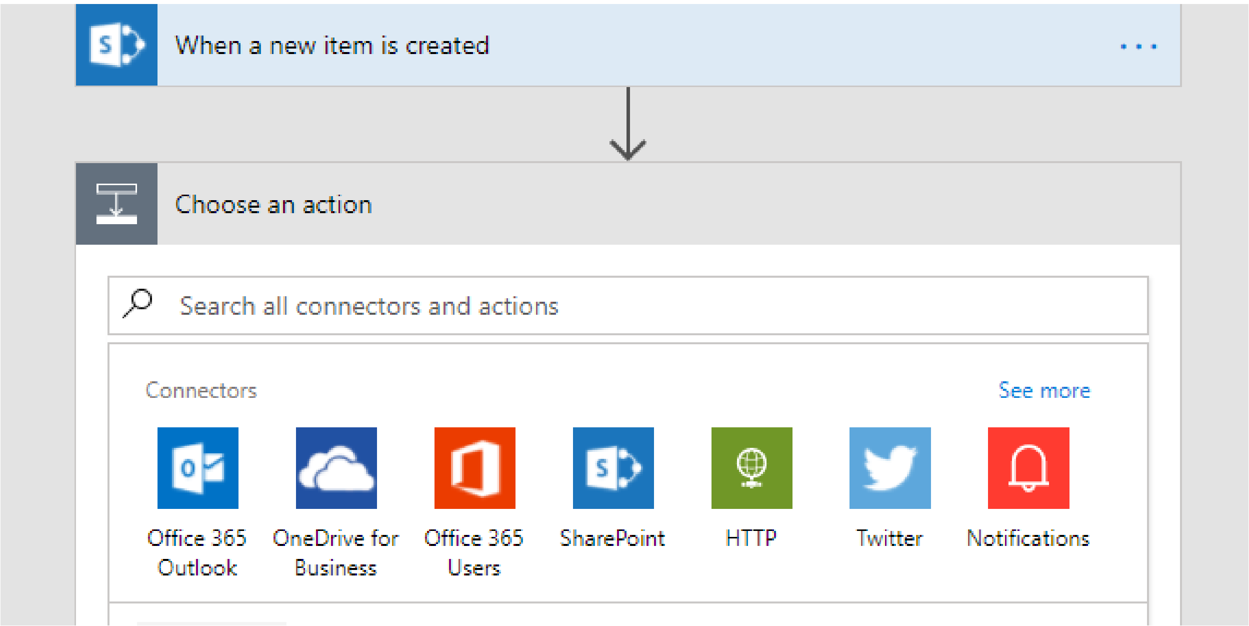

Secure
https://us.flow.microsoft.com/manage/environments/Default-0636de90-4fd8-4d8e-9695-9f0749b3a48c/flows/new?trigger=%2Fproviders%2FMicrosoft.Pow...
Flow
My flows
Approvals
Templates
Connectors
Learn
Onboarding Demo 2
Save
Test
Show advanced options
+ New step
Save

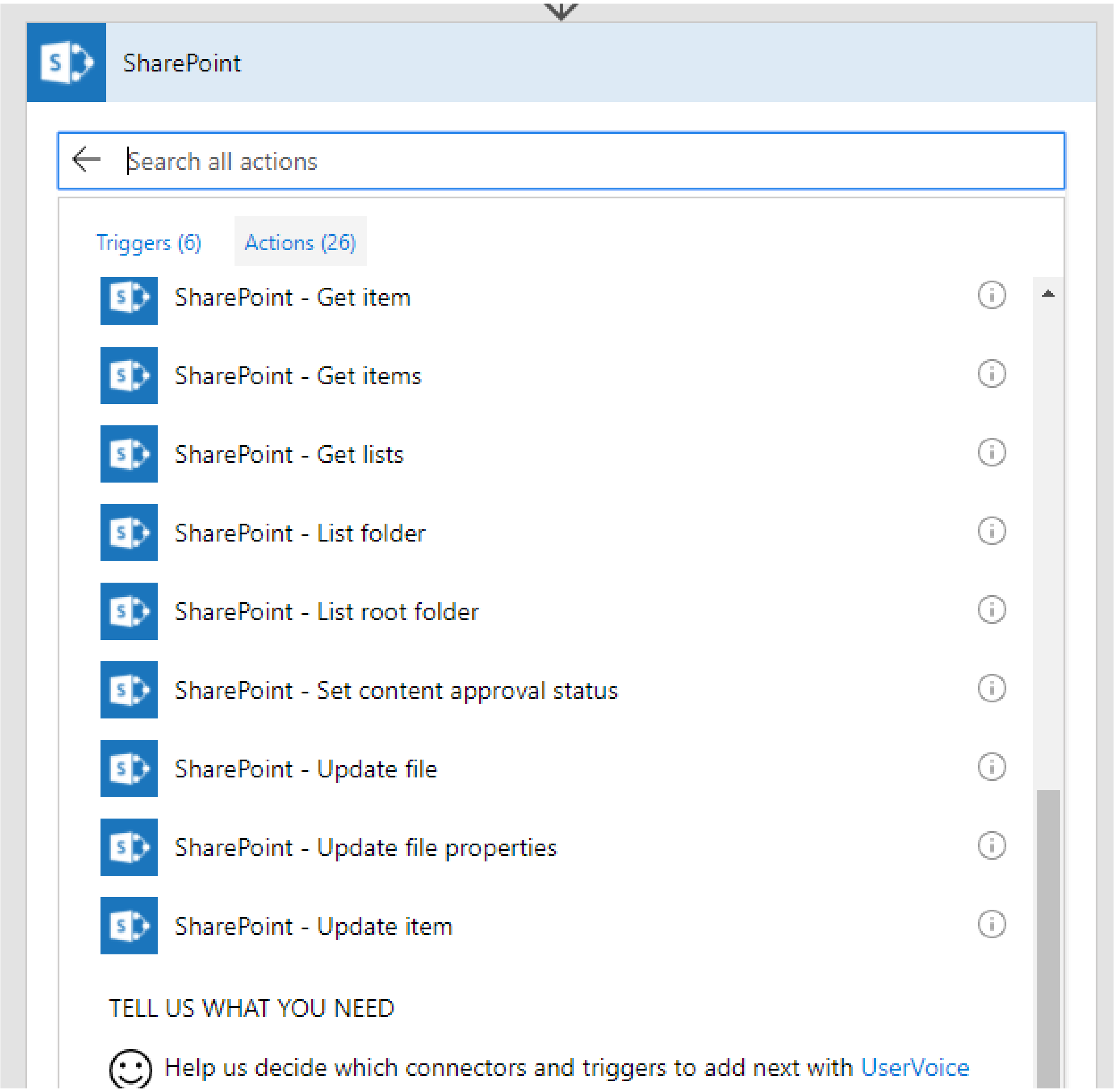
When an item is created

*Site Address: THInc.IT - https://simplesharepointsolutions.sharepoint.com/sites/thincit

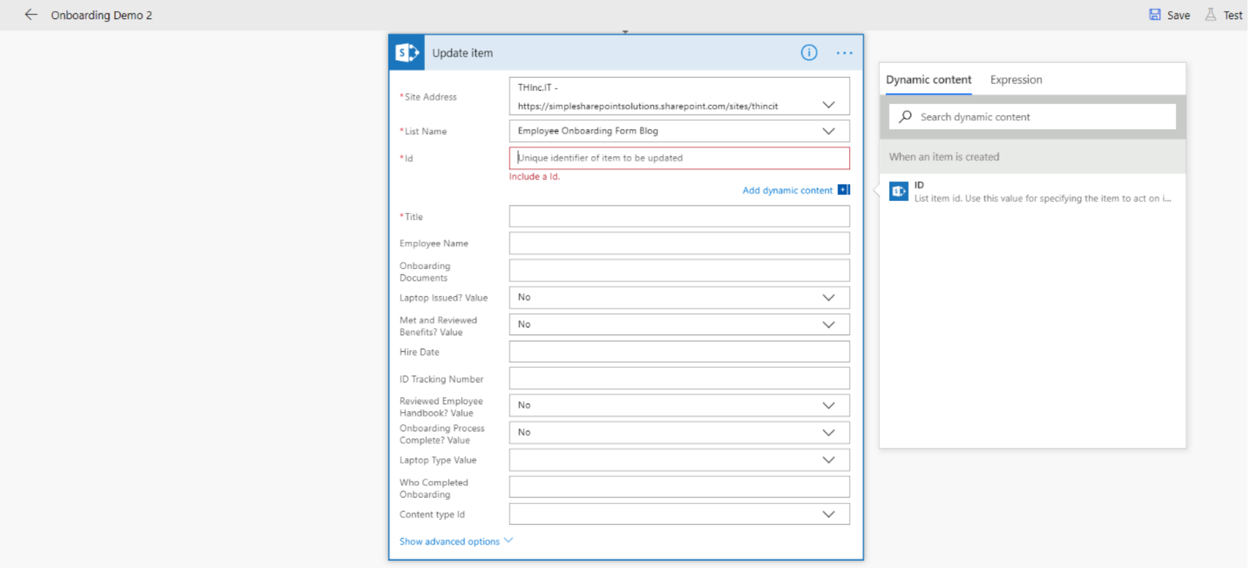
*List Name: Employee Onboarding Form Blog

Once we verify the list click “New step”, then “Add an action”, then “SharePoint” under Connectors, and select “Update Item.”





From here we will select the SharePoint site where our list resides, select the current list, and at that point a menu will drop down with all of our available fields.



*Site Address	THInc.IT - https://simplesharepointsolutions.sharepoint.com/sites/thincit
*List Name	Employee Onboarding Form Blog
*Id	Unique identifier of item to be updated Include a Id.
*Title	
Employee Name	
Onboarding Documents	
Laptop Issued? Value	No
Met and Reviewed Benefits? Value	No
Hire Date	
ID Tracking Number	
Reviewed Employee Handbook? Value	No
Onboarding Process Complete? Value	No
Laptop Type Value	
Who Completed Onboarding	
Content type Id	

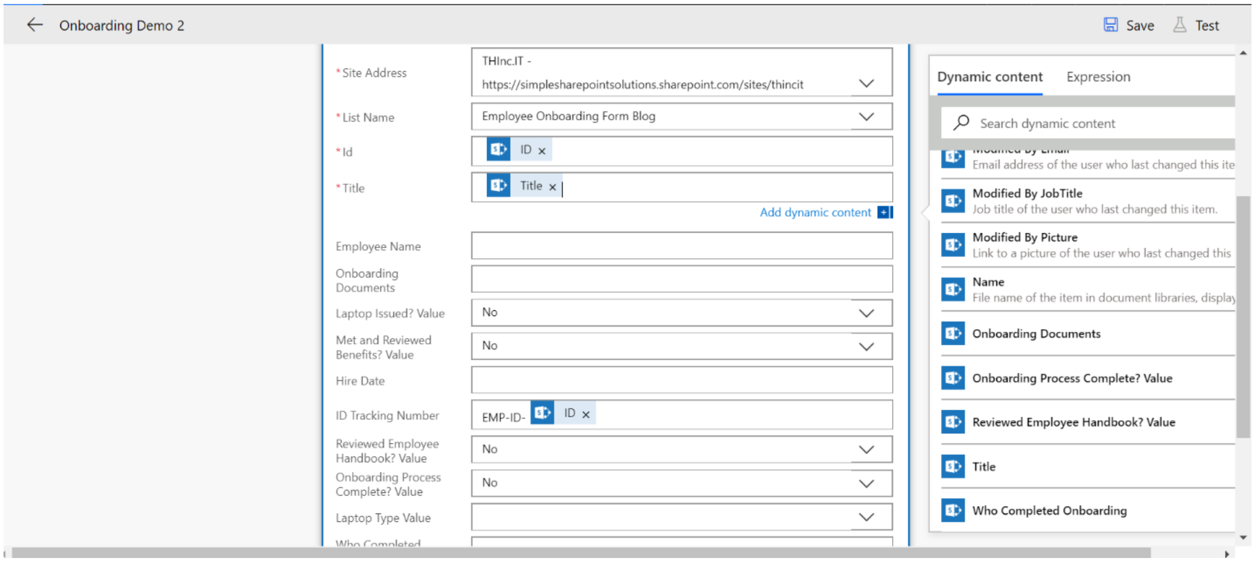
When an item is created

ID
List item id. Use this value for specifying the item to act on i...

Click into the ID field and select the ID option in the right pop-out menu under “When a new item is created.” This specifies we are pulling the ID from the item we are working with from the above step.

Click into the “ID Tracking Number” field. Type in your custom prefix then click “ID” from the right pop-out menu again. This will attach the SharePoint List Item ID to your Prefix when you create the item and update the field.

Your Update item box should like this:



*Site Address	THInc.IT - https://simplesharepointsolutions.sharepoint.com/sites/thincit
*List Name	Employee Onboarding Form Blog
*Id	ID ×
*Title	Title ×
Employee Name	
Onboarding Documents	
Laptop Issued? Value	No
Met and Reviewed Benefits? Value	No
Hire Date	
ID Tracking Number	EMP-ID- ID ×
Reviewed Employee Handbook? Value	No
Onboarding Process Complete? Value	No
Laptop Type Value	
Who Completed	

Modified by Email
Email address of the user who last changed this ite

Modified By JobTitle
Job title of the user who last changed this item.

Modified By Picture
Link to a picture of the user who last changed this

Name
File name of the item in document libraries, display

Onboarding Documents

Onboarding Process Complete? Value

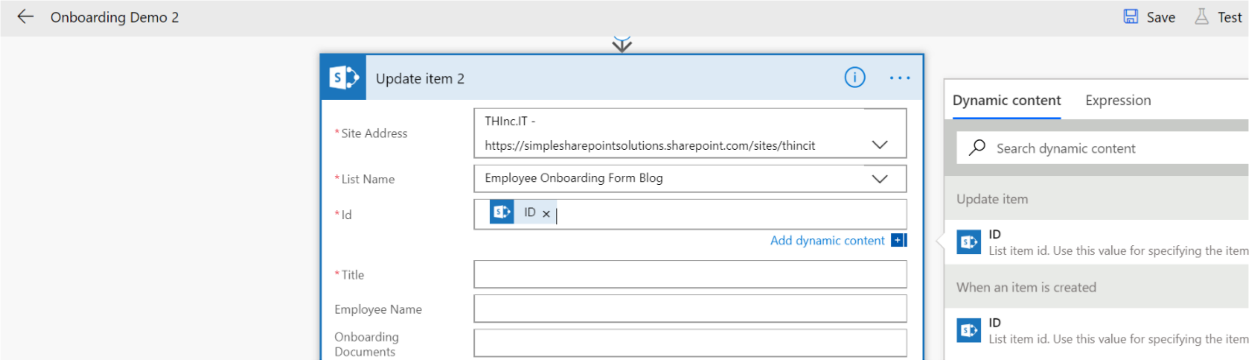
Reviewed Employee Handbook? Value

Title

Who Completed Onboarding

Below our Update item box, click “New Step”, then “Add an action”, “SharePoint”, and then “Update item.”

Repeat the process from above of selecting our site, current list, but now in the Id field choose the ID under "Update item" in the right pop-out menu. It should look like this:



Update item 2

*Site Address: THInc.IT - https://simplesharepointsolutions.sharepoint.com/sites/thincit

*List Name: Employee Onboarding Form Blog

*Id: ID ×

*Title:

Employee Name:

Onboarding Documents:

Update item

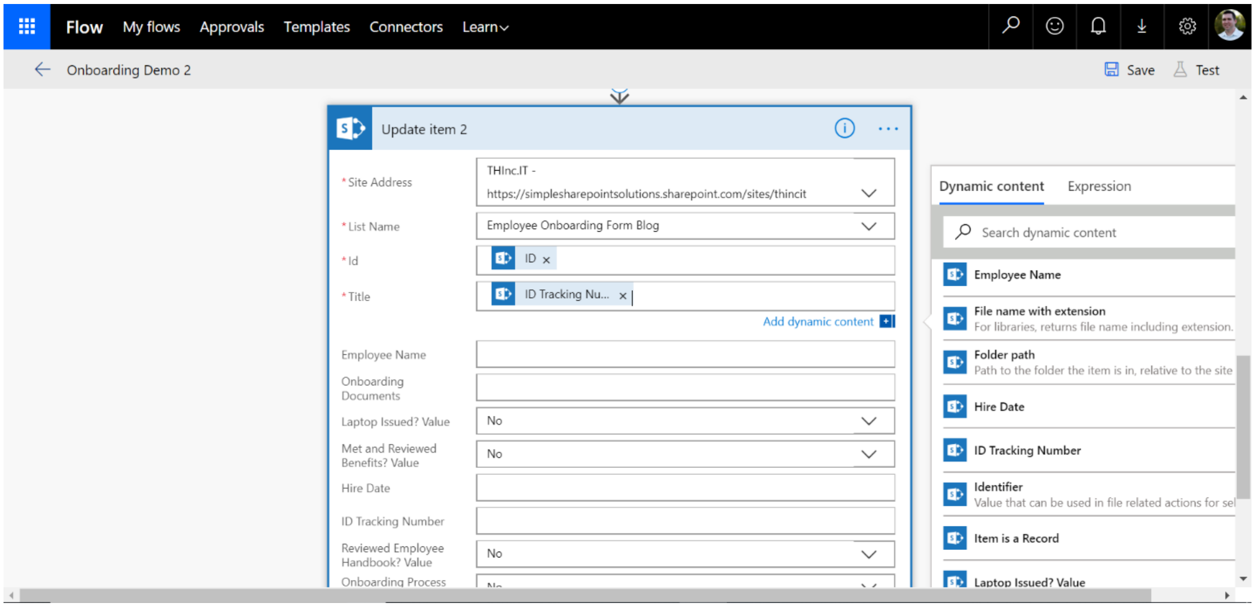
ID
List item id. Use this value for specifying the item

When an item is created

ID
List item id. Use this value for specifying the item

Now click into the title field and delete the "New Onboarding Item" value.

In the right pop-out menu select "ID Tracking Number – Demo" under "Update item" and ensure it clicks into the Title field as show below:



Update item 2

*Site Address: THInc.IT - https://simplesharepointsolutions.sharepoint.com/sites/thincit

*List Name: Employee Onboarding Form Blog

*Id: ID ×

*Title: ID Tracking Nu... ×

Employee Name:

Onboarding Documents:

Laptop Issued? Value: No

Met and Reviewed Benefits? Value: No

Hire Date:

ID Tracking Number:

Reviewed Employee Handbook? Value: No

Onboarding Process

Employee Name

File name with extension
For libraries, returns file name including extension.

Folder path
Path to the folder the item is in, relative to the site

Hire Date

ID Tracking Number

Identifier
Value that can be used in file related actions for sel

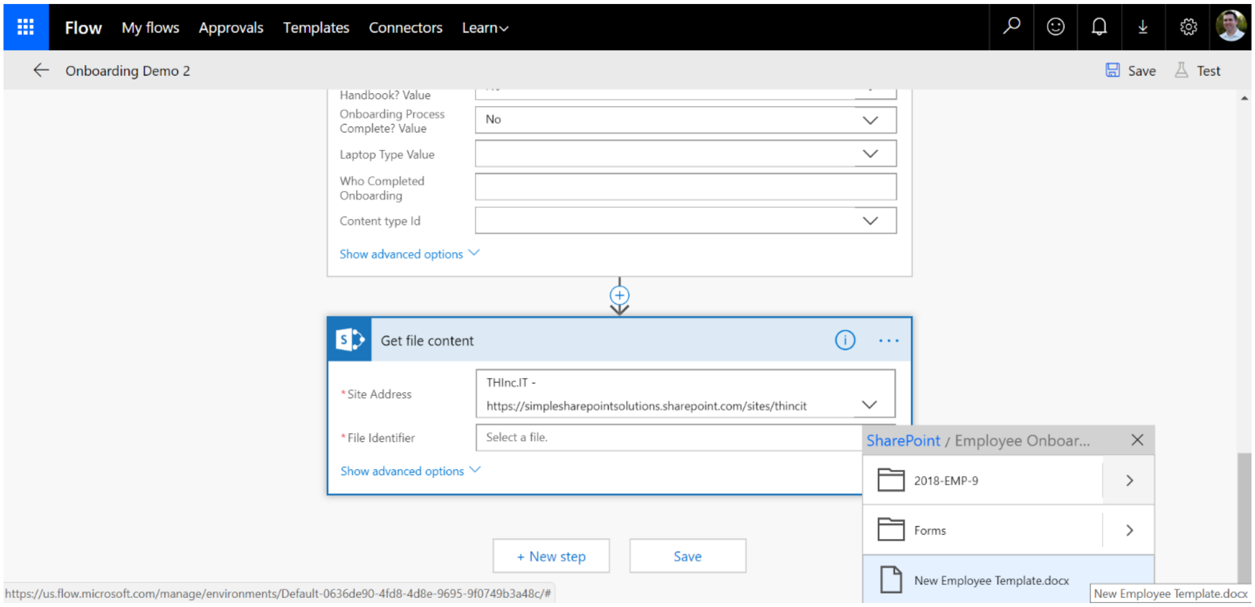
Item is a Record

Laptop Issued? Value

It is vital to the success of the Flow that you choose the "ID Tracking Number – Demo" column from "Update Item" and NOT from "When an item is created." For the Flow to succeed, we need to get the value from the previous step above that was created. If we pull from "When an item is created", it will pull a blank value.

Under the Update Item 2 box, click "New Step", "Add an Action", "SharePoint", then "Get file content."

In the "Get file content" box, choose the current site, then select the path of the file we created back in Step 2.

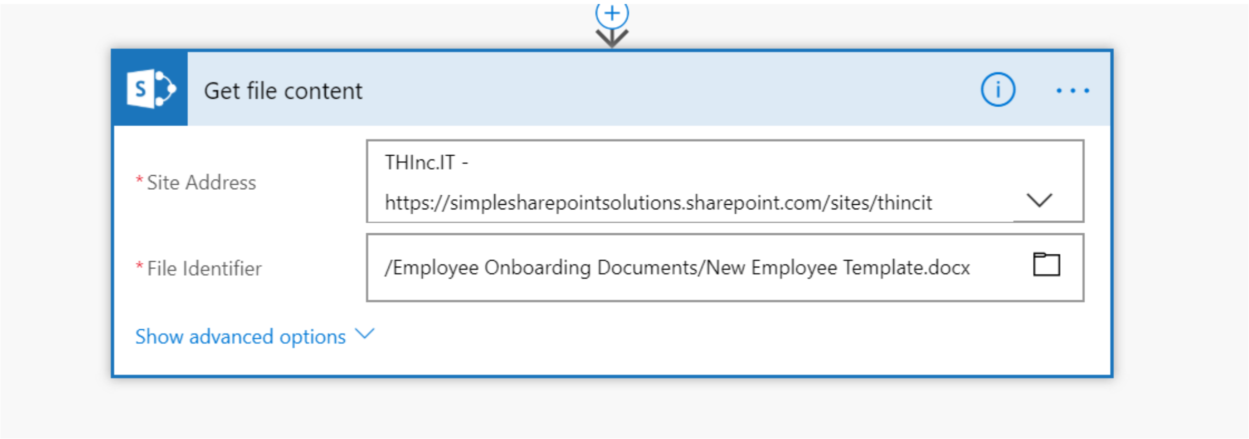


Handbook? Value
Onboarding Process Complete? Value: No
Laptop Type Value
Who Completed Onboarding
Content type Id

* Site Address: THInc.IT - https://simplesharepointsolutions.sharepoint.com/sites/thincit
* File Identifier: Select a file.

https://us.flow.microsoft.com/manage/environments/Default-0636de90-4fd8-4d8e-9695-9f0749b3a48c/#

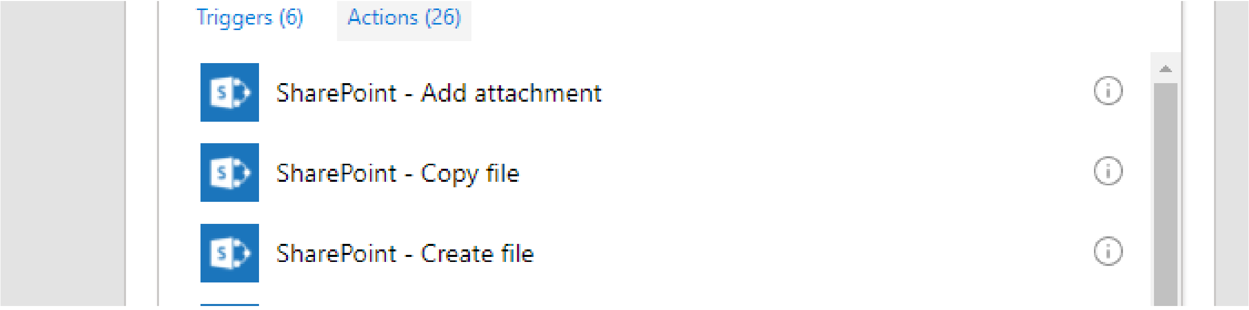
After selecting the document, the “Get file content” box should look like this:



* Site Address: THInc.IT - https://simplesharepointsolutions.sharepoint.com/sites/thincit
* File Identifier: /Employee Onboarding Documents/New Employee Template.docx

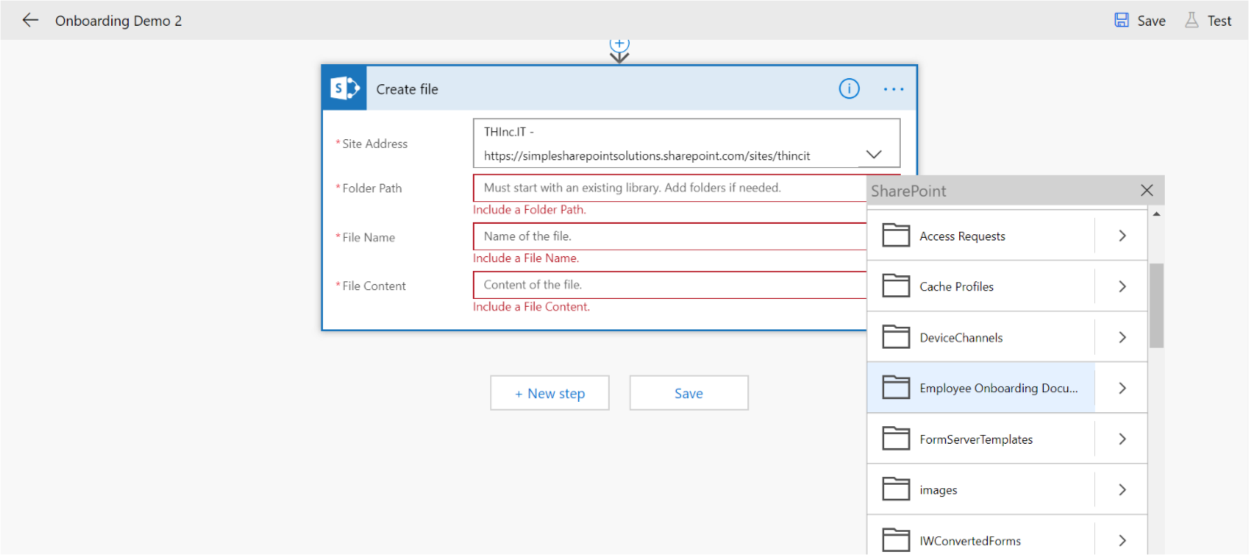
The last step of the Flow is creating the folder with this document template.

Under the “Get file content” box, click “New Step”, “Add an action”, “SharePoint”, then “Create File.”



In the “Create file” box, choose the current site.

In the “Folder Path” field, choose the Employee Onboarding Documents library.



*Site Address: THInc.IT - https://simplesharepointsolutions.sharepoint.com/sites/thincit

*Folder Path: Must start with an existing library. Add folders if needed.
Include a Folder Path.

*File Name: Name of the file.
Include a File Name.

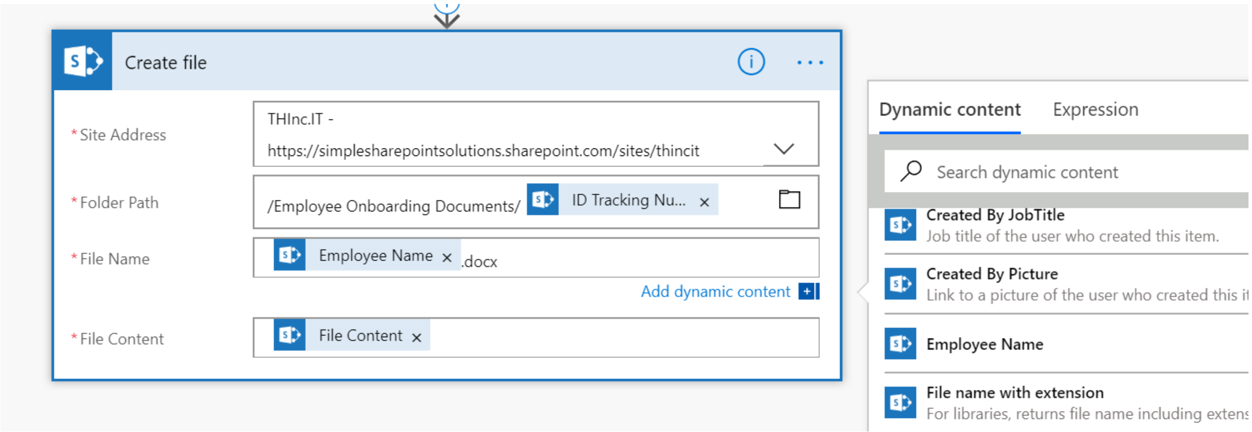
*File Content: Content of the file.
Include a File Content.

Click back into the Folder path and add a forward slash (/), then in the right pop-out menu choose “ID Tracking Number” under the Update item 2 section. Occasionally Flow will delete your forward slash and you will have to click to the left of the field and re-insert it.

In the file name field, choose “Employee Name” from the right pop-out menu and add .docx to the right of the field ID.

In the “File Content” field, choose “File Content” from the right pop-out menu.

The Create file box should now look like this:



*Site Address: THInc.IT - https://simplesharepointsolutions.sharepoint.com/sites/thincit

*Folder Path: /Employee Onboarding Documents/ ID Tracking Nu... ×

*File Name: Employee Name × .docx

*File Content: File Content ×

Created By JobTitle
Job title of the user who created this item.

Created By Picture
Link to a picture of the user who created this it

Employee Name

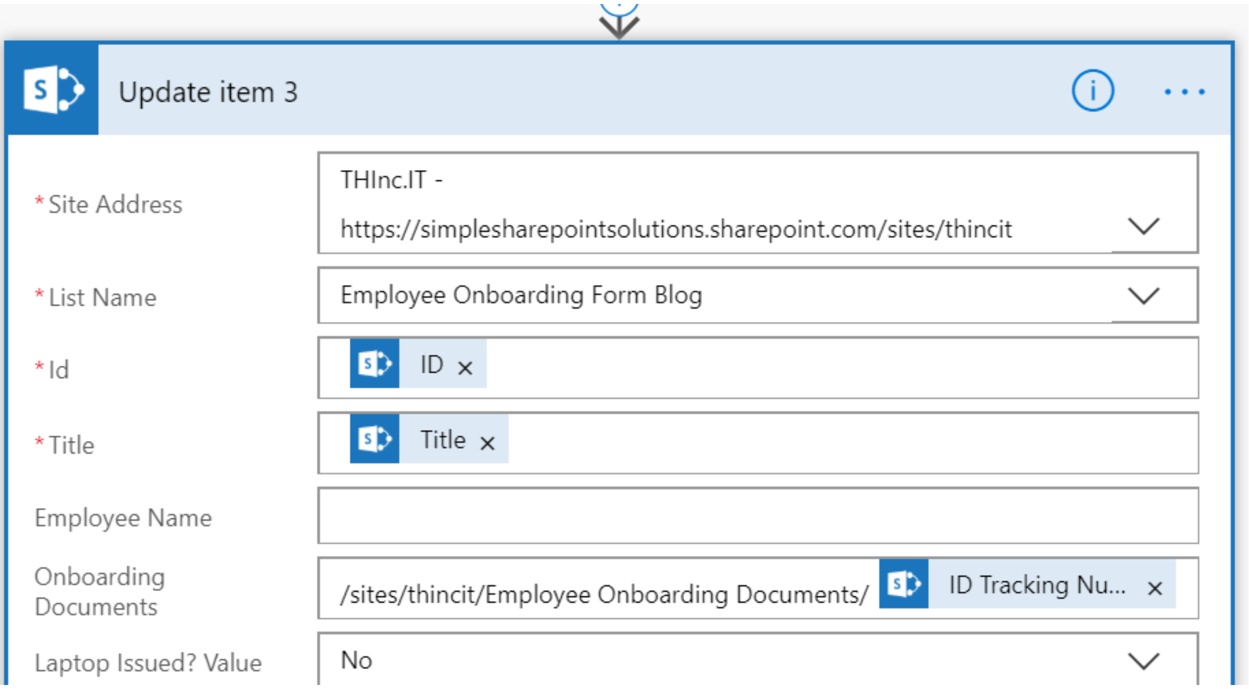
File name with extension
For libraries, returns file name including extens

Now we want to update the list item with a link to our Onboarding Documents folder.

Below our Create File box, click “New Step”, then “Add an action”, “SharePoint”, and then “Update item.”

Select the correct ID and Title fields from Update Item 2 in the appropriate places.

Then in the Onboarding Documents field, put the relative path of the URL to the library and at the end add the “ID Tracking Number”. It will look like this:



Update item 3

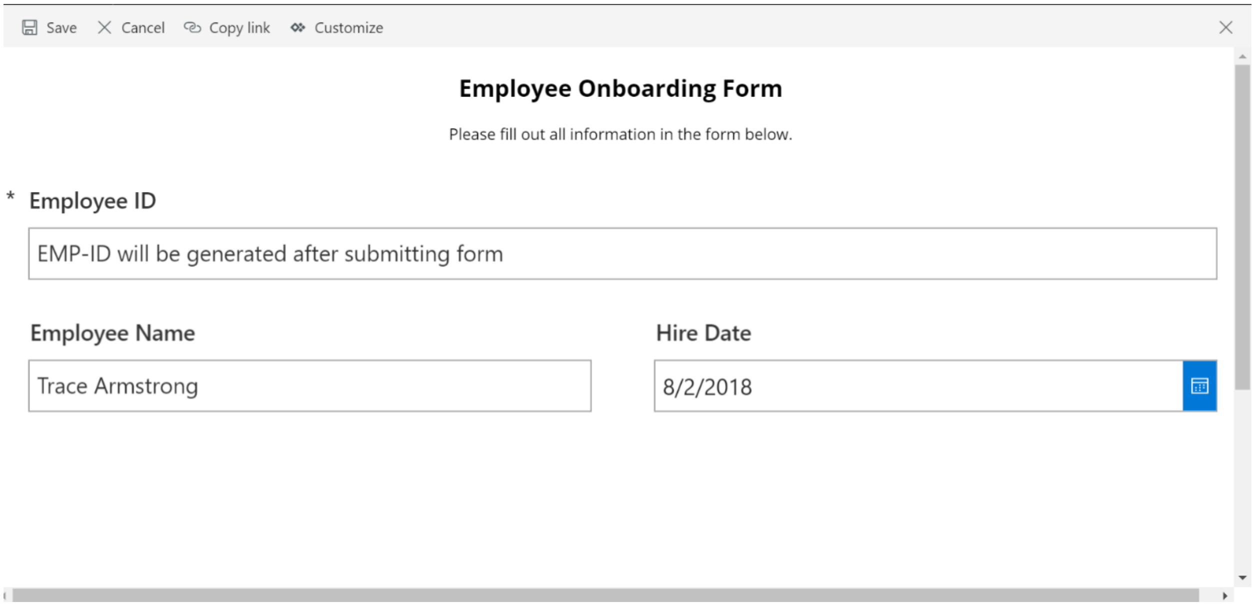
* Site Address	THInc.IT - https://simplesharepointsolutions.sharepoint.com/sites/thincit
* List Name	Employee Onboarding Form Blog
* Id	ID ×
* Title	Title ×
Employee Name	
Onboarding Documents	/sites/thincit/Employee Onboarding Documents/ ID Tracking Nu... ×
Laptop Issued? Value	No

Save the Flow and exit.

We have now created our custom form and our Microsoft Flow.

STEP 5: TEST THE ENTIRE SOLUTION

Navigate back to the list and create a new item filling out the three fields and clicking “Save.”



Employee Onboarding Form

Please fill out all information in the form below.

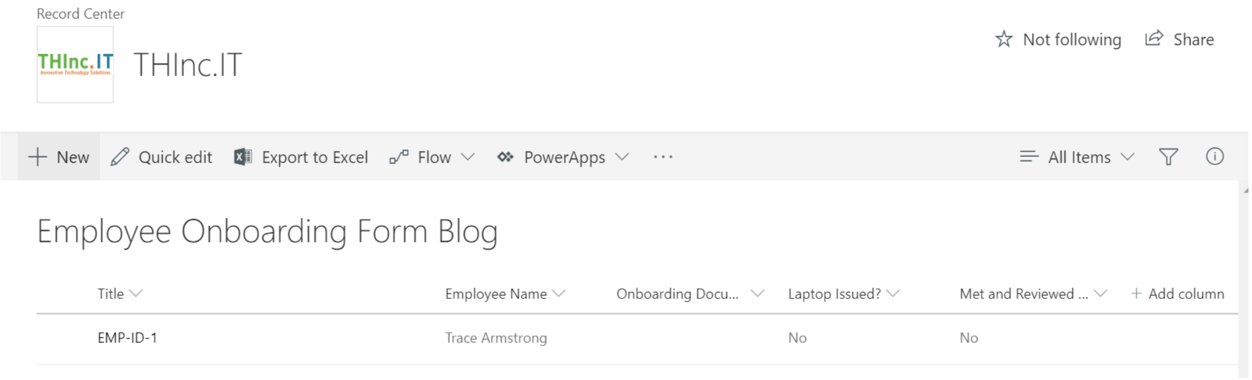
* Employee ID

EMP-ID will be generated after submitting form

Employee Name: Trace Armstrong

Hire Date: 8/2/2018

After the form closes, refresh your screen. You should see the Title field has been updated to reflect the custom ID.



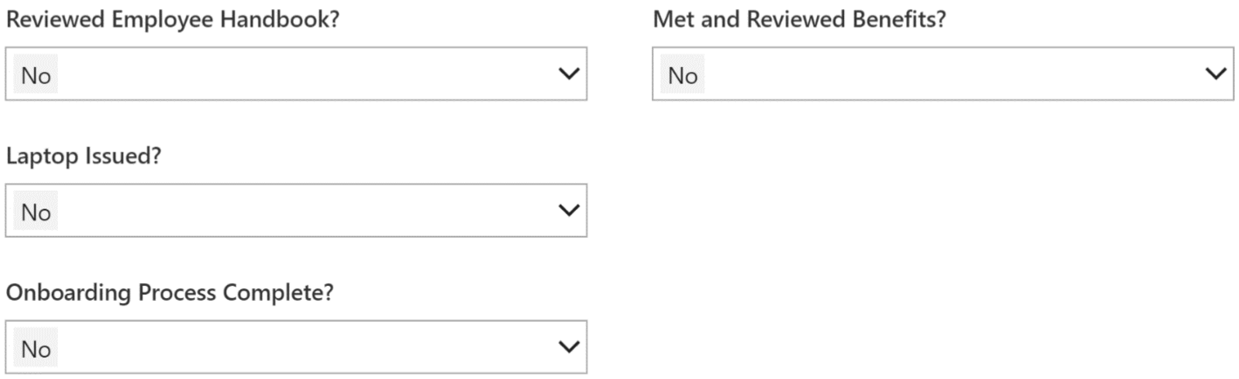
Employee Onboarding Form Blog

Title	Employee Name	Onboarding Docu...	Laptop Issued?	Met and Reviewed ...	+ Add column
EMP-ID-1	Trace Armstrong		No	No	

Now test the View and Edit Forms by clicking on the Title field.

Now click “Edit All” to test the conditional fields we created.

At first it will look like this:



Reviewed Employee Handbook?

No

Met and Reviewed Benefits?

No

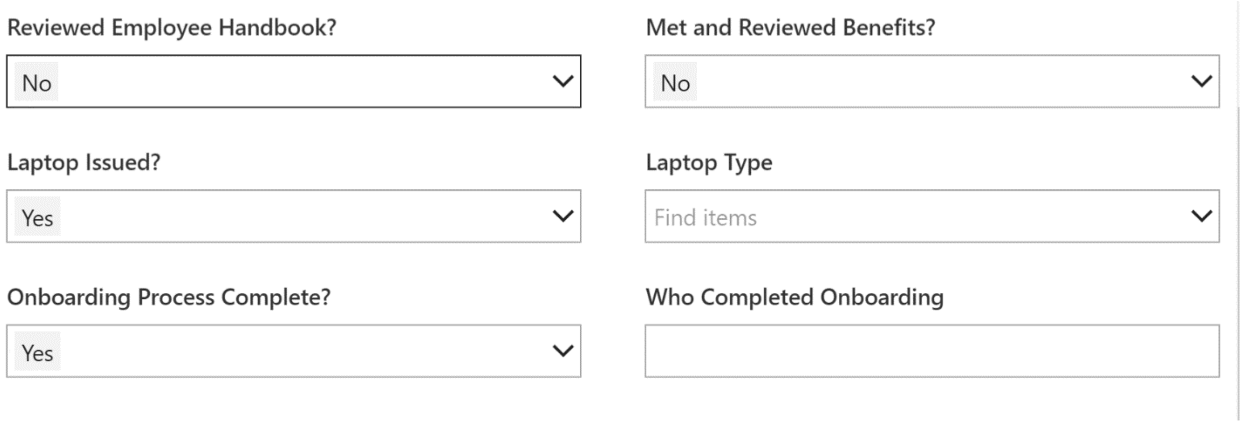
Laptop Issued?

No

Onboarding Process Complete?

No

Change Laptop Issued? and Onboarding Process Complete? To “Yes” and watch your fields magically appear.



Reviewed Employee Handbook?

No

Met and Reviewed Benefits?

No

Laptop Issued?

Yes

Laptop Type

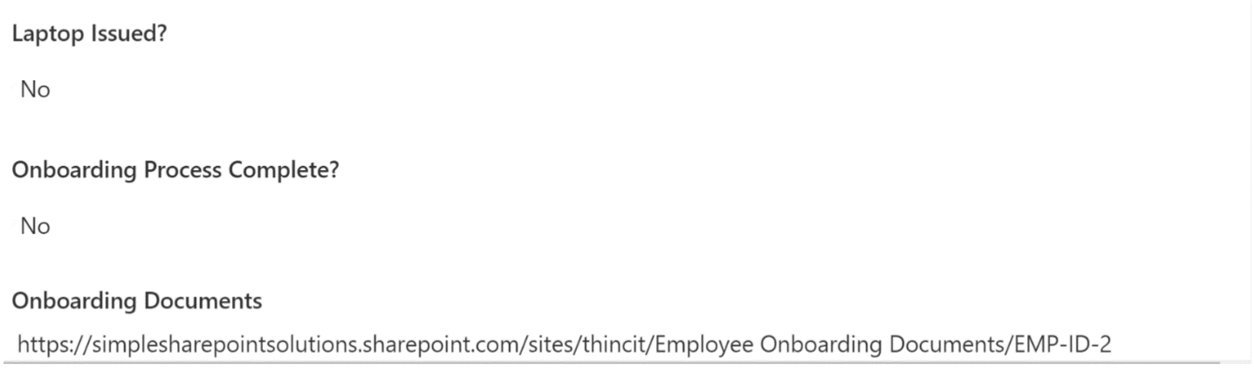
Find items

Onboarding Process Complete?

Yes

Who Completed Onboarding

On the View Screen we also see that a link to our Onboarding Documents folder has been updated!



Laptop Issued?

No

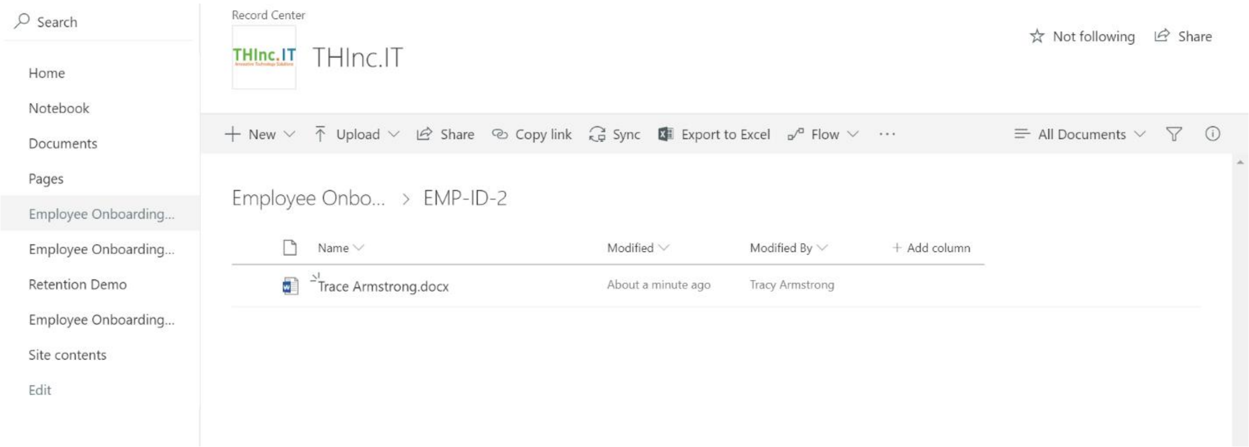
Onboarding Process Complete?

No

Onboarding Documents

https://simplesharepointsolutions.sharepoint.com/sites/thincit/Employee Onboarding Documents/EMP-ID-2

Now we can also click the link to the documents in the List View to verify our folder and file was created!



Name	Modified	Modified By
Trace Armstrong.docx	About a minute ago	Tracy Armstrong

Congratulations on creating your first PowerApps form and automated Flow!