



# PowerApps Bootcamp: Expert

Hands-On Lab Guide

Published (Jan 2026)

# Disclaimer

The content, code samples, and guidance provided in this lab document (the "Hands-On Lab Guide") are for informational and educational purposes only. The Materials are provided "as is" and reflect general practices and platform capabilities at the time of publication. While reasonable efforts have been made to ensure accuracy, no representation or warranty, express or implied, is made as to the completeness, accuracy, reliability, or suitability of the Materials for any purpose.

The Materials are not intended to be used as-is in production environments. It is the sole responsibility of the user to evaluate and test all solutions, configurations, and code examples in a controlled, non-production environment prior to any implementation in a live setting. Use of these Materials in a production environment is done entirely at the user's own risk.

No warranties or guarantees of any kind are provided, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or the absence of errors or defects. Under no circumstances shall the authors, contributors, or any affiliated parties be held liable for any damages, including but not limited to direct, indirect, incidental, consequential, or special damages, arising out of or in connection with the use of the Materials.

By using or adapting any portion of the lab document in any capacity, including in production environments, you expressly acknowledge and agree that you are assuming full responsibility and liability for such use, and that you release the authors and affiliated parties from any and all claims or liabilities that may arise as a result.

## Contents

Disclaimer .....	1
Exercise 1: Preparing data and creating a new solution .....	1
Objectives.....	1
Requires .....	1
Estimated time to complete this lab.....	1
Task 1: Importing Invoice Table Schema and Data.....	2
Task 2: Creating a Solution.....	9
Exercise 2: Creating the Canvas App and Designing the Home Screen .....	12
Objectives.....	12
Requires .....	12
Estimated time to complete this lab.....	12
Task 1: Creating the responsive app.....	13
Task 2: Editing Home Screen .....	15
Exercise 3: Connecting to Data and Displaying it.....	20
Objectives.....	20
Estimated time to complete this lab.....	20
Task 1: Connecting to the Dataverse Table .....	20
Task 2: Displaying the Data in a Table Control.....	22
Exercise 4: Filtering the Table .....	27
Objectives.....	27
Estimated time to complete this lab.....	27
Task 1: Creating Filter Controls.....	28
Task 2: Applying Filter .....	31
Exercise 5: Allow Users to Submit New Invoices.....	32
Objectives.....	32
Estimated time to complete this lab.....	32
Task 1: Duplicate the existing Screen.....	33
Task 2: Adding Controls to the Screen .....	35
Task 3: Extracting Values from Receipt Processor.....	40
Task 4: Submitting Extracted Values.....	43
Exercise 6: Creating Message Popups using Components .....	46
Objectives.....	46
Estimated time to complete this lab.....	46

Task 1: Adding a New Component with Custom Properties.....	47
Task 2: Adding Controls into the Component.....	52
Task 3: Adding Popup Component to Submission Screen.....	57

# Exercise 1: Preparing data and creating a new solution

In this exercise, we will create a new solution to store all the flows and apps related to our app.

## Objectives

After completing this lab, you will be able to:

- Create a new solution for your Power Platform developments

## Requires

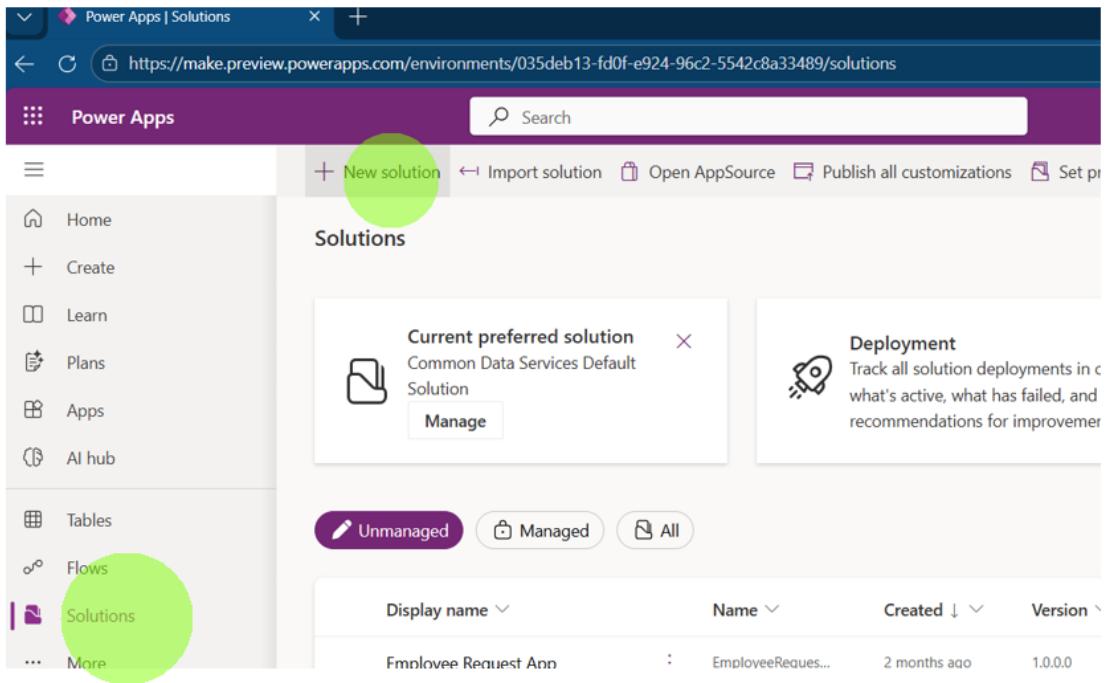
- InvoiceProcessingAppImportData\_1\_0\_0\_1.zip
- invoices\_last3months\_simple\_approval.xlsx

## Estimated time to complete this lab

15 mins

## Task 1: Importing Invoice Table Schema and Data

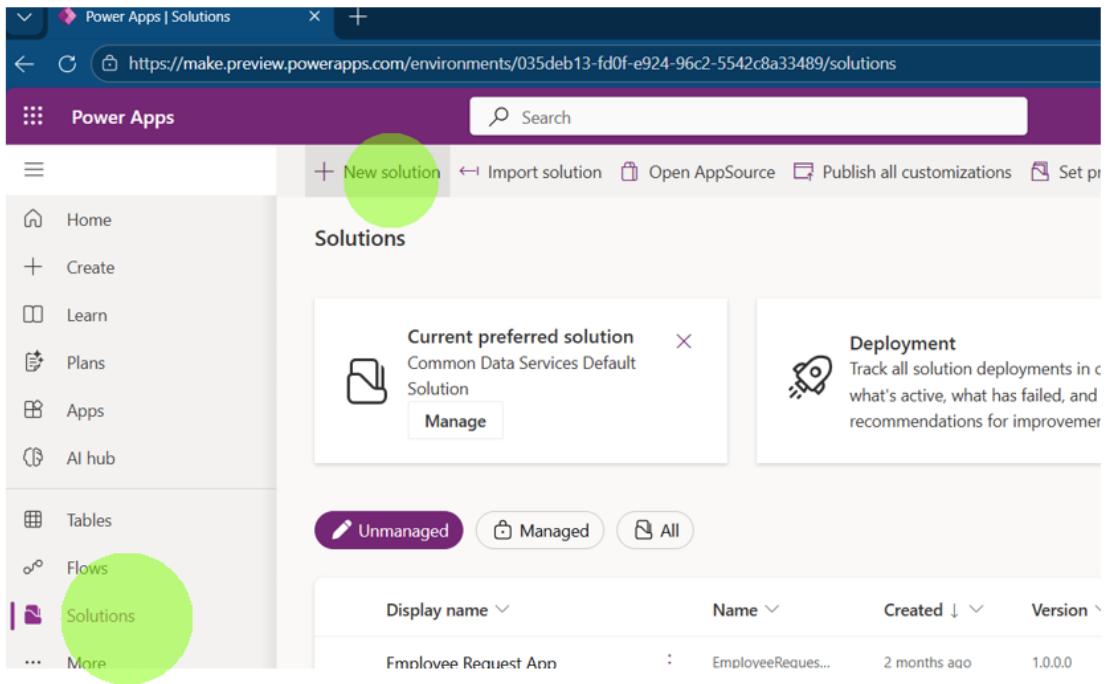
1. Go to [make.powerapps.com](https://make.powerapps.com/).



The screenshot shows the Microsoft Power Apps | Solutions interface. The left sidebar has a purple header with 'Power Apps' and a search bar. Below it are links for Home, Create, Learn, Plans, Apps, AI hub, Tables, Flows, and Solutions. The 'Solutions' link is highlighted with a green circle. The main area is titled 'Solutions' and shows a 'Current preferred solution' card for 'Common Data Services Default Solution' with a 'Manage' button. To the right is a 'Deployment' card with a lightbulb icon. Below these are three filter buttons: 'Unmanaged' (purple), 'Managed' (light blue), and 'All'. A table lists solutions with columns for Display name, Name, Created, and Version. One row is visible: 'Employee Request App' (Name), 'EmployeeReques...' (Created: 2 months ago), and '1.0.0.0' (Version).

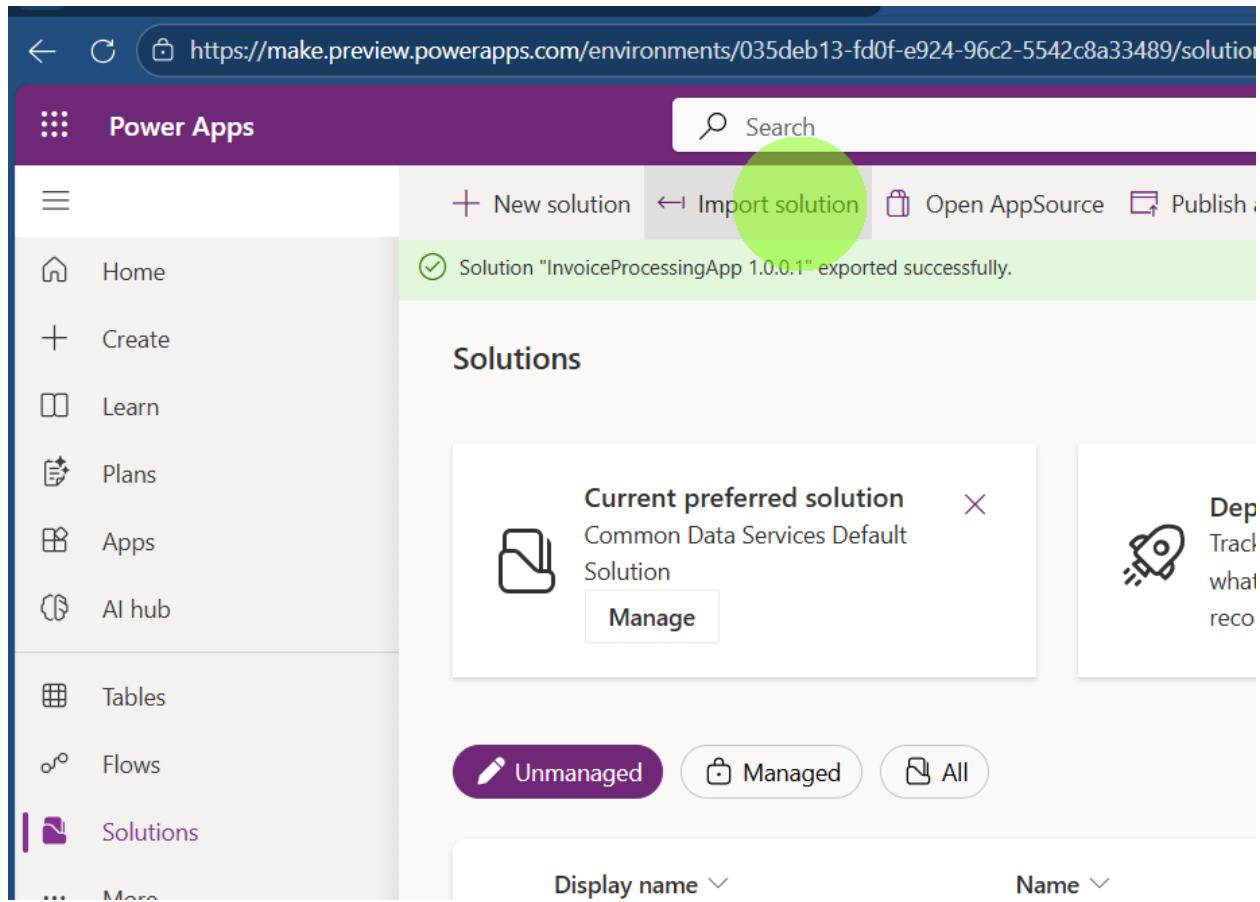
Display name	Name	Created	Version
Employee Request App	EmployeeReques...	2 months ago	1.0.0.0

2. Click Solutions.



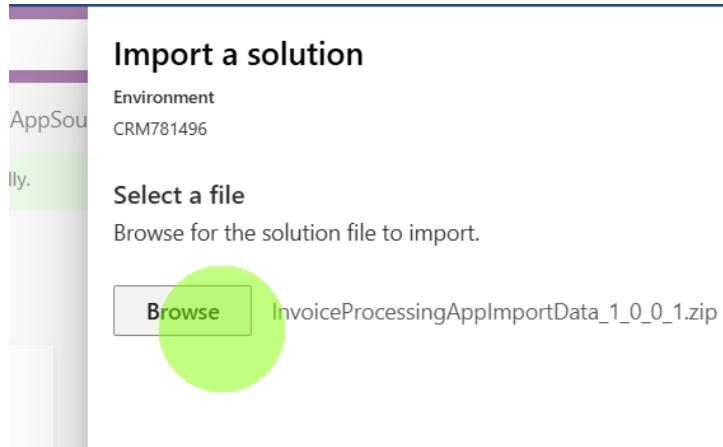
The screenshot shows the 'Power Apps | Solutions' interface. The left sidebar includes options like Home, Create, Learn, Plans, Apps, AI hub, Tables, Flows, Solutions (which is highlighted with a green circle), and More. The main area displays a 'Current preferred solution' card for 'Common Data Services Default Solution' with a 'Manage' button. It also features a 'Deployment' section for tracking solutions. Below these are buttons for Unmanaged, Managed, and All. A table lists solutions, showing one entry: Employee Request App, EmployeeReques..., Created 2 months ago, Version 1.0.0.0. The top navigation bar includes links for New solution, Import solution, Open AppSource, Publish all customizations, and Set pr...

3. Click Import Solution.



The screenshot shows the Microsoft Power Apps Solutions page. On the left, there's a navigation menu with options like Home, Create, Learn, Plans, Apps, AI hub, Tables, Flows, and Solutions. The Solutions option is currently selected. At the top right, there are buttons for New solution, Import solution (which has a green circle around it), Open AppSource, and Publish. A success message at the top right says "Solution 'InvoiceProcessingApp 1.0.0.1' exported successfully." Below this, a card displays the "Current preferred solution" as "Common Data Services Default Solution" with a "Manage" button. There are also tabs for Unmanaged, Managed, and All. The main area lists solutions with columns for Display name and Name.

4. Click Browse and select InvoiceProcessingApplImportData\_1\_0\_0\_1.zip from lab files.

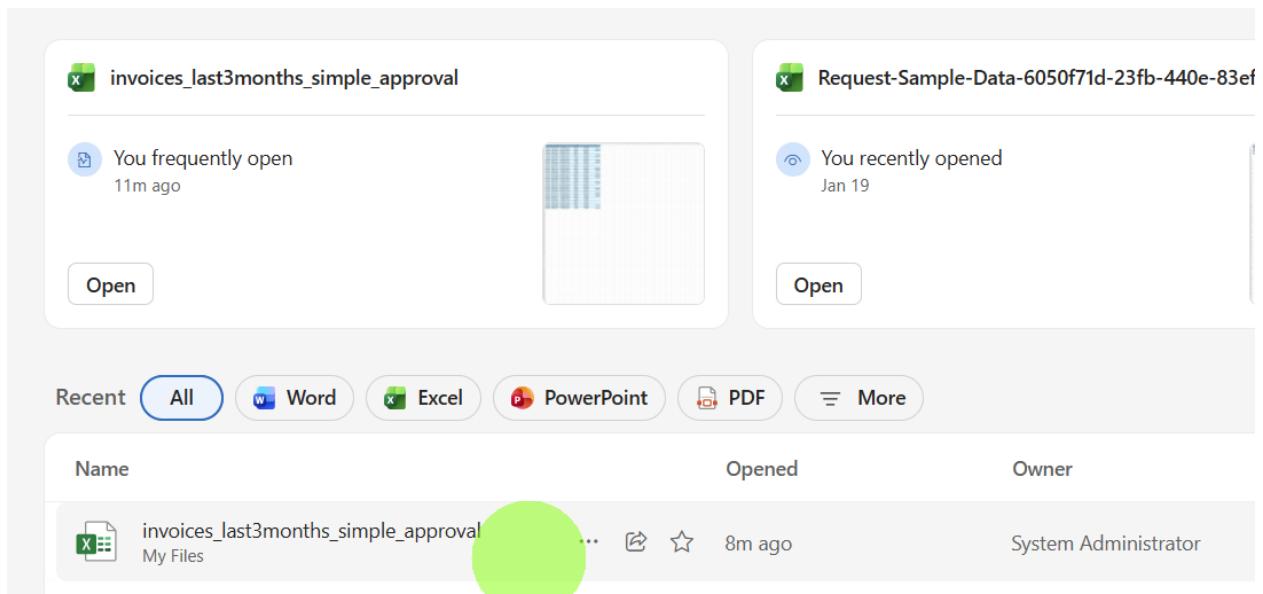


The screenshot shows the "Import a solution" dialog. It has a sidebar with "AppSource" and "Solutions". The main area is titled "Import a solution" and shows the "Environment" as "CRM781496". Below this, there's a section titled "Select a file" with the instruction "Browse for the solution file to import." A "Browse" button is highlighted with a green circle. To its right, the file "InvoiceProcessingApplImportData\_1\_0\_0\_1.zip" is listed.

5. Click Next and Import. The solution will take awhile to import.

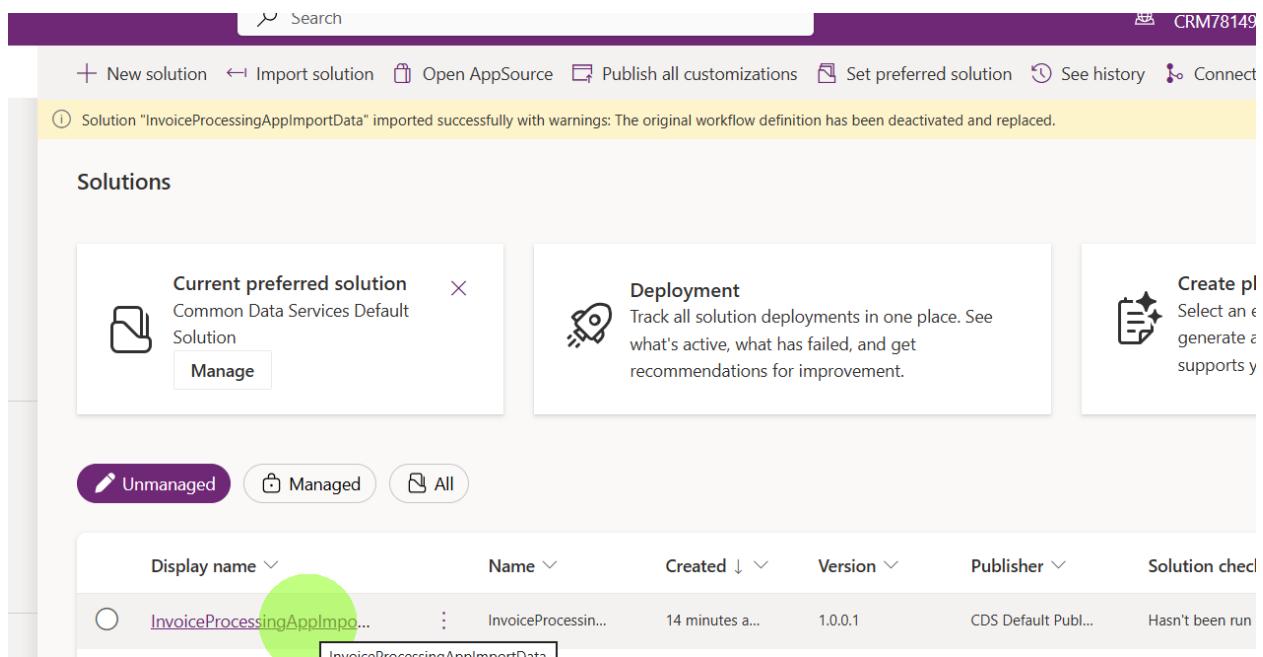


6. Meanwhile upload invoices\_last3months\_simple\_approval.xlsx to your OneDrive folder.



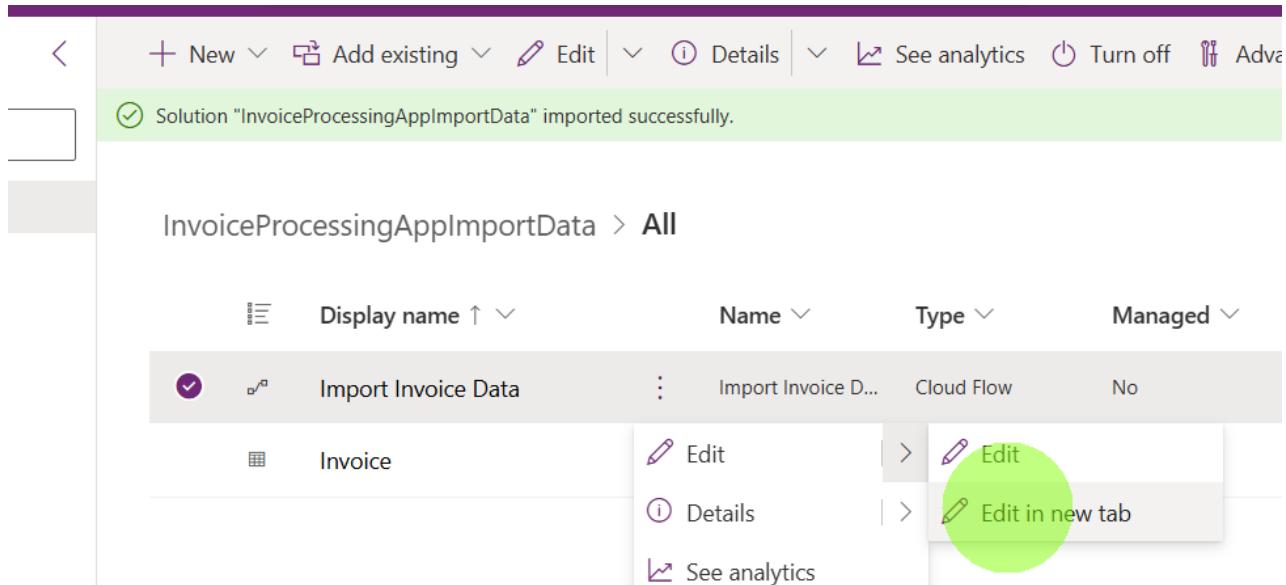
Name	Opened	Owner
invoices_last3months_simple_approval	8m ago	System Administrator

7. Once the solution is imported, you will receive a message as below. Click into the imported solution.



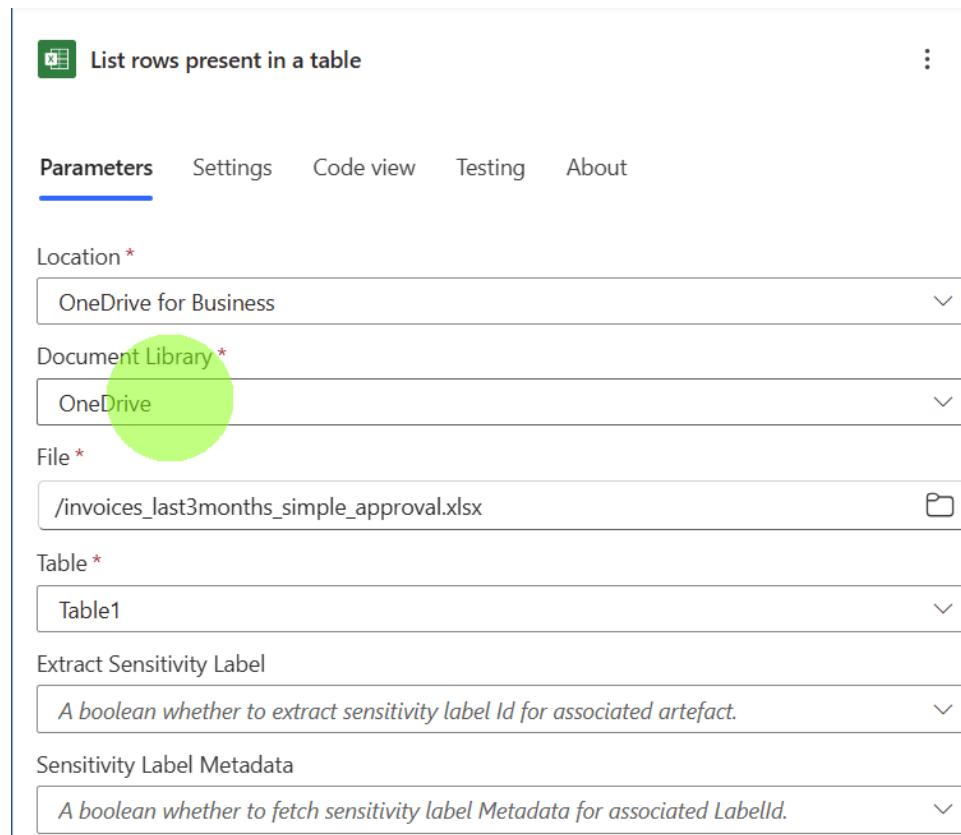
The screenshot shows the Microsoft Power Platform Solutions page. At the top, there is a navigation bar with links for 'New solution', 'Import solution', 'Open AppSource', 'Publish all customizations', 'Set preferred solution', 'See history', and 'Connect'. Below the navigation bar, a message box displays: 'Solution "InvoiceProcessingApplImportData" imported successfully with warnings: The original workflow definition has been deactivated and replaced.' The main area is titled 'Solutions' and contains three cards: 'Current preferred solution' (Common Data Services Default Solution), 'Deployment' (Track all solution deployments in one place), and 'Create pl' (Select an e... generate a... supports y...). Below the cards, there are filter buttons for 'Unmanaged', 'Managed' (which is selected), and 'All'. A table lists solutions, with the imported solution 'InvoiceProcessingApplImportData' highlighted by a green circle. The table columns include 'Display name', 'Name', 'Created', 'Version', 'Publisher', and 'Solution check'. The 'InvoiceProcessingApplImportData' row shows 'InvoiceProcessin...', '14 minutes ago', '1.0.0.1', 'CDS Default Publ...', and 'Hasn't been run'.

8. Once inside the solution, open Import Invoice Data Flow to edit in a new tab.

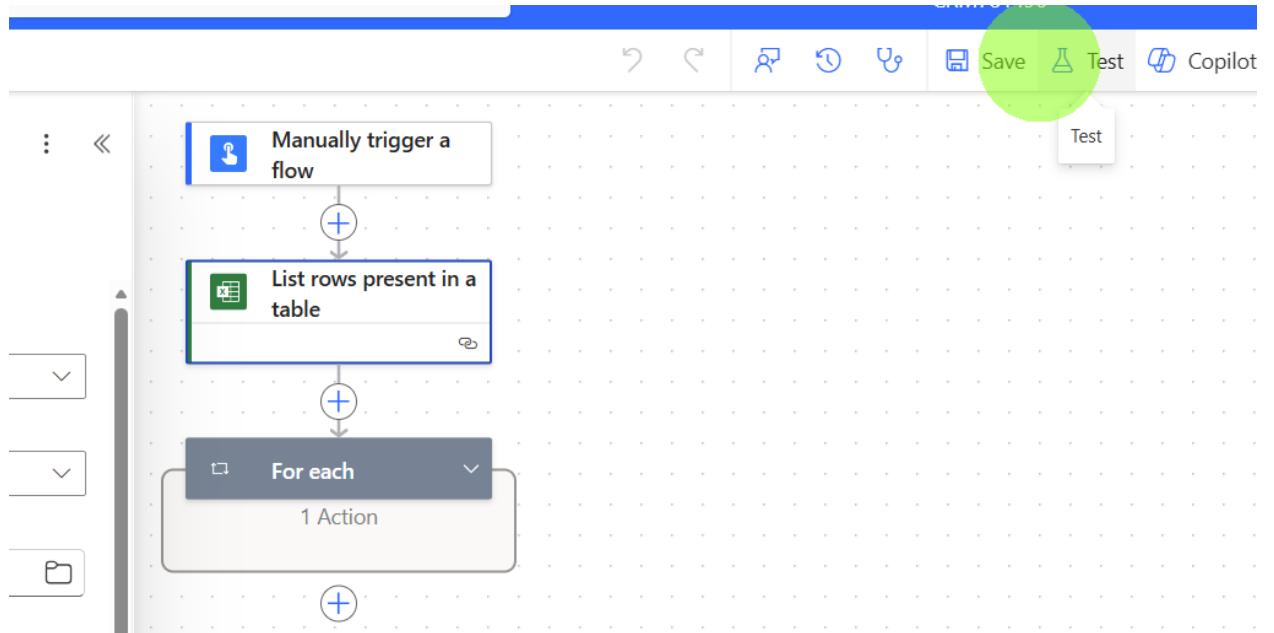


The screenshot shows the Microsoft Power Platform solution editor for the 'InvoiceProcessingApplImportData' solution. The top navigation bar includes 'New', 'Add existing', 'Edit', 'Details', 'See analytics', 'Turn off', and 'Advanced'. A message box at the top states: 'Solution "InvoiceProcessingApplImportData" imported successfully.' The main area is titled 'InvoiceProcessingApplImportData > All' and displays a list of flows. The first flow, 'Import Invoice Data', is selected and shown in detail. The flow card includes 'Display name' (Import Invoice Data), 'Name' (Import Invoice D...), 'Type' (Cloud Flow), and 'Managed' (No). To the right of the flow card, there are three actions: 'Edit' (highlighted with a green circle), 'Details', and 'See analytics'. Below the flow card, there is a link 'Invoice' with three actions: 'Edit', 'Details', and 'See analytics'. The 'Edit' action for the 'Edit' link is also highlighted with a green circle.

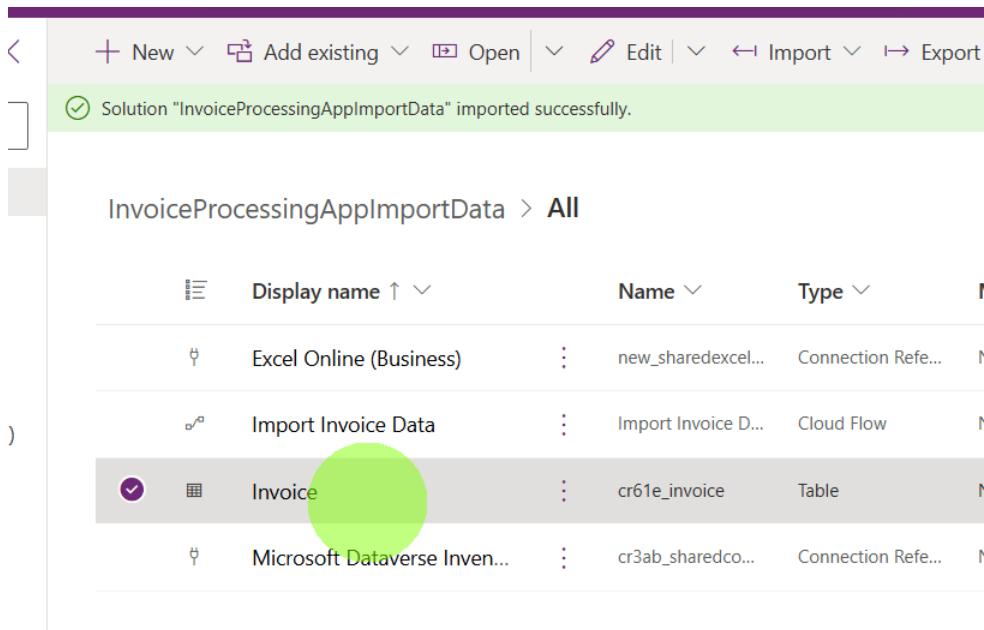
9. Expand List rows present in a table step and select "My Drive" for Document Library.



10. Save and Test run the flow to import data.

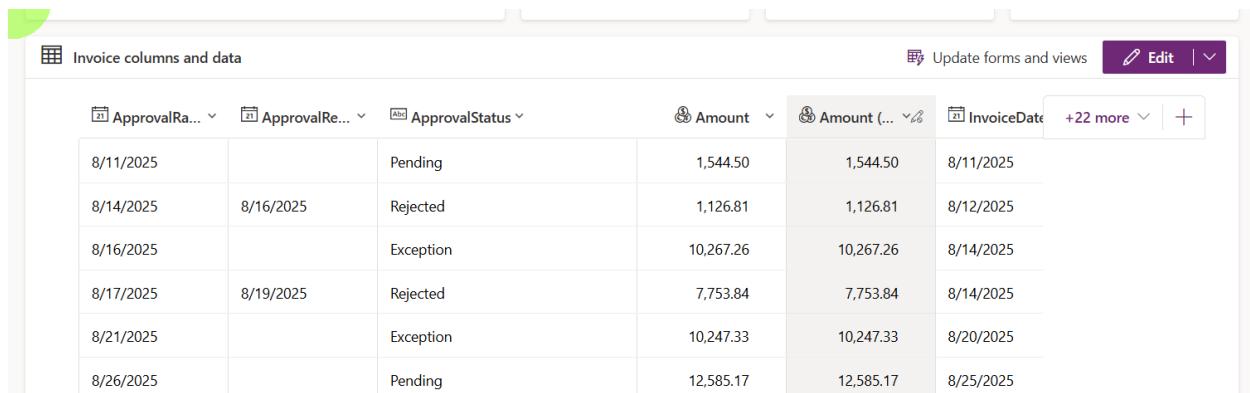


11. Once run, Invoice table should have data inside.



The screenshot shows the Microsoft Power Platform canvas interface. At the top, there are navigation links: '+ New', 'Add existing', 'Open', 'Edit', 'Import', and 'Export'. A green notification bar at the top indicates: 'Solution "InvoiceProcessingAppImportData" imported successfully.' Below the notification, the title 'InvoiceProcessingAppImportData > All' is displayed. A table list is shown with the following items:

Display name ↑	Name	Type	M
Excel Online (Business)	new_sharedexcel...	Connection Refe...	N
Import Invoice Data	Import Invoice D...	Cloud Flow	N
Invoice	cr61e_invoice	Table	N
Microsoft Dataverse Inven...	cr3ab_sharedco...	Connection Refe...	N



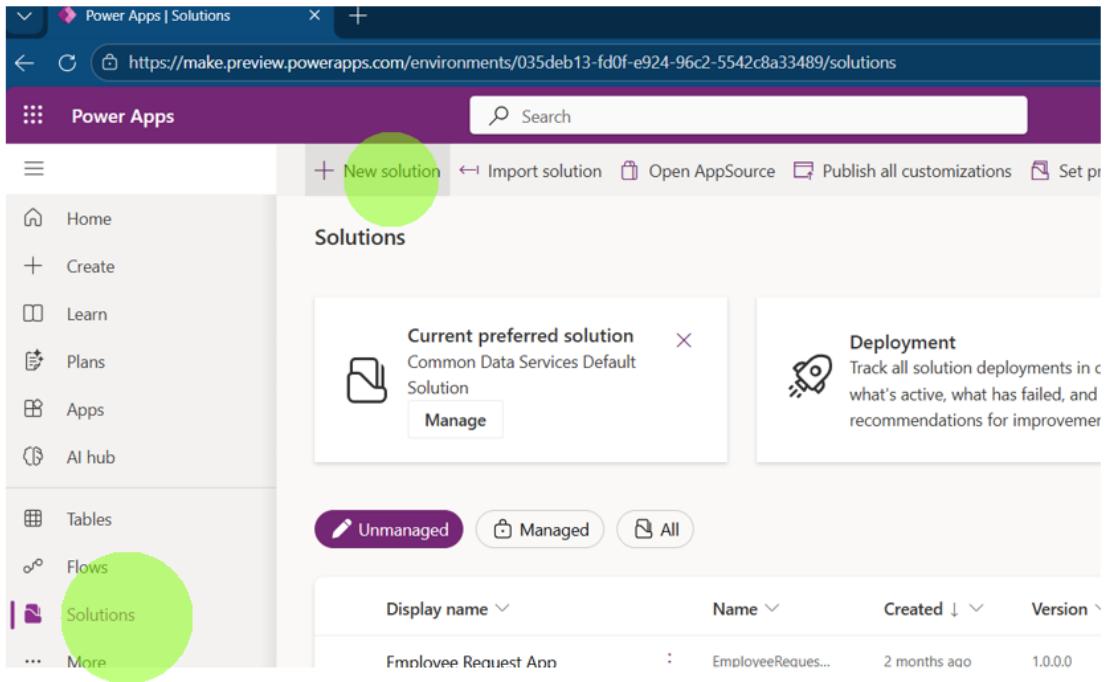
The screenshot shows the 'Invoice' table in the Microsoft Power Platform canvas. The table has the following columns: ApprovalReason, ApprovalRequester, ApprovalStatus, Amount, Amount (USD), and InvoiceDate. The data rows are:

ApprovalReason	ApprovalRequester	ApprovalStatus	Amount	Amount (USD)	InvoiceDate
8/11/2025		Pending	1,544.50	1,544.50	8/11/2025
8/14/2025	8/16/2025	Rejected	1,126.81	1,126.81	8/12/2025
8/16/2025		Exception	10,267.26	10,267.26	8/14/2025
8/17/2025	8/19/2025	Rejected	7,753.84	7,753.84	8/14/2025
8/21/2025		Exception	10,247.33	10,247.33	8/20/2025
8/26/2025		Pending	12,585.17	12,585.17	8/25/2025

----- End of Task -----

## Task 2: Creating a Solution

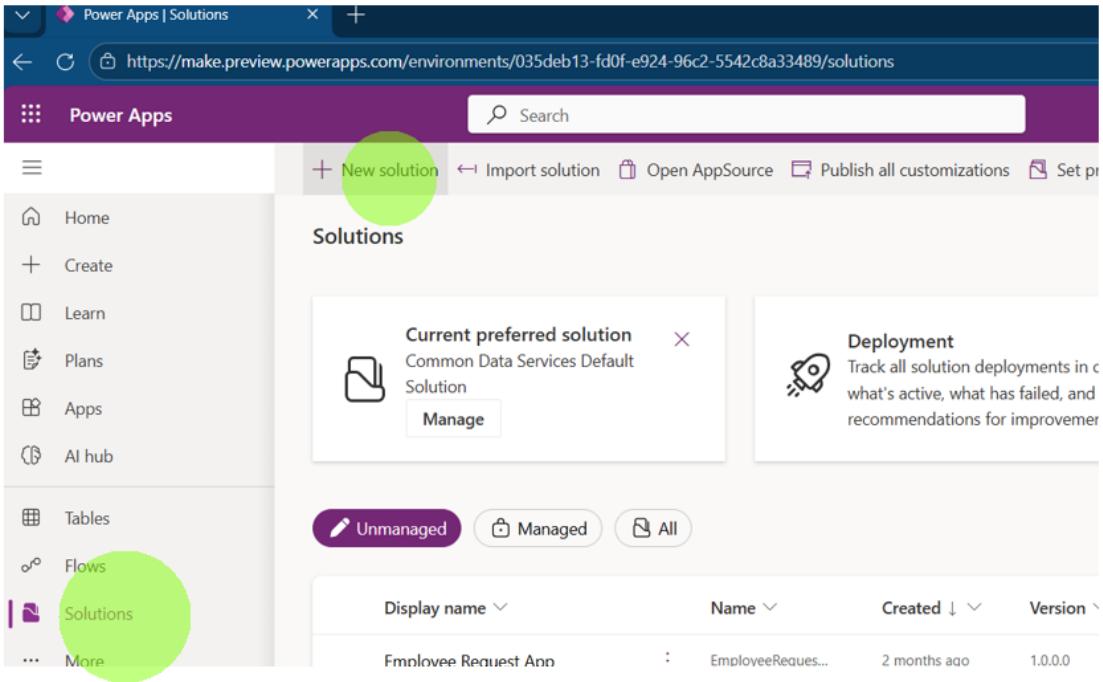
12. Go to [make.powerapps.com.](https://make.powerapps.com/)



The screenshot shows the Microsoft Power Apps Solutions interface. The top navigation bar includes a 'New solution' button, which is highlighted with a green circle. Below the navigation bar is a search bar and several other navigation links: 'Import solution', 'Open AppSource', 'Publish all customizations', and 'Set pri'. The main content area is titled 'Solutions' and displays the 'Current preferred solution' as 'Common Data Services Default Solution'. It also features a 'Deployment' section for tracking solution deployments. Below these sections is a table listing solutions, with one entry visible: 'Employee Request App' (Display name), 'EmployeeReques...' (Name), '2 months ago' (Created), and '1.0.0.0' (Version). The left sidebar contains links for Home, Create, Learn, Plans, Apps, AI hub, Tables, Flows, and Solutions, with 'Solutions' also highlighted with a green circle.

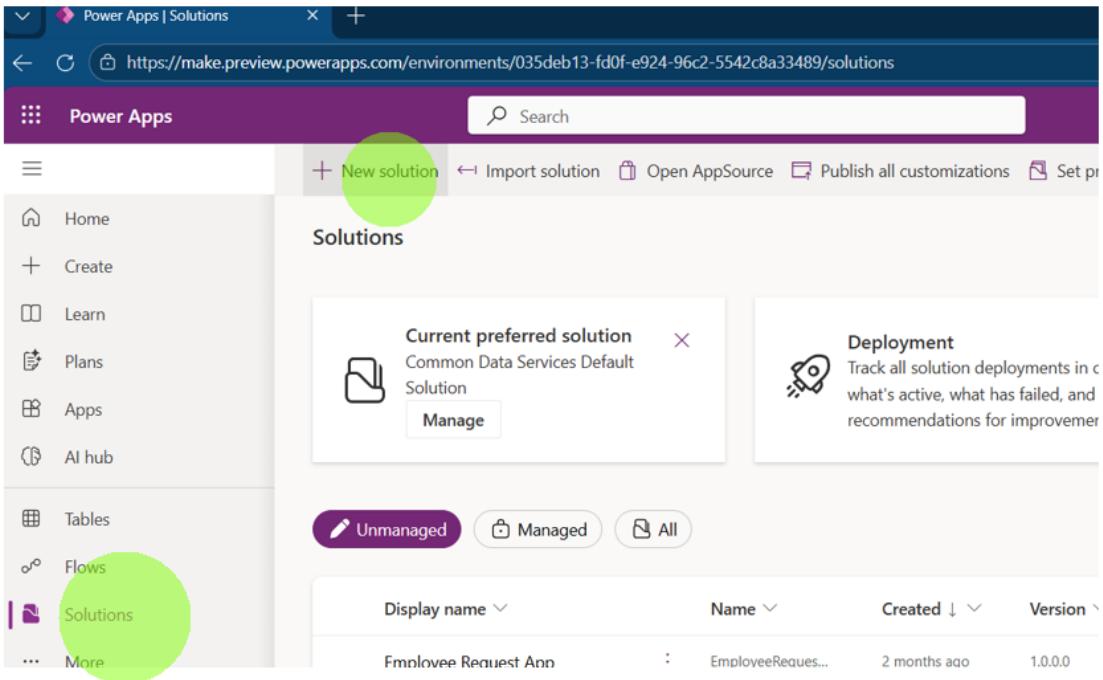
Display name	Name	Created	Version
Employee Request App	EmployeeReques...	2 months ago	1.0.0.0

13. Click Solutions.



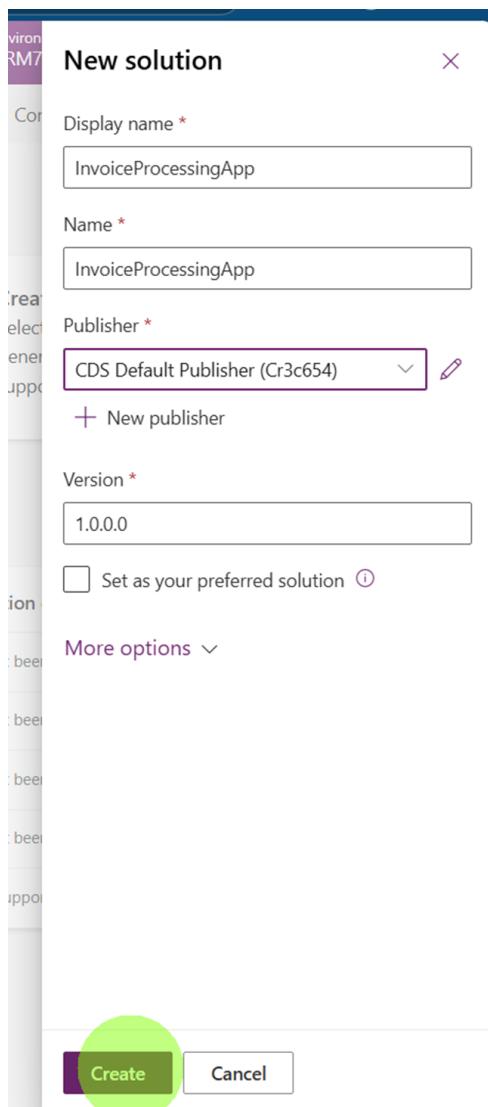
The screenshot shows the Microsoft Power Apps Solutions interface. On the left, there's a navigation sidebar with links like Home, Create, Learn, Plans, Apps, AI hub, Tables, Flows, and Solutions. The 'Solutions' link is highlighted with a green circle. The main area is titled 'Solutions' and displays a summary of the current preferred solution ('Common Data Services Default Solution') and deployment status. Below this, there are tabs for 'Unmanaged', 'Managed', and 'All'. A table lists solutions based on display name, name, created date, and version. One entry is visible: 'Employee Request App' (Name: EmployeeReques..., Created: 2 months ago, Version: 1.0.0.0).

14. Click + New Solution.



This screenshot is identical to the one above, showing the Microsoft Power Apps Solutions page. The 'Solutions' link in the sidebar is highlighted with a green circle. The main content area shows the current preferred solution, deployment status, and a table of existing solutions. The key difference is that the '+ New solution' button in the top navigation bar is also highlighted with a green circle, indicating the next step in the process.

15. Enter Solution Details as below. For Publisher, you will find a default publisher with a different ID to select.



New solution

Display name \*

Name \*

Publisher \*

CDS Default Publisher (Cr3c654)

+ New publisher

Version \*

Set as your preferred solution ⓘ

More options ▾

**Create**

16. You will be navigated to the empty Solution page.

----- End of Task -----

# Exercise 2: Creating the Canvas App and Designing the Home Screen

In this next exercise, we will create the canvas app from the solution and design a home screen for user navigation.

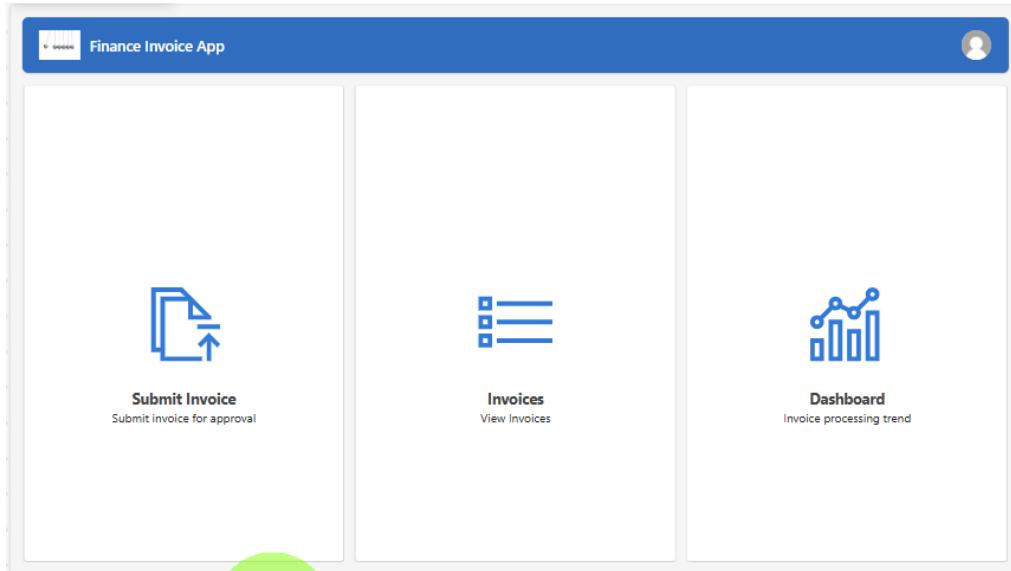
## Objectives

After completing this lab, you will be able to:

- Use built-in responsive layout templates
- Change the app settings to be more responsive

## Requires

- App Icons



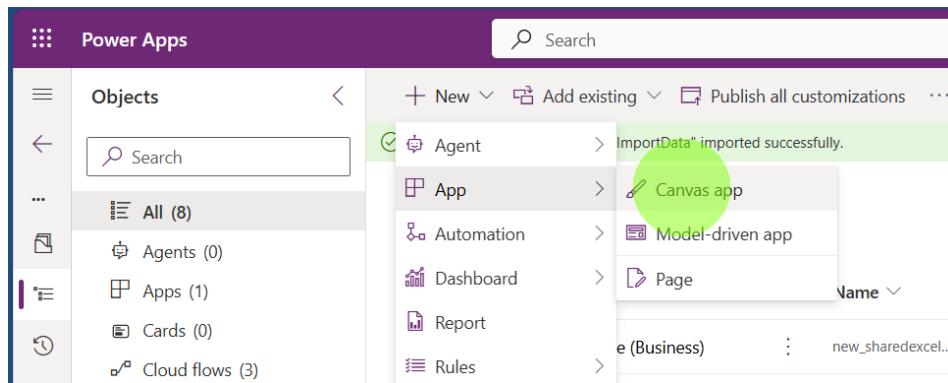
## Estimated time to complete this lab

20 mins

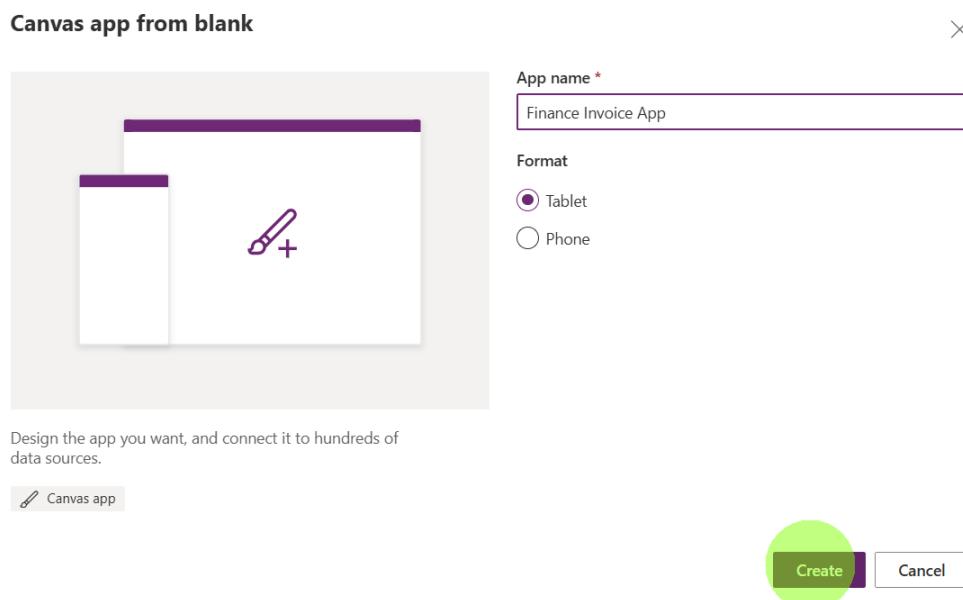
## Task 1: Creating the responsive app

In this Task you will learn how to create a new app from scratch and how to make it responsive.

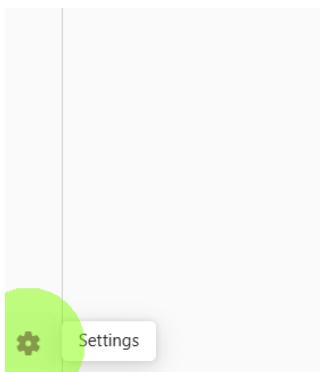
1. In solution, click +New > App > Canvas app



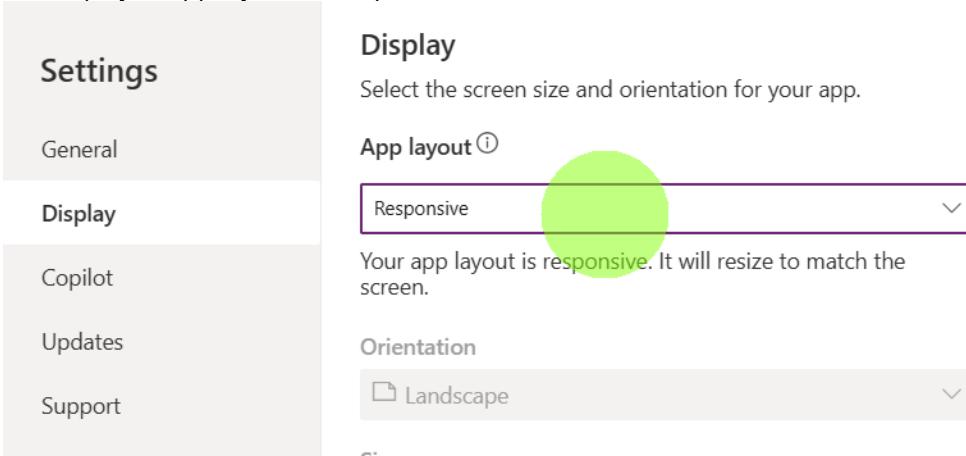
Provide a name and click Create.



2. Once in the app designer, click setting icon at the bottom right corner.



Set Display > App layout to Responsive.



**Settings**

General

Display

Copilot

Updates

Support

**Display**

Select the screen size and orientation for your app.

**App layout** ⓘ

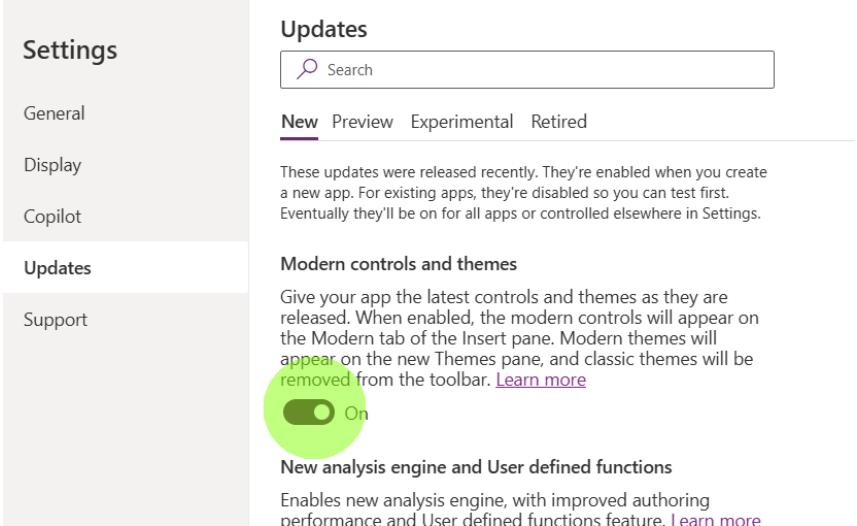
Responsive

Your app layout is **responsive**. It will resize to match the screen.

**Orientation**

Landscape

Switch on Modern controls and themes under Updates.



**Settings**

General

Display

Copilot

**Updates**

Support

**Updates**

Search

New Preview Experimental Retired

These updates were released recently. They're enabled when you create a new app. For existing apps, they're disabled so you can test first. Eventually they'll be on for all apps or controlled elsewhere in Settings.

**Modern controls and themes**

Give your app the latest controls and themes as they are released. When enabled, the modern controls will appear on the Modern tab of the Insert pane. Modern themes will appear on the new Themes pane, and classic themes will be removed from the toolbar. [Learn more](#)

On

**New analysis engine and User defined functions**

Enables new analysis engine, with improved authoring performance and User defined functions feature. [Learn more](#)

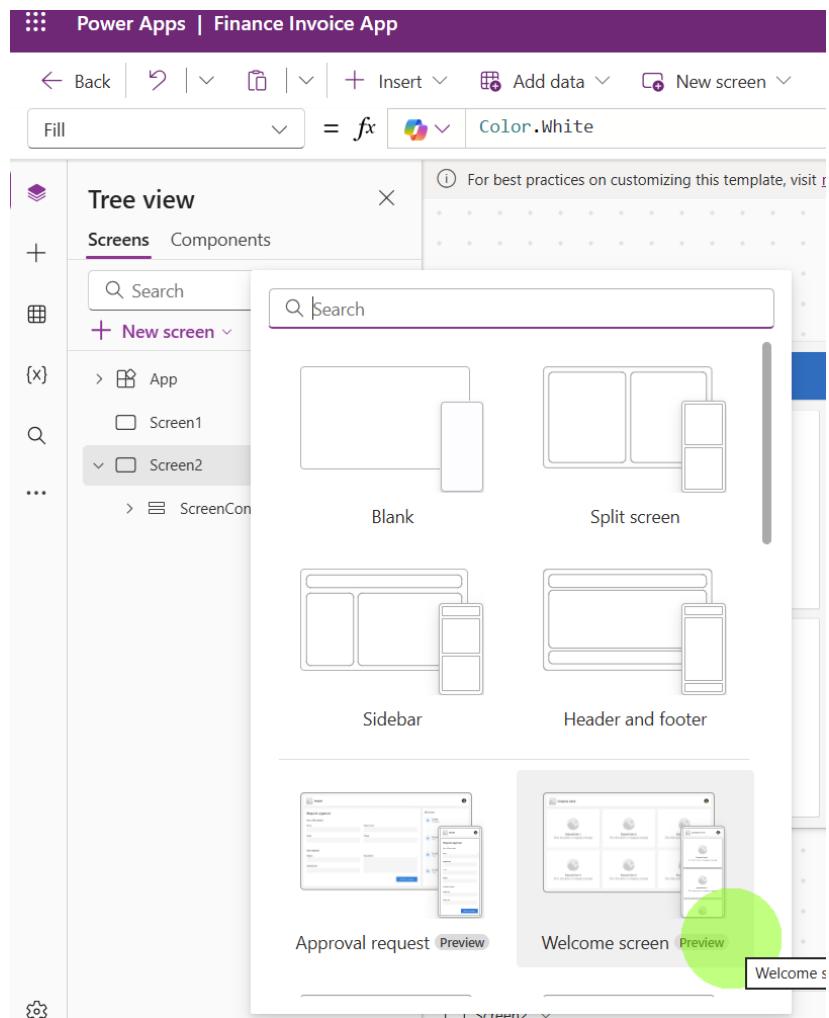
Close the Settings window.

----- End of Task -----

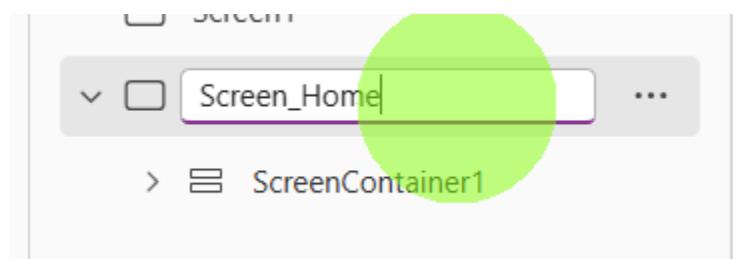
## Task 2: Editing Home Screen

In this Task you will learn how to use built-in Power App screens.

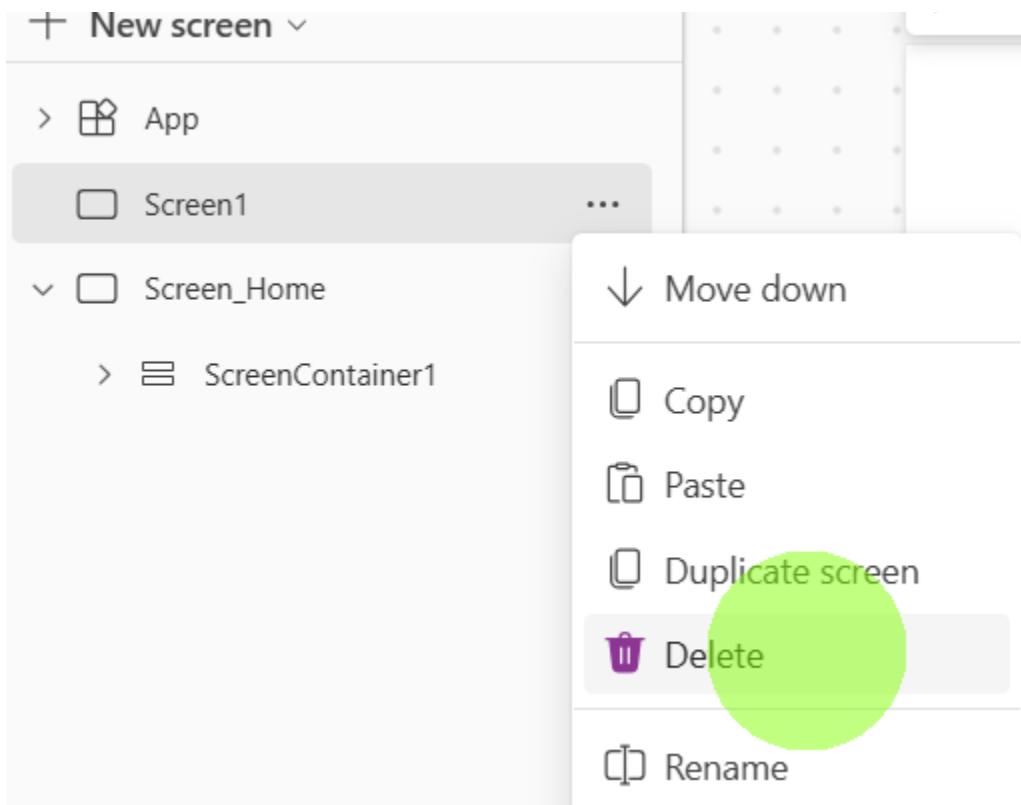
1. Click + New screen > Welcome screen.



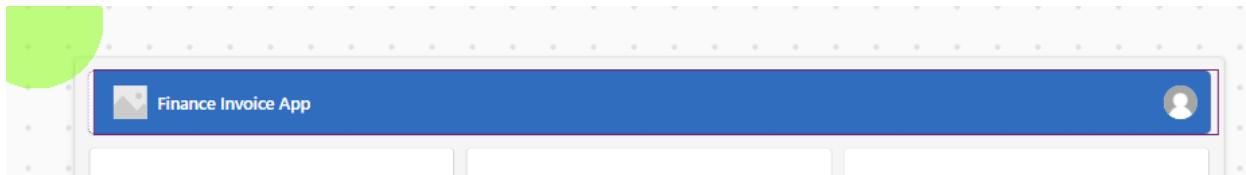
Rename the screen to Screen\_Home.



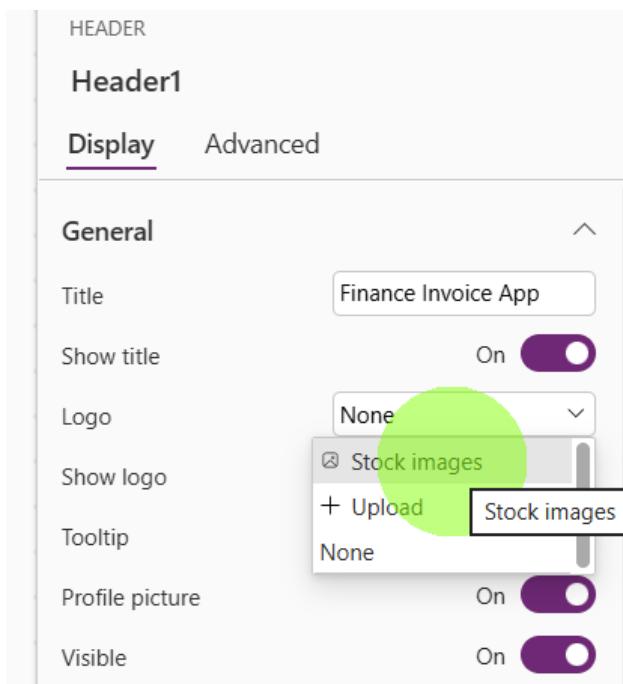
Delete Screen1.



2. Select Header and change Title to Finance Invoice App.

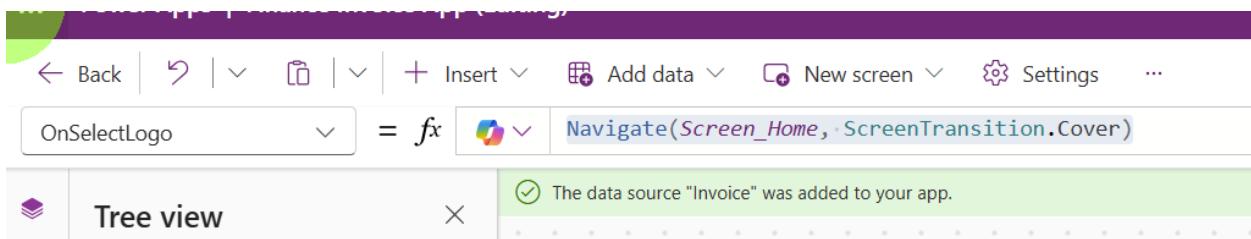


Click Logo and select any stock image that you like.

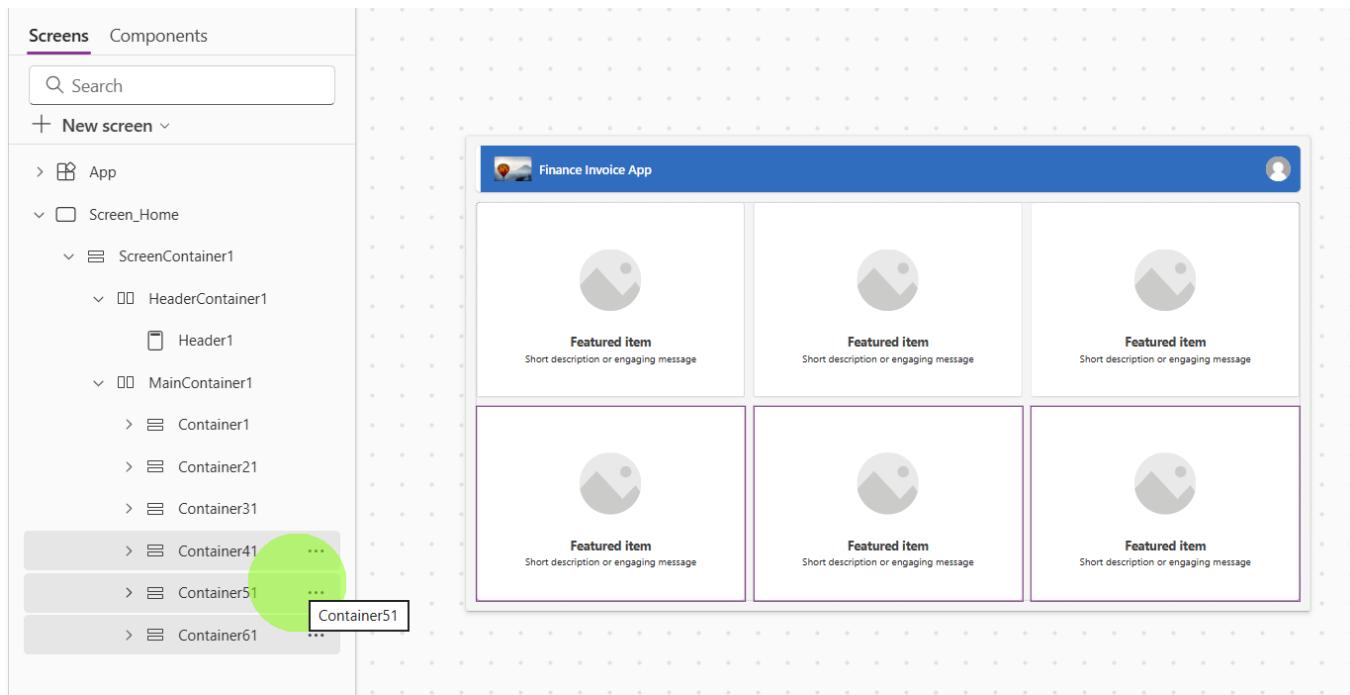


3. Change the OnSelectLogo property of Header to below.

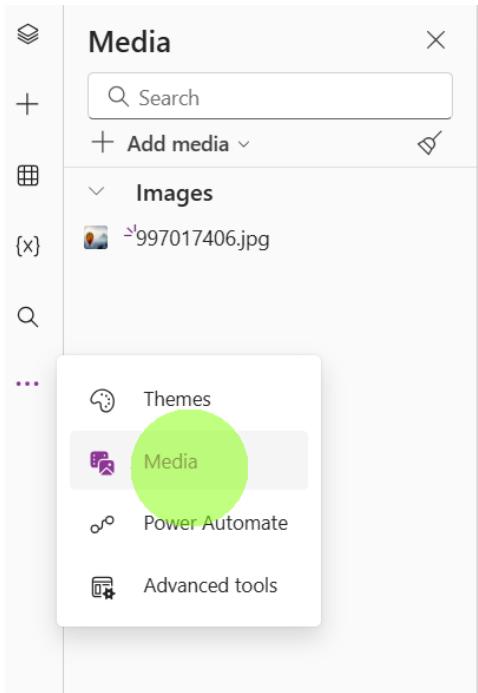
`Navigate(Screen_Home, ScreenTransition.Cover)`



4. Expand Main Container from Tree View. Select and delete the bottom 3 containers.



5. Upload Icons to the app by clicking Media > + Add media



Upload these 3 icons from Upload Icons folder.

Name	Status	Date modified	Type	Size
Dashboard.svg	✓	18/1/2026 5:47 pm	Microsoft Edge HT...	2 KB
List.svg	✓	18/1/2026 5:48 pm	Microsoft Edge HT...	1 KB
Upload.svg	✓	18/1/2026 5:45 pm	Microsoft Edge HT...	1 KB

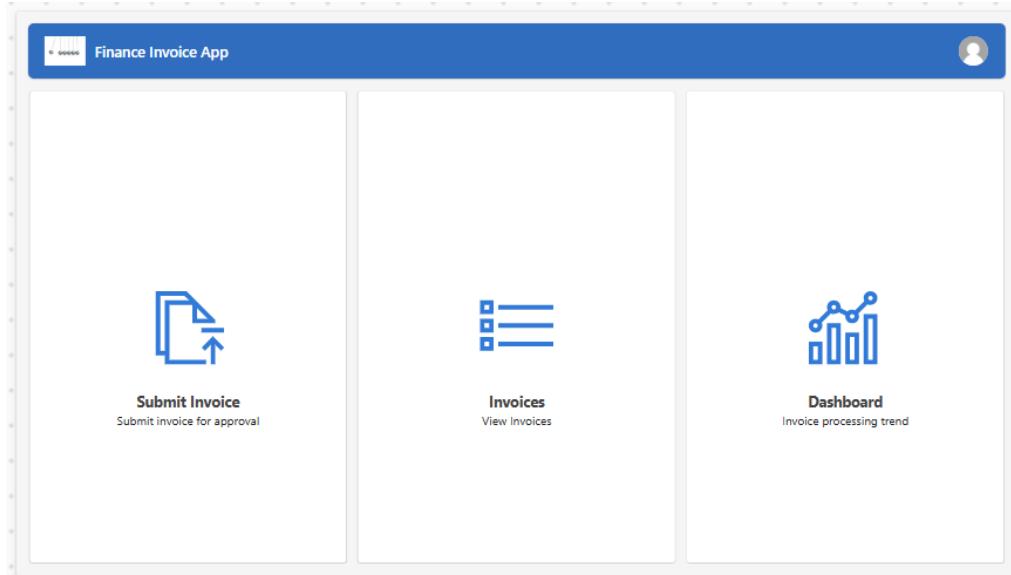
6. Select each icon on Home screen and update Image properties accordingly.

	Image Property	Border Radius
<b>First Icon</b>	Upload	0
<b>Second Icon</b>	List	0
<b>Third Icon</b>	Dashboard	0

7. Select the Text controls under each image and update accordingly.

	Title	Description
<b>First Icon</b>	Submit Invoice	Submit invoice for approval
<b>Second Icon</b>	Invoices	View Invoices
<b>Third Icon</b>	Dashboard	Invoice processing trend

Your home screen should look like below.

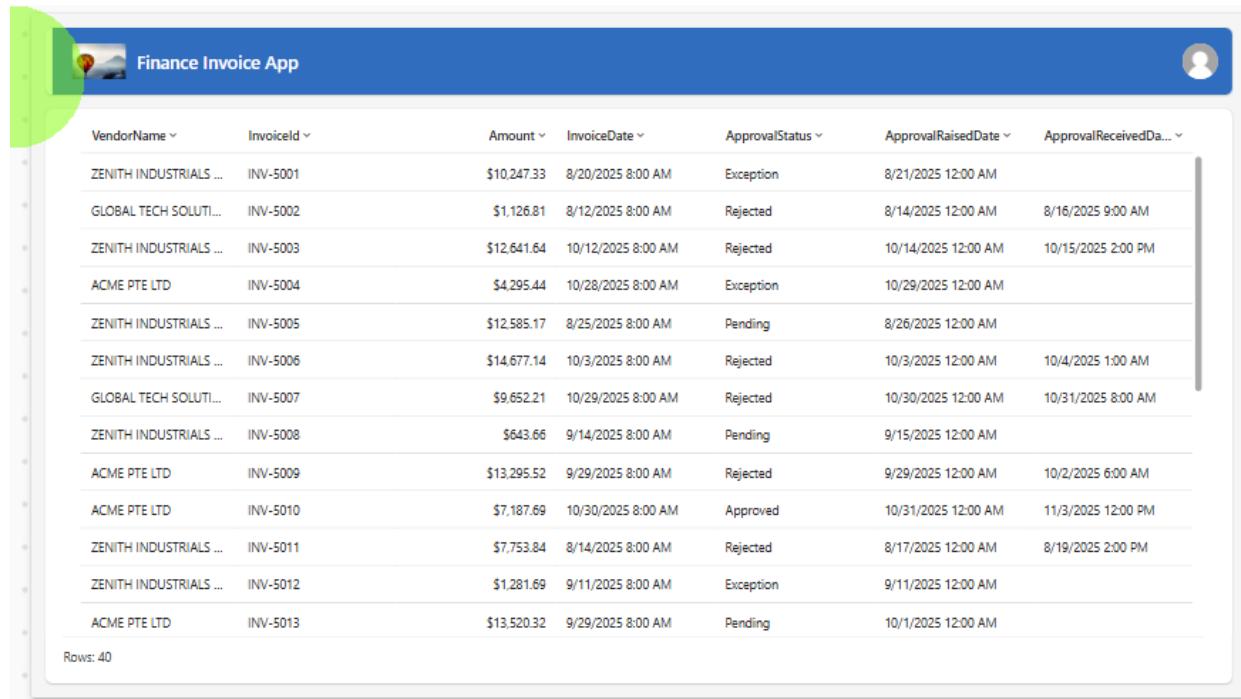


**Important! Remember to Save the App.**

----- End of Task -----

# Exercise 3: Connecting to Data and Displaying it

In this exercise, we will be displaying data in the app from the Invoices table that we created in exercise 1. We will then customize the look and feel of the data being displayed to create a nice UI format.



The screenshot shows a Microsoft Power App interface titled "Finance Invoice App". The main area contains a table control displaying data from the Invoices table. The columns are: VendorName, InvoiceId, Amount, InvoiceDate, ApprovalStatus, ApprovalRaisedDate, and ApprovalReceivedDa... (truncated). The data includes various invoices from vendors like ZENITH INDUSTRIALS, GLOBAL TECH SOLUTIONS, ACME PTE LTD, etc., with details such as invoice amounts ranging from \$1,126.81 to \$10,247.33 and approval statuses like Pending, Approved, Rejected, or Exception.

VendorName	InvoiceId	Amount	InvoiceDate	ApprovalStatus	ApprovalRaisedDate	ApprovalReceivedDa...
ZENITH INDUSTRIALS ...	INV-5001	\$10,247.33	8/20/2025 8:00 AM	Exception	8/21/2025 12:00 AM	
GLOBAL TECH SOLUTIONS ...	INV-5002	\$1,126.81	8/12/2025 8:00 AM	Rejected	8/14/2025 12:00 AM	8/16/2025 9:00 AM
ZENITH INDUSTRIALS ...	INV-5003	\$12,641.64	10/12/2025 8:00 AM	Rejected	10/14/2025 12:00 AM	10/15/2025 2:00 PM
ACME PTE LTD	INV-5004	\$4,295.44	10/28/2025 8:00 AM	Exception	10/29/2025 12:00 AM	
ZENITH INDUSTRIALS ...	INV-5005	\$12,585.17	8/25/2025 8:00 AM	Pending	8/26/2025 12:00 AM	
ZENITH INDUSTRIALS ...	INV-5006	\$14,677.14	10/3/2025 8:00 AM	Rejected	10/3/2025 12:00 AM	10/4/2025 1:00 AM
GLOBAL TECH SOLUTIONS ...	INV-5007	\$9,652.21	10/29/2025 8:00 AM	Rejected	10/30/2025 12:00 AM	10/31/2025 8:00 AM
ZENITH INDUSTRIALS ...	INV-5008	\$643.66	9/14/2025 8:00 AM	Pending	9/15/2025 12:00 AM	
ACME PTE LTD	INV-5009	\$13,295.52	9/29/2025 8:00 AM	Rejected	9/29/2025 12:00 AM	10/2/2025 6:00 AM
ACME PTE LTD	INV-5010	\$7,187.69	10/30/2025 8:00 AM	Approved	10/31/2025 12:00 AM	11/3/2025 12:00 PM
ZENITH INDUSTRIALS ...	INV-5011	\$7,753.84	8/14/2025 8:00 AM	Rejected	8/17/2025 12:00 AM	8/19/2025 2:00 PM
ZENITH INDUSTRIALS ...	INV-5012	\$1,281.69	9/11/2025 8:00 AM	Exception	9/11/2025 12:00 AM	
ACME PTE LTD	INV-5013	\$13,520.32	9/29/2025 8:00 AM	Pending	10/1/2025 12:00 AM	

## Objectives

The exercise will achieve the following objectives.

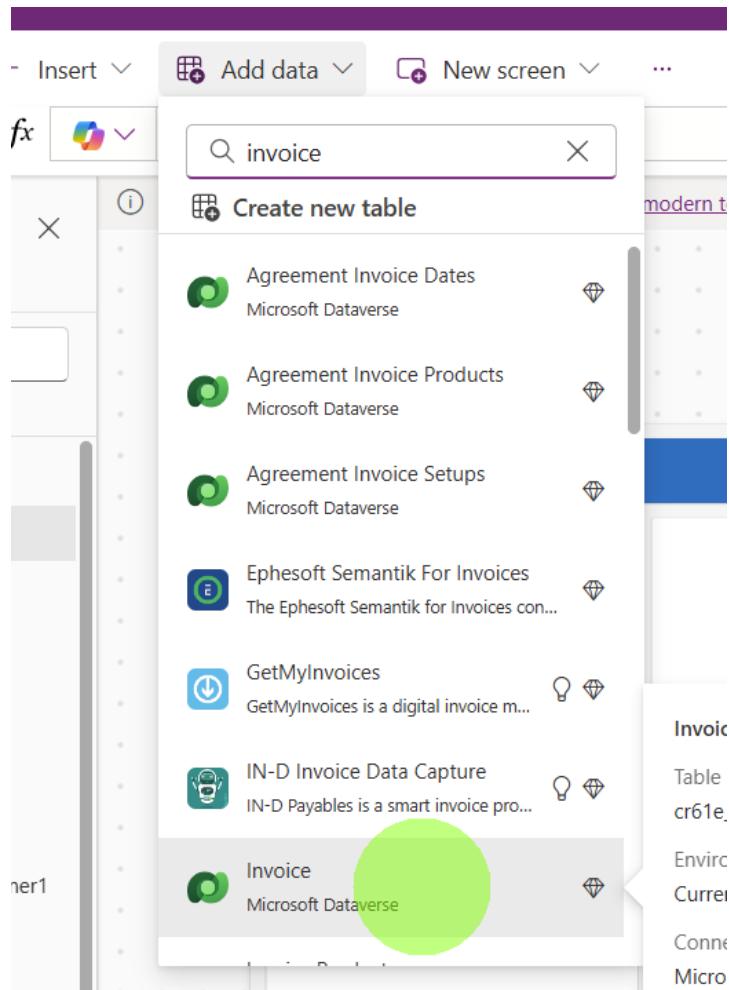
- Learn how to connect to a Dataverse table
- Display data in a Table control
- Filter data displayed in Table control

## Estimated time to complete this lab

20 mins

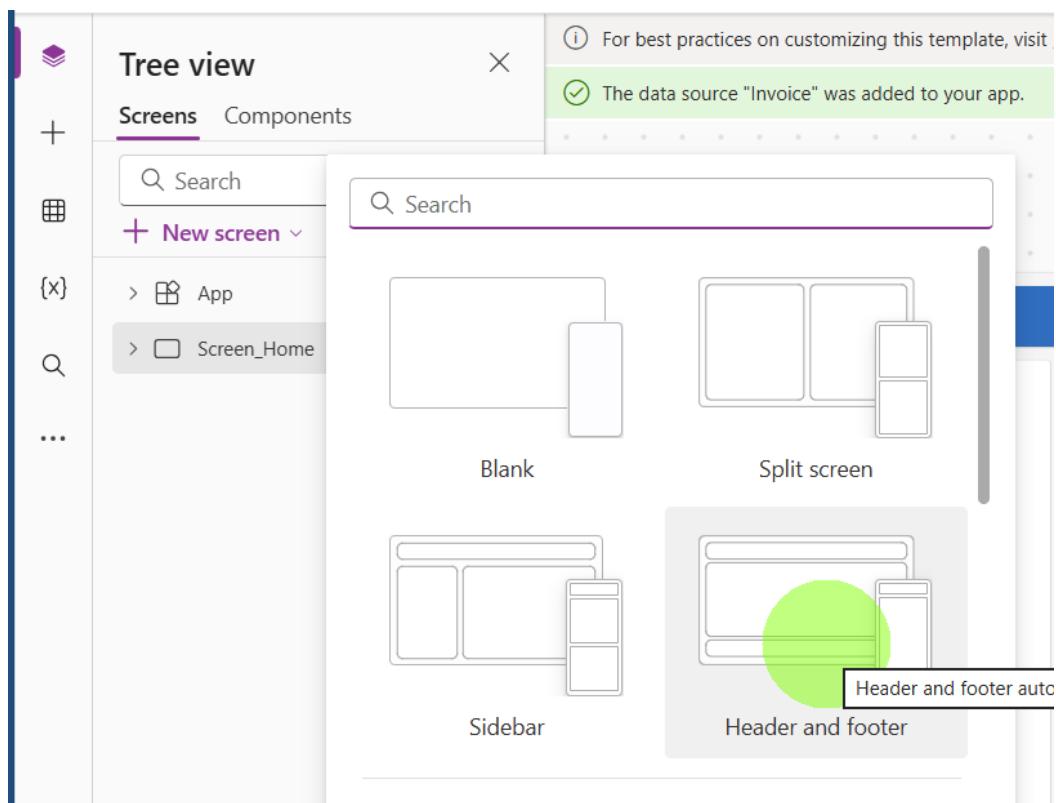
## Task 1: Connecting to the Dataverse Table

1. Click on Add data and locate the Invoice table and click.

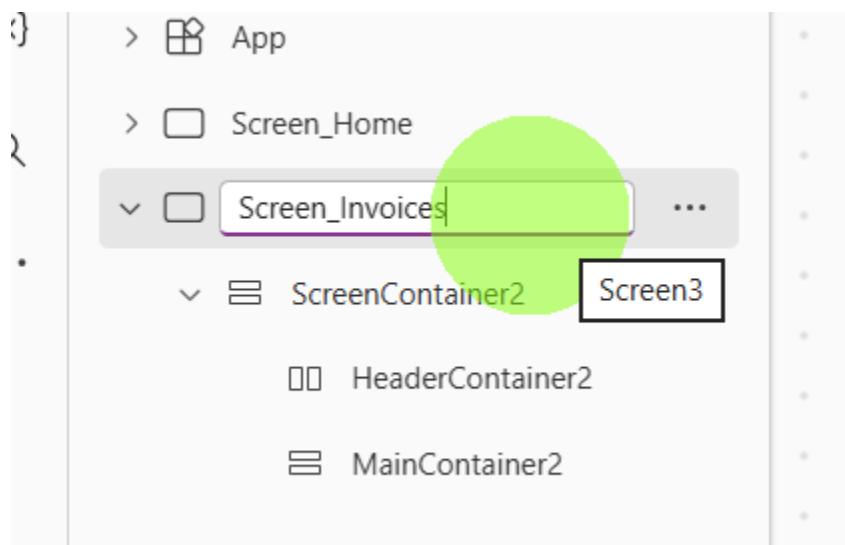


## Task 2: Displaying the Data in a Table Control

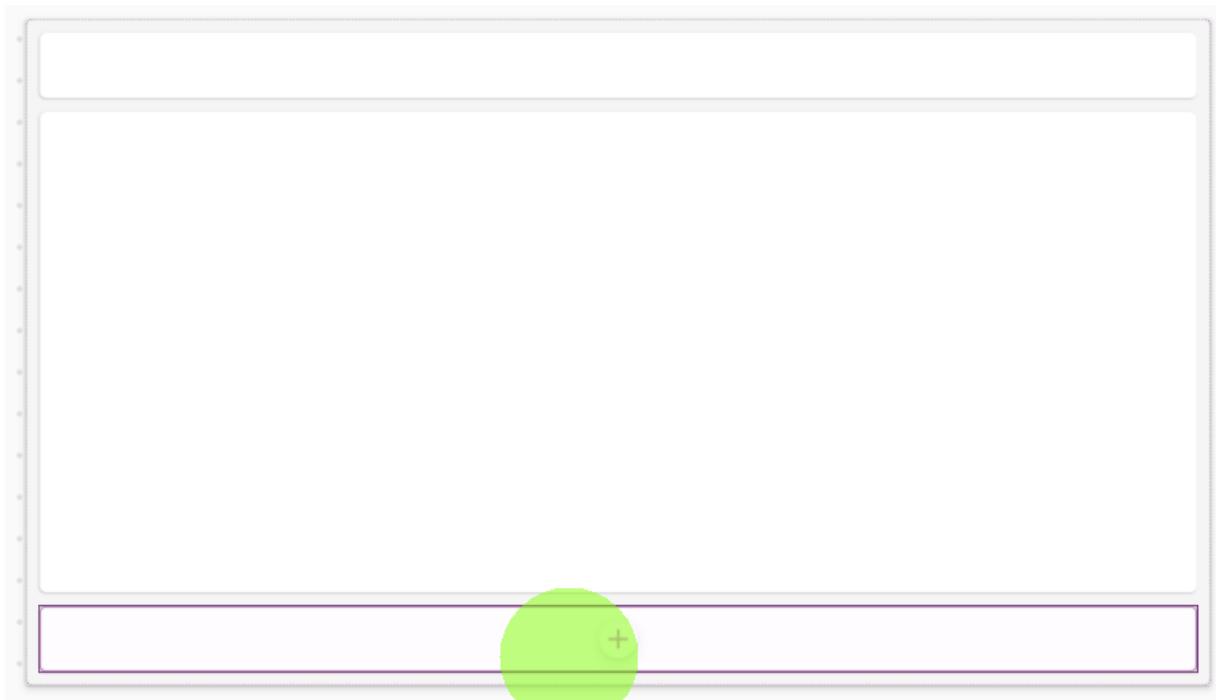
1. Click New screen > Header and footer to add a new screen.



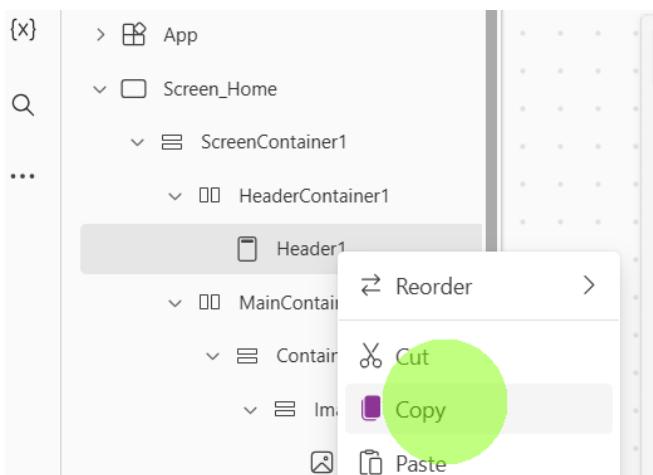
2. Rename the screen to Screen\_Invoices.

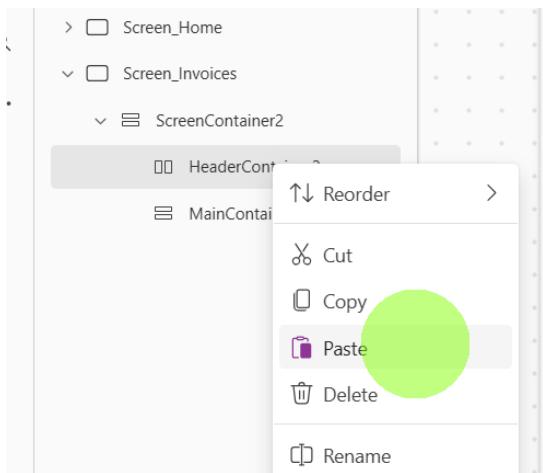


3. Select the bottom container and delete.

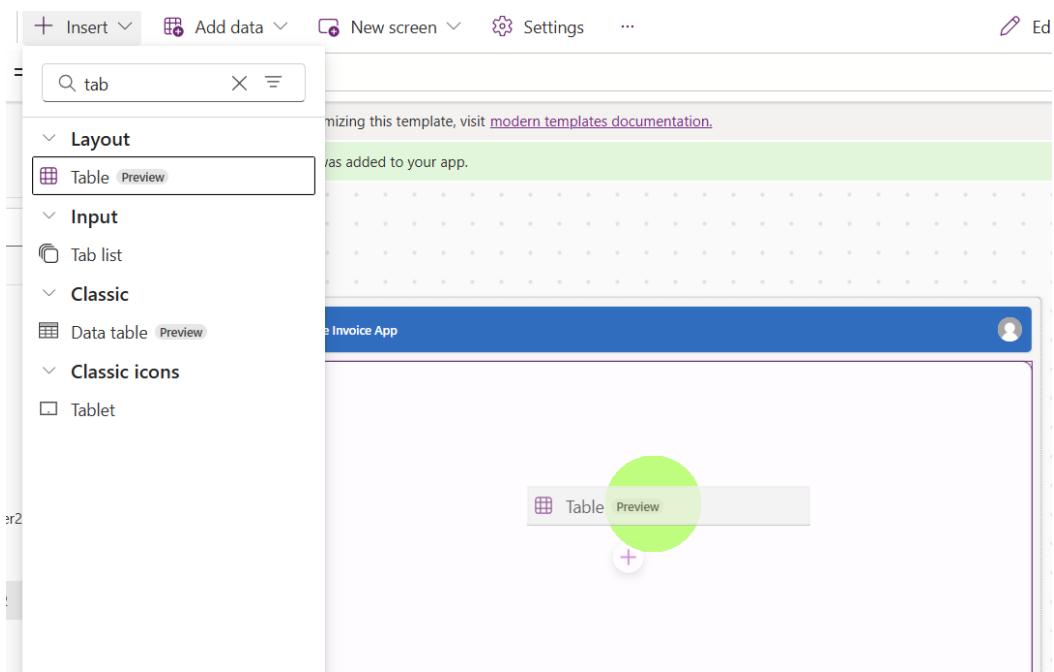


4. Copy Header1 from Screen\_Home and paste in Screen\_Invoices's Header Container.

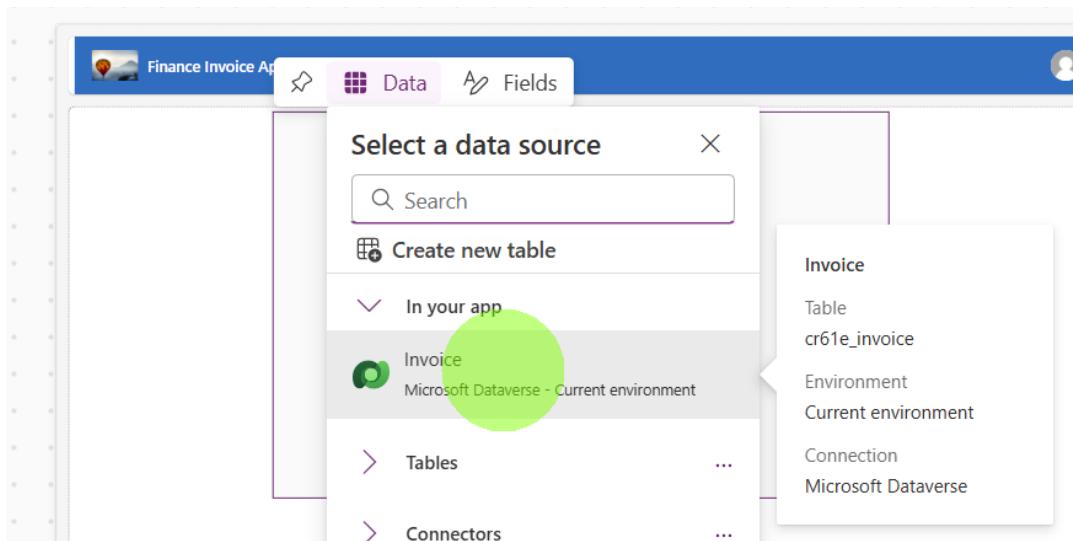




8. Click Insert > Table into the MainContainer.



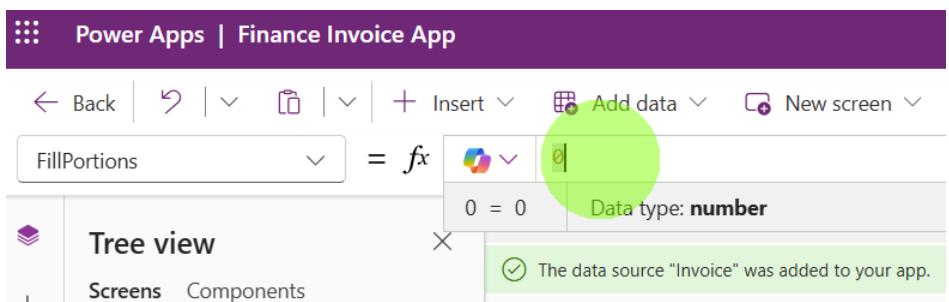
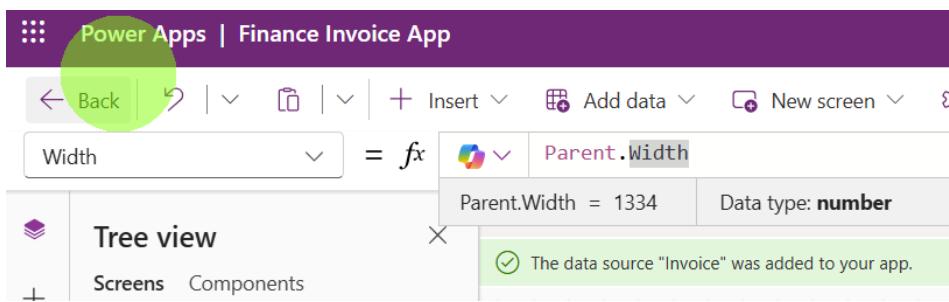
Select Invoice table as data source for the table.



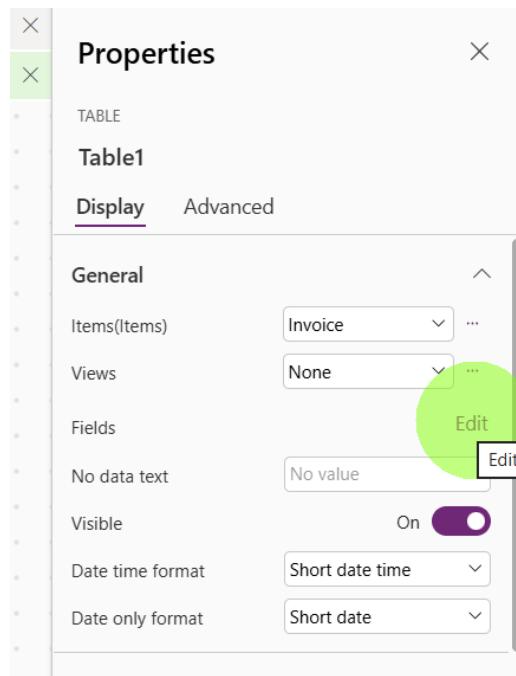
9. Change below properties of table.

Width: **Parent.Width**

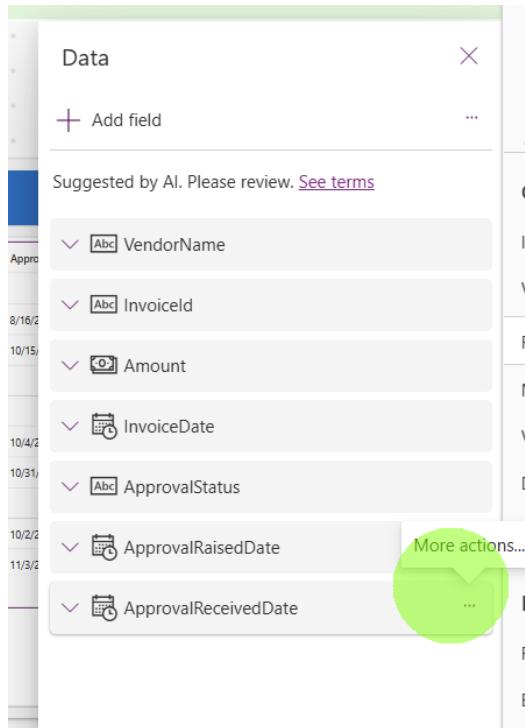
FillPortions: 1



10. Change the fields displayed by clicking on Edit beside Fields.



Edit the fields list to only keep these fields.



Data

Add field

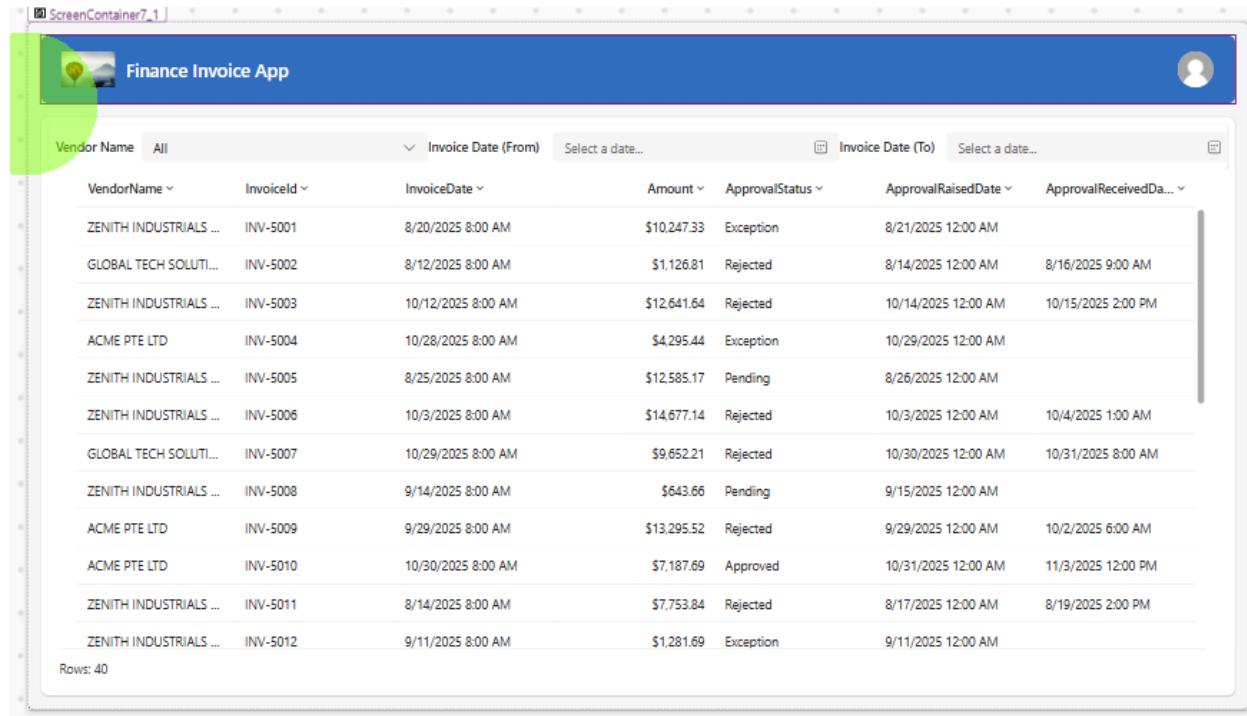
Suggested by AI. Please review. [See terms](#)

Field	Type	Value
VendorName	Text	Abc
InvoiceId	Text	Abc
Amount	Number	8/16/2
InvoiceDate	Date	10/15/2
ApprovalStatus	Text	10/31/2
ApprovalRaisedDate	Date	10/2/2
ApprovalReceivedDate	Date	11/3/2

----- End of Task -----

# Exercise 4: Filtering the Table

In this exercise, you will learn how to perform filtering and on the table to quickly find information when needed.



Vendor Name	All	Invoice Date (From)	Select a date...	Invoice Date (To)	Select a date...	
VendorName	InvoiceId	InvoiceDate	Amount	ApprovalStatus	ApprovalRaisedDate	ApprovalReceivedDa...
ZENITH INDUSTRIALS ...	INV-5001	8/20/2025 8:00 AM	\$10,247.33	Exception	8/21/2025 12:00 AM	
GLOBAL TECH SOLUTI...	INV-5002	8/12/2025 8:00 AM	\$1,126.81	Rejected	8/14/2025 12:00 AM	8/16/2025 9:00 AM
ZENITH INDUSTRIALS ...	INV-5003	10/12/2025 8:00 AM	\$12,641.64	Rejected	10/14/2025 12:00 AM	10/15/2025 2:00 PM
ACME PTE LTD	INV-5004	10/28/2025 8:00 AM	\$4,295.44	Exception	10/29/2025 12:00 AM	
ZENITH INDUSTRIALS ...	INV-5005	8/25/2025 8:00 AM	\$12,585.17	Pending	8/26/2025 12:00 AM	
ZENITH INDUSTRIALS ...	INV-5006	10/3/2025 8:00 AM	\$14,677.14	Rejected	10/3/2025 12:00 AM	10/4/2025 1:00 AM
GLOBAL TECH SOLUTI...	INV-5007	10/29/2025 8:00 AM	\$9,652.21	Rejected	10/30/2025 12:00 AM	10/31/2025 8:00 AM
ZENITH INDUSTRIALS ...	INV-5008	9/14/2025 8:00 AM	\$643.66	Pending	9/15/2025 12:00 AM	
ACME PTE LTD	INV-5009	9/29/2025 8:00 AM	\$13,295.52	Rejected	9/29/2025 12:00 AM	10/2/2025 6:00 AM
ACME PTE LTD	INV-5010	10/30/2025 8:00 AM	\$7,187.69	Approved	10/31/2025 12:00 AM	11/3/2025 12:00 PM
ZENITH INDUSTRIALS ...	INV-5011	8/14/2025 8:00 AM	\$7,753.84	Rejected	8/17/2025 12:00 AM	8/19/2025 2:00 PM
ZENITH INDUSTRIALS ...	INV-5012	9/11/2025 8:00 AM	\$1,281.69	Exception	9/11/2025 12:00 AM	

## Objectives

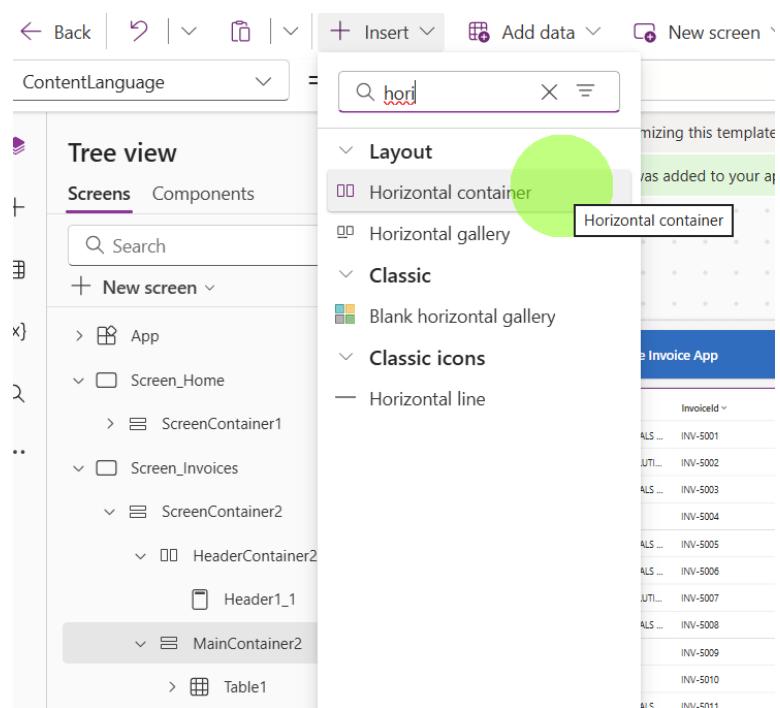
- Filter a table by Vendor Name
- Filter a table by Date Range

## Estimated time to complete this lab

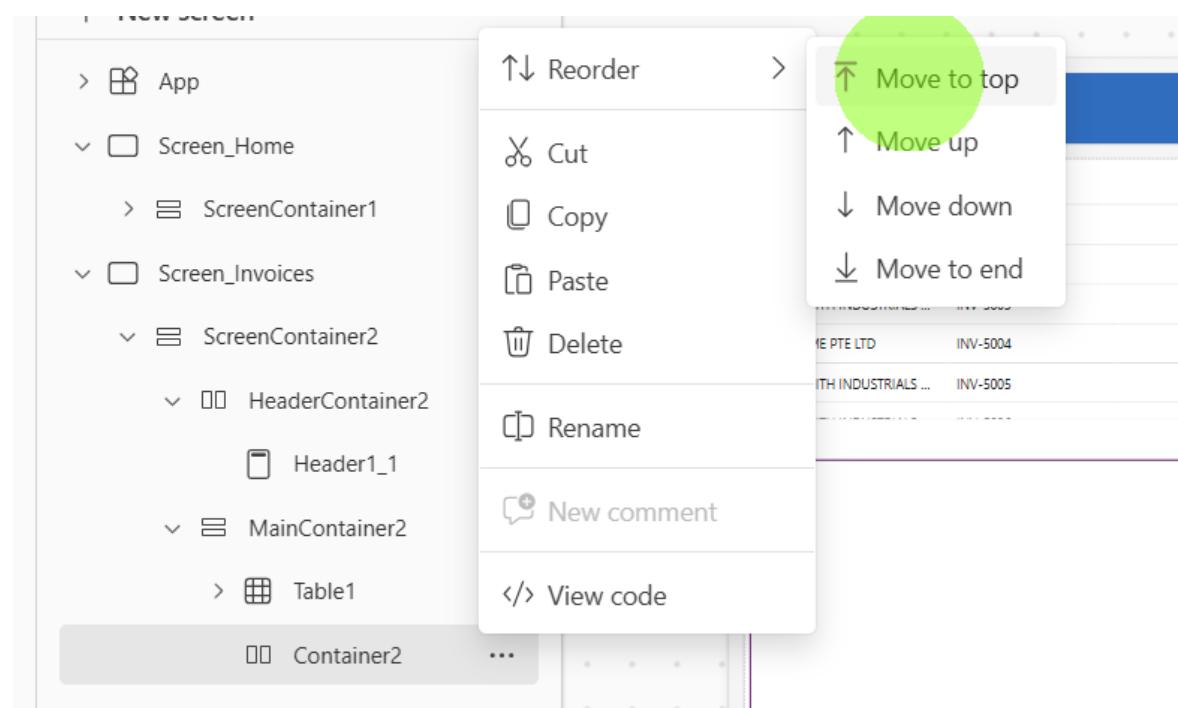
20 mins

## Task 1: Creating Filter Controls

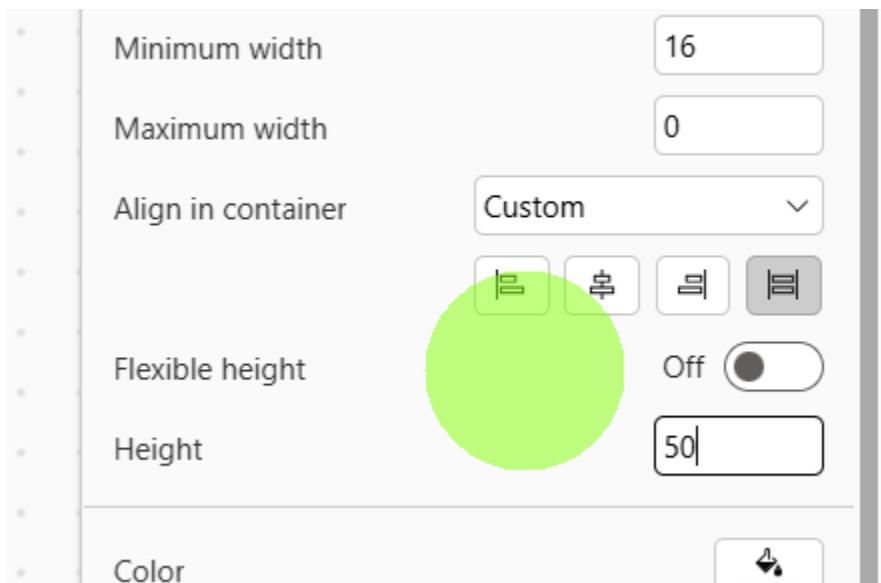
1. Insert a new Horizontal Container to Main Container.



Reorder the new Container to top.

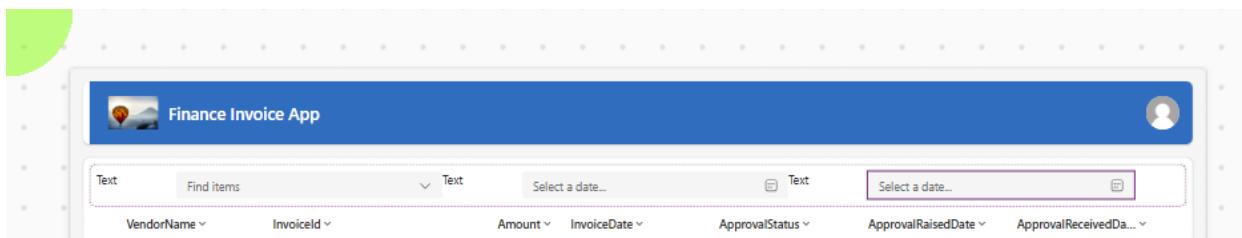


Switch Off Flexible height and set Height to 50.



2. Add 6 controls below to the container.

- Text Control
- ComboBox Control
- Text Control
- Date Picker Control
- Text Control
- Date Picker Control

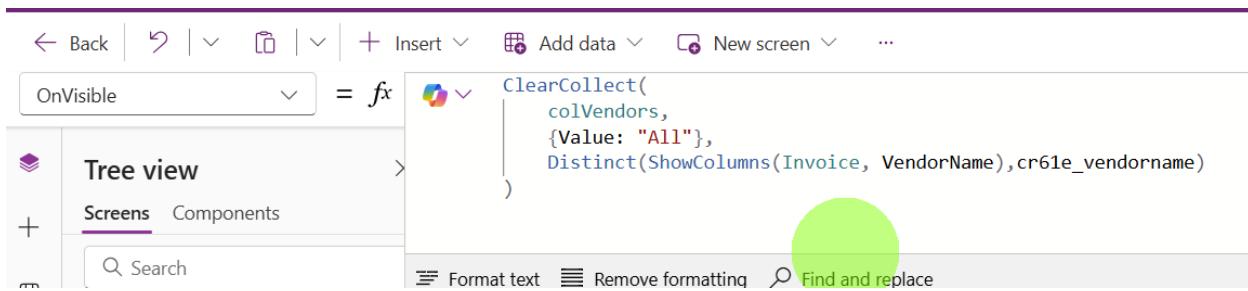


3. Change properties of those controls to below.

	<b>Text</b>	<b>ControlName</b>	<b>Width</b>
<b>Text Control</b>	Vendor Name	<i>No change</i>	<i>No change</i>
<b>ComboBox Control</b>	-	<i>Combo_VendorName</i>	300
<b>Text Control</b>	Invoice Date (From)	<i>No change</i>	140
<b>Date Picker Control</b>	-	<i>Date_From</i>	300
<b>Text Control</b>	Invoice Date (To)	<i>No change</i>	140
<b>Date Picker Control</b>	-	<i>Date_To</i>	300

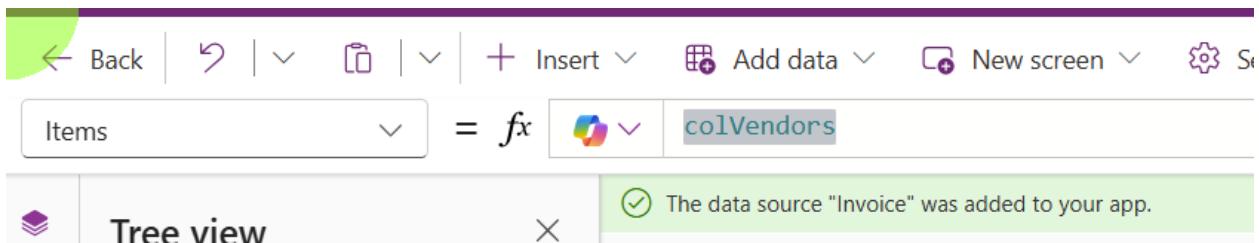
4. Insert this formula into OnVisible property of Screen\_Invoices.

```
ClearCollect(
    colVendors,
    {Value: "All"},
    Distinct(ShowColumns(Invoice, VendorName), cr61e_vendorname)
)
```



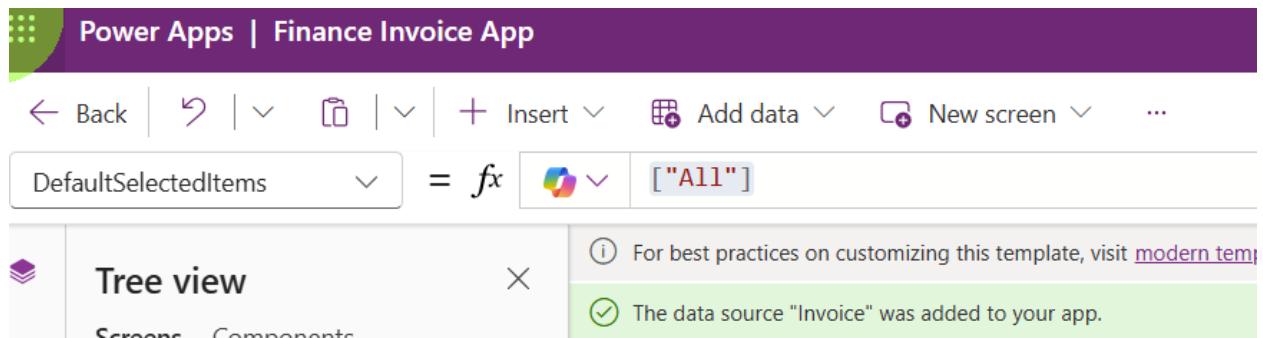
5. Change Items property of Combo\_VendorName to below.

colVendors



11. Change DefaultSelectedItems property of Combo\_VendorName to below.

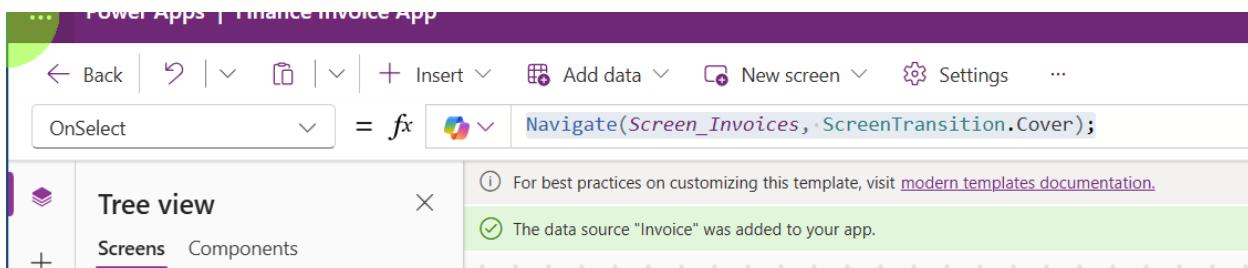
["All"]



## Task 2: Applying Filter

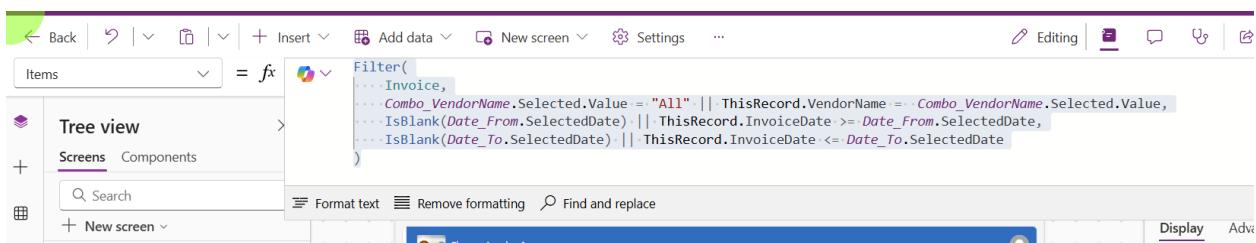
- Click Screen\_Home. And set OnSelect property of Invoices Icon to below.

```
Navigate(Screen_Invoices, ScreenTransition.Cover);
```



- Set Items property of Table to below.

```
Filter(
    Invoice,
    Combo_VendorName.Selected.Value = "All" || ThisRecord.VendorName
= Combo_VendorName.Selected.Value,
    IsBlank(Date_From.SelectedDate) || ThisRecord.InvoiceDate >=
Date_From.SelectedDate,
    IsBlank(Date_To.SelectedDate) || ThisRecord.InvoiceDate <=
Date_To.SelectedDate
)
```



You should see the vendor list populated and be able to filter the table by vendor name, date range.

Sorting is automatically done by the Table control.

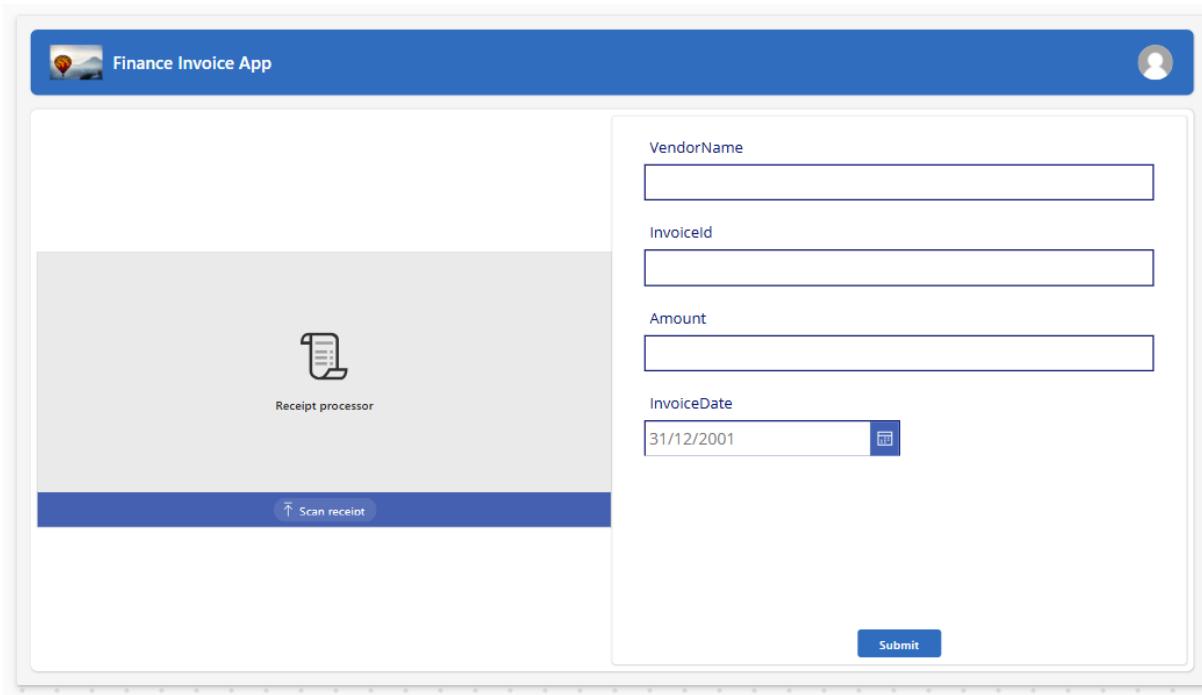
---

**End of Task**

---

# Exercise 5: Allow Users to Submit New Invoices

Users need to be able to submit new invoices to be processed using Receipt Processor. This exercise will get users to create a new screen, allow users upload invoices and automatically extract information.



## Objectives

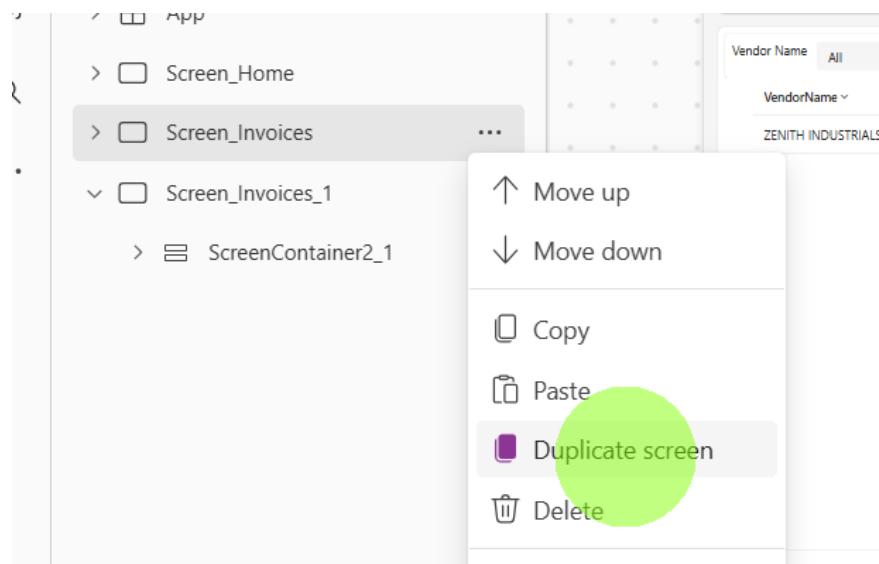
- User Receipt Processor to extract values from PDF
- Add in form that allows users to create a new request

## Estimated time to complete this lab

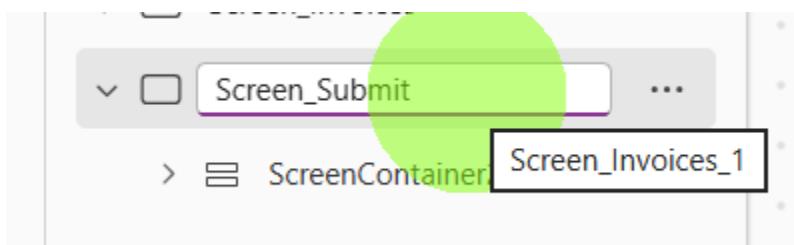
30 mins

## Task 1: Duplicate the existing Screen

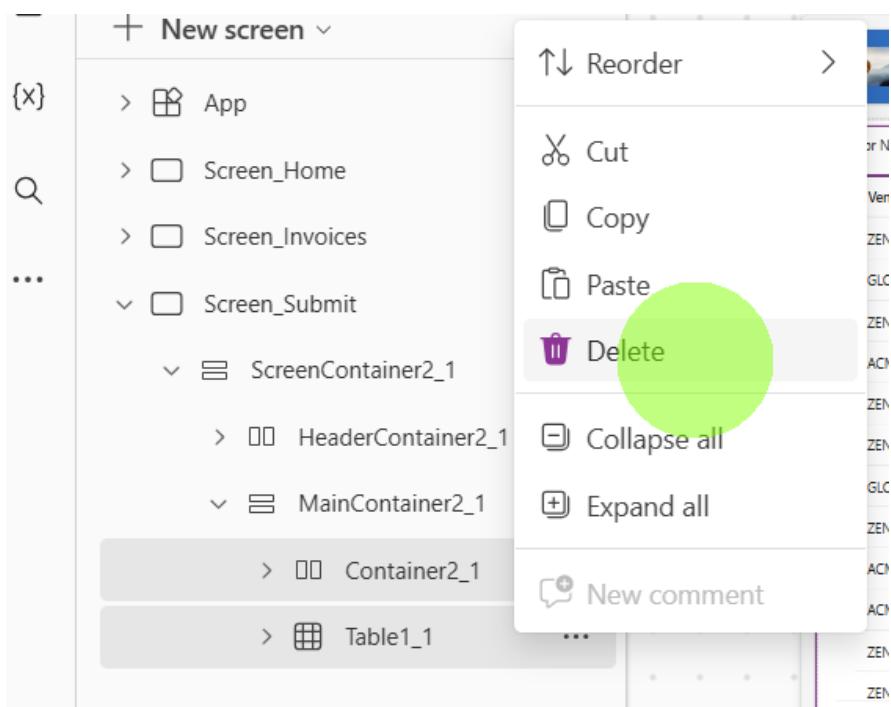
1. Duplicate Screen\_Invoices by following selecting the screen, right clicking it, and then selecting the 'Duplicate Screen' option.



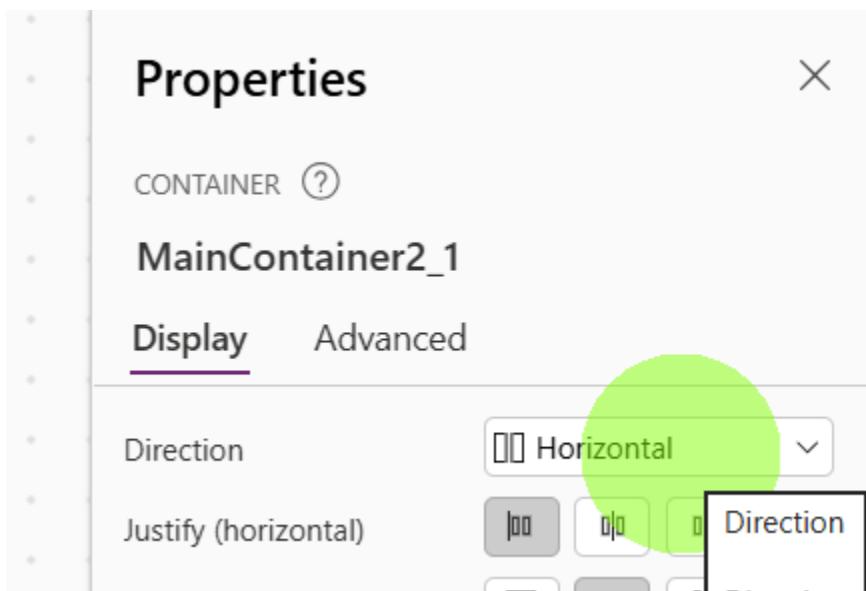
Rename the new screen to Screen\_Submit.



2. Expand the Main Container and delete the controls inside.

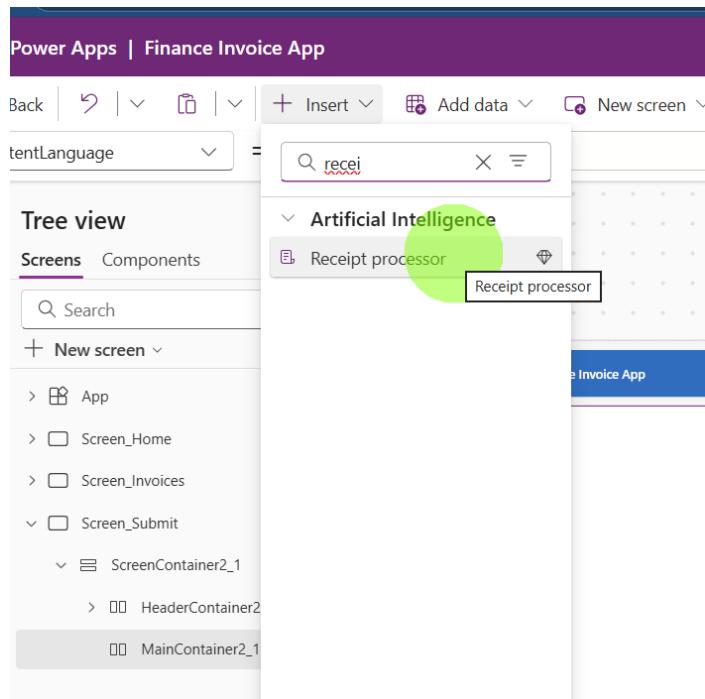


3. Change Main Container's Direction to Horizontal.



## Task 2: Adding Controls to the Screen

1. Select Main Container and add Receipt Processor control.

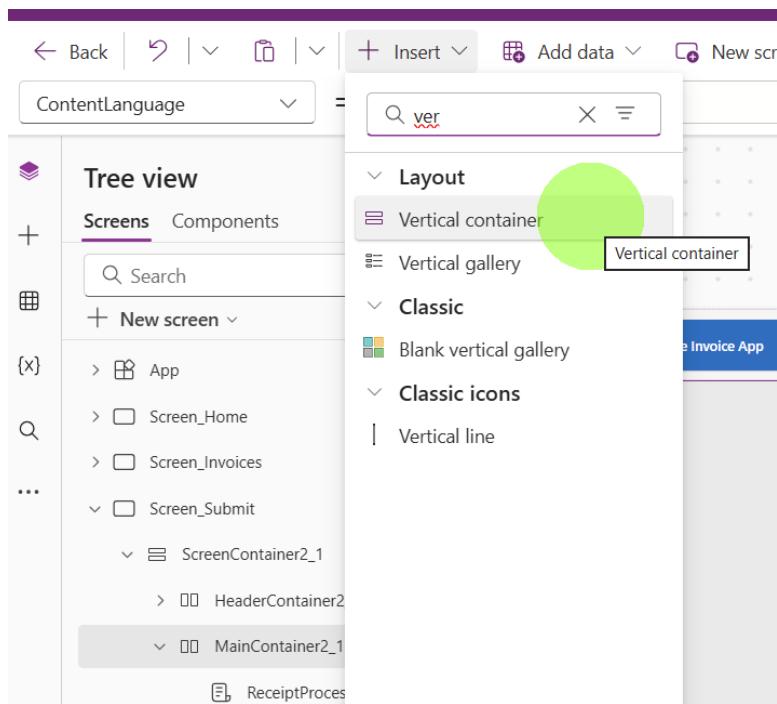


Change below properties of the Receipt Processor.

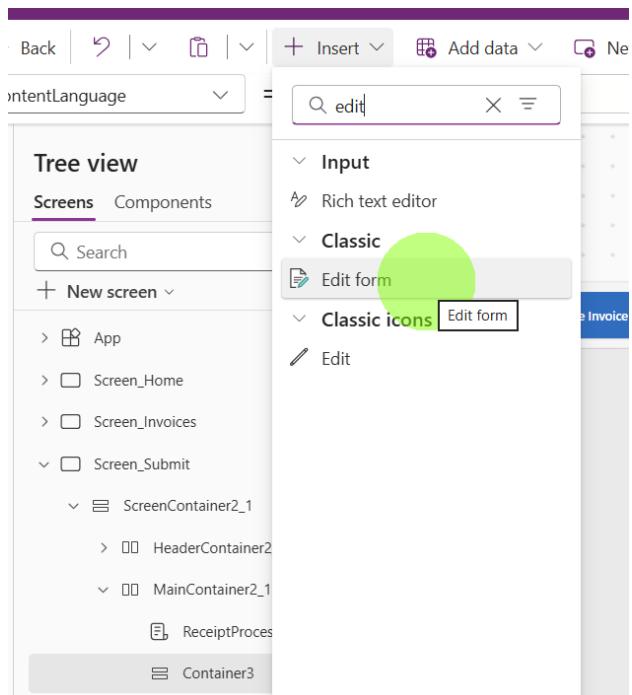
Height: **Parent**.Height

FillPortions: 1

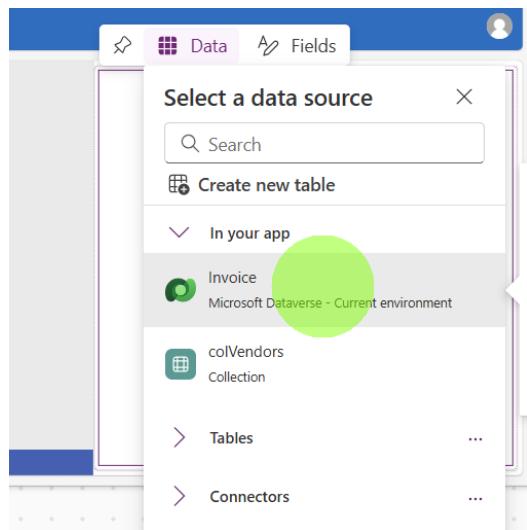
2. Add a new Vertical Container into the Main Container.



3. Add a new Form into the Vertical Container.



4. Point the Data Source for the Form to Invoice.



5. Change the number of Columns to 1

EDIT ⓘ

### Form1

Display Advanced

Data source: Invoice

Views: None

Fields: [Edit fields](#)

Snap to columns: On

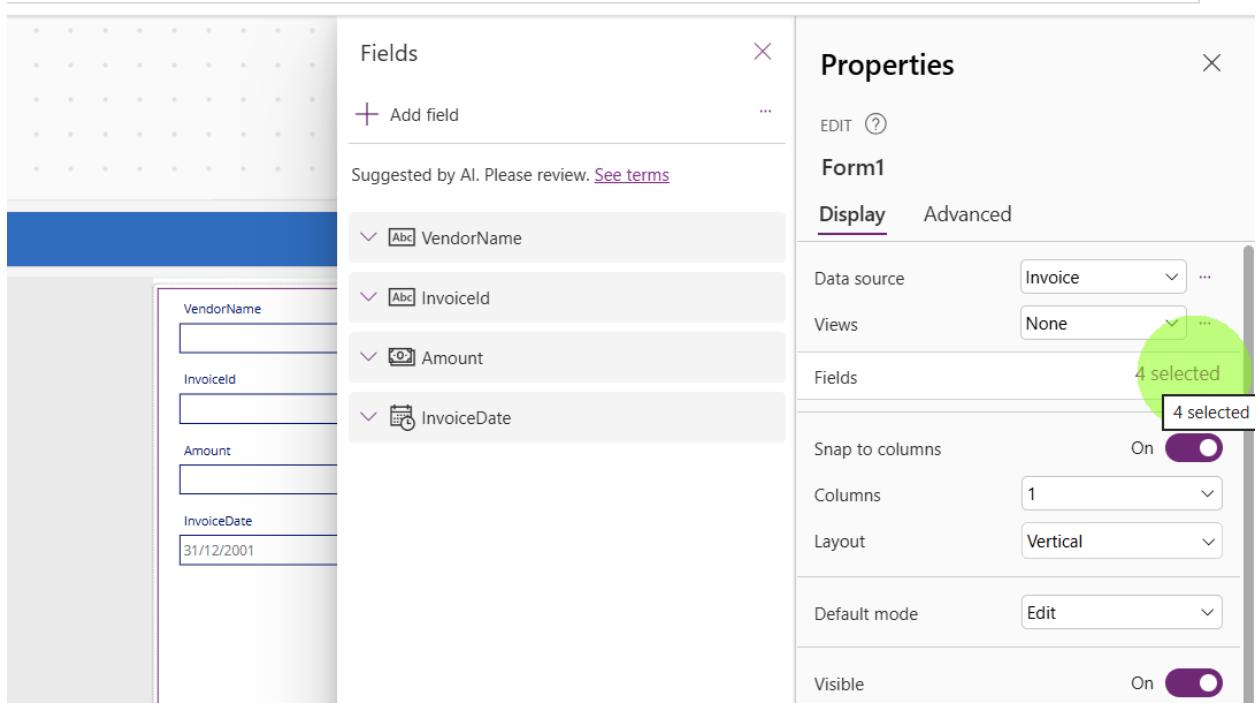
Columns: 3 (1 highlighted with a green circle)

Layout: 1, 2, 3, 4

Default mode: 3, 4

Visible: 6, 12

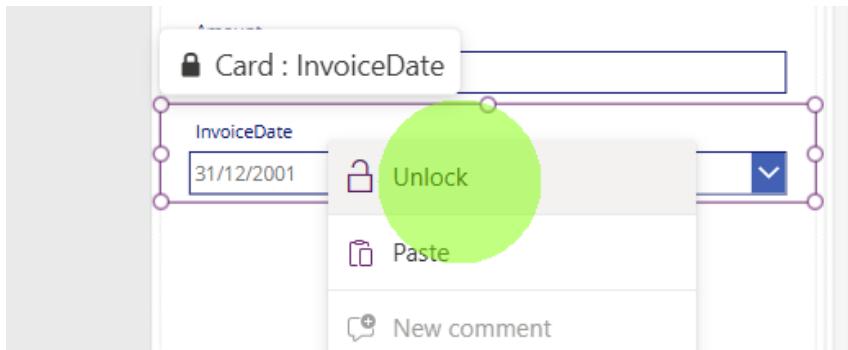
6. Select Fields to edit the fields displayed and only keep these 4 fields below.

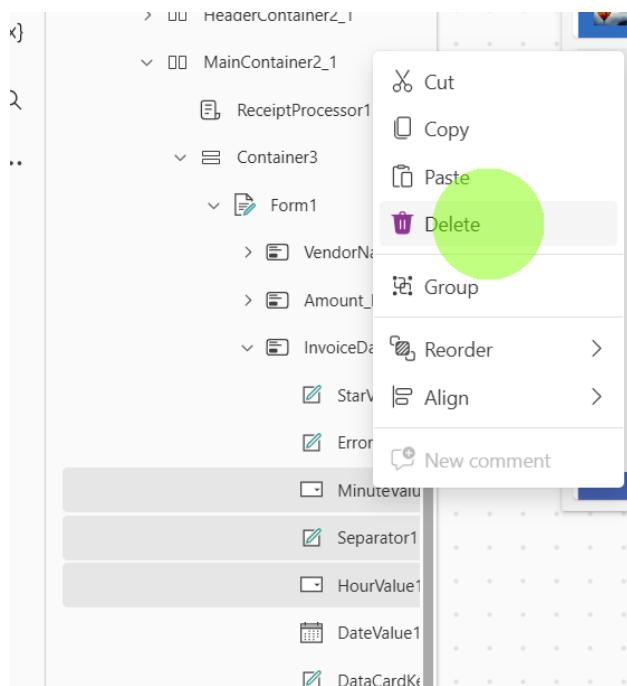


The screenshot shows the Microsoft Power Platform interface. On the left is a preview of a card with four fields: VendorName, InvoiceId, Amount, and InvoiceDate. To the right is the 'Fields' panel, which lists these four fields with dropdown arrows indicating they are expandable. Below the fields is a note: 'Suggested by AI. Please review. [See terms](#)'. On the far right is the 'Properties' panel for 'Form1'. The 'Display' tab is selected. Key settings shown include:

- Data source: Invoice
- Views: None
- Fields: 4 selected
- Snap to columns: On
- Columns: 1
- Layout: Vertical
- Default mode: Edit
- Visible: On

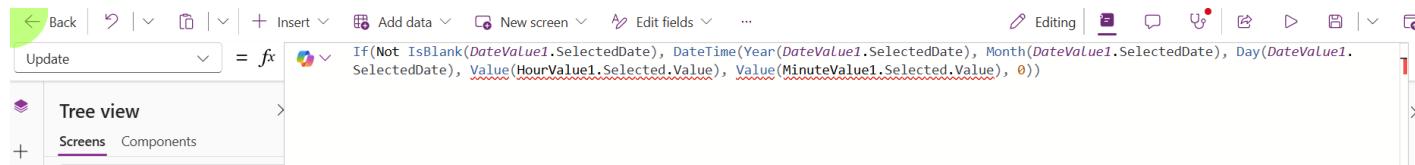
7. Unlock Invoice Date Field and delete Hours, Minutes and Separator controls.





There will be errors generated. To fix the error, replace the Error Values with Zeros.

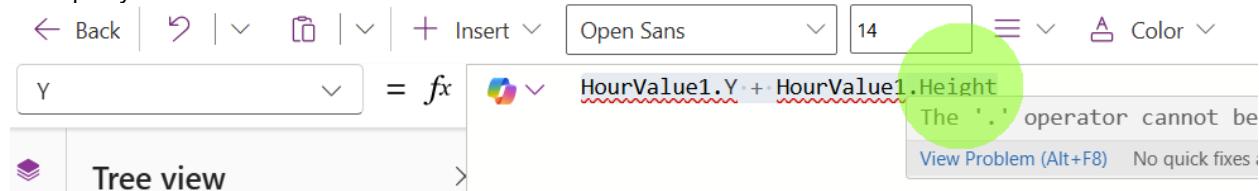
## Update Property



`Value(HourValue1.Selected.Value) → 0`

`Value(MinuteValue1.Selected.Value) → 0`

## Y Property



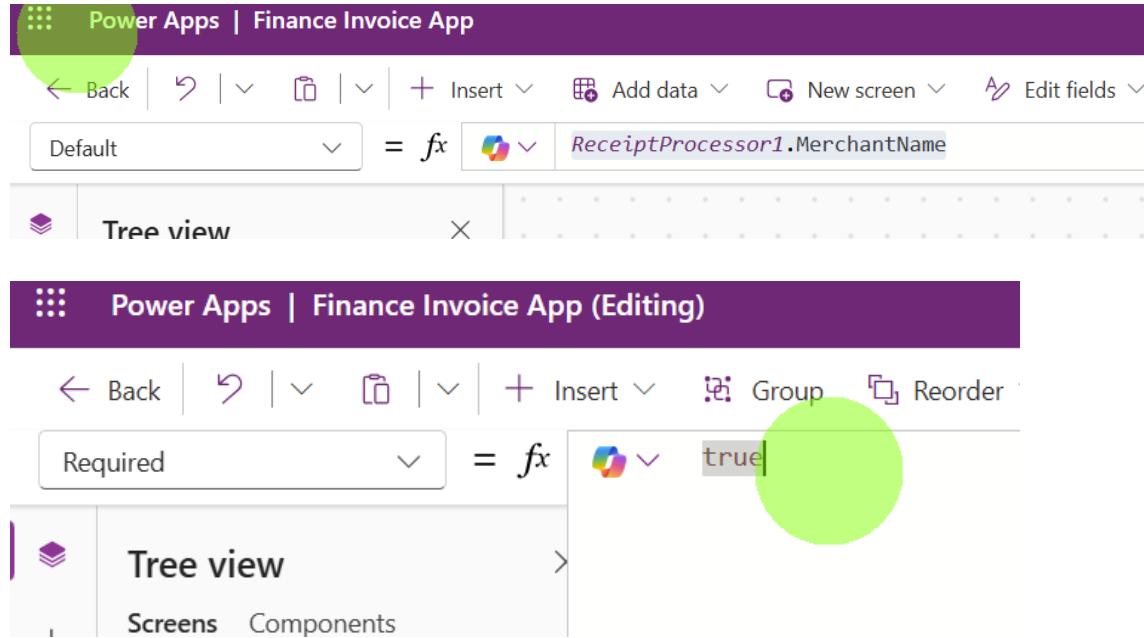
`HourValue1.Y + HourValue1.Height → 0`

----- End of Task -----

## Task 3: Extracting Values from Receipt Processor

1. Unlock VendorName Card and set Default property to below and Required to true.

*ReceiptProcessor1*.MerchantName



The screenshot shows the Power Apps studio interface. The top part is a screenshot of a browser window titled "Power Apps | Finance Invoice App". It shows a formula editor with the expression `= fx [ReceiptProcessor1.MerchantName]`. The bottom part is a screenshot of the Power Apps studio interface titled "Power Apps | Finance Invoice App (Editing)". It shows the "Required" property being set to "true" in a formula editor. A green circle highlights the word "true" in the formula bar.

2. Unlock Invoiceld Card and set Default property to below and Required to true.

```
With(
{
    DetectedText: First(
        Filter(ReceiptProcessor1.DetectedText, "Invoice Number" in
Value)
        ).Value
},
Trim(
    Substitute(DetectedText, "Invoice Number: ", ""))
)
```

[Back](#) | [Edit](#) | [Add data](#) | [New screen](#) | [Edit fields](#) | ...

Default = [fx](#)

```
With(
{
    DetectedText: First(
        Filter(ReceiptProcessor1.DetectedText, "Invoice Number" in Value)
    ).Value
},
Trim(
    Substitute(DetectedText, "Invoice Number: ", "")
))
```

Tree view

Screens Components

Search

+ New screen

{x} Screen\_Invoices

## Power Apps | Finance Invoice App (Editing)

[Back](#) | [Edit](#) | [Add data](#) | [New screen](#) | [Edit fields](#) | [Group](#) | [Reorder](#)

Required = [fx](#)

true

Tree view

Screens Components

3. Unlock Amount Card and set Default property to below and Required to true.

*ReceiptProcessor1*.Total

[Back](#) | [Edit](#) | [Add data](#) | [New screen](#) | [Edit fields](#)

Default = [fx](#)

*ReceiptProcessor1*.Total

Tree view

## Power Apps | Finance Invoice App (Editing)

[Back](#) | [Edit](#) | [Add data](#) | [New screen](#) | [Edit fields](#) | [Group](#) | [Reorder](#)

Required = [fx](#)

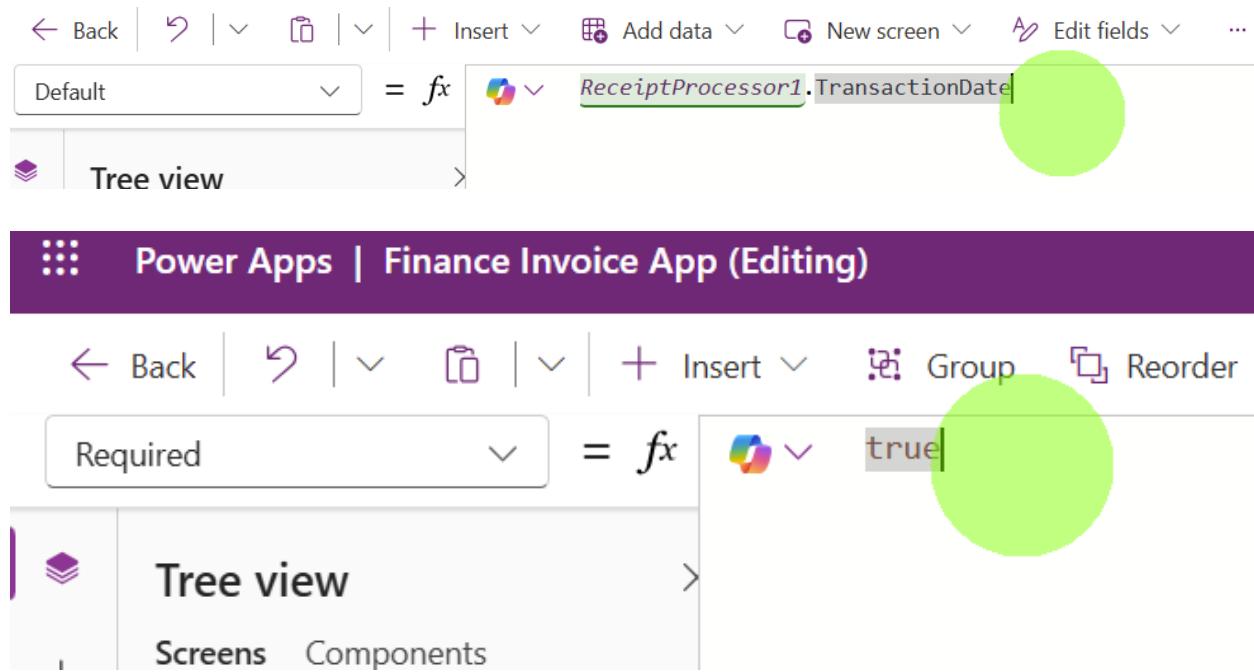
true

Tree view

Screens Components

4. Unlock InvoiceDate Card and set Default property to below and Required to true.

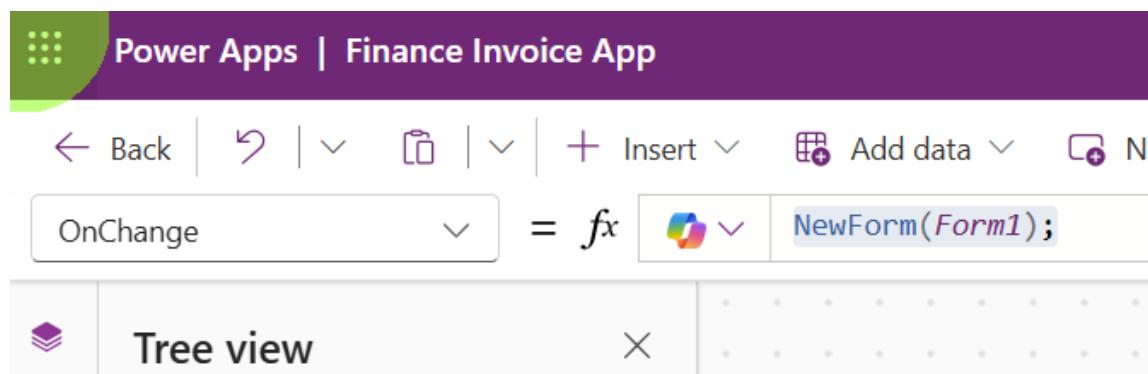
*ReceiptProcessor1*.TransactionDate



The screenshot shows the 'Edit fields' screen in the Power Apps studio. At the top, there are navigation icons: Back, Undo, Redo, Insert, Add data, New screen, Edit fields, and more. Below these are two dropdown menus: 'Default' set to `ReceiptProcessor1.TransactionDate` and 'Required' set to `true`. The main area is titled 'Tree view' with tabs for 'Screens' and 'Components'. A purple header bar at the top says 'Power Apps | Finance Invoice App (Editing)'.

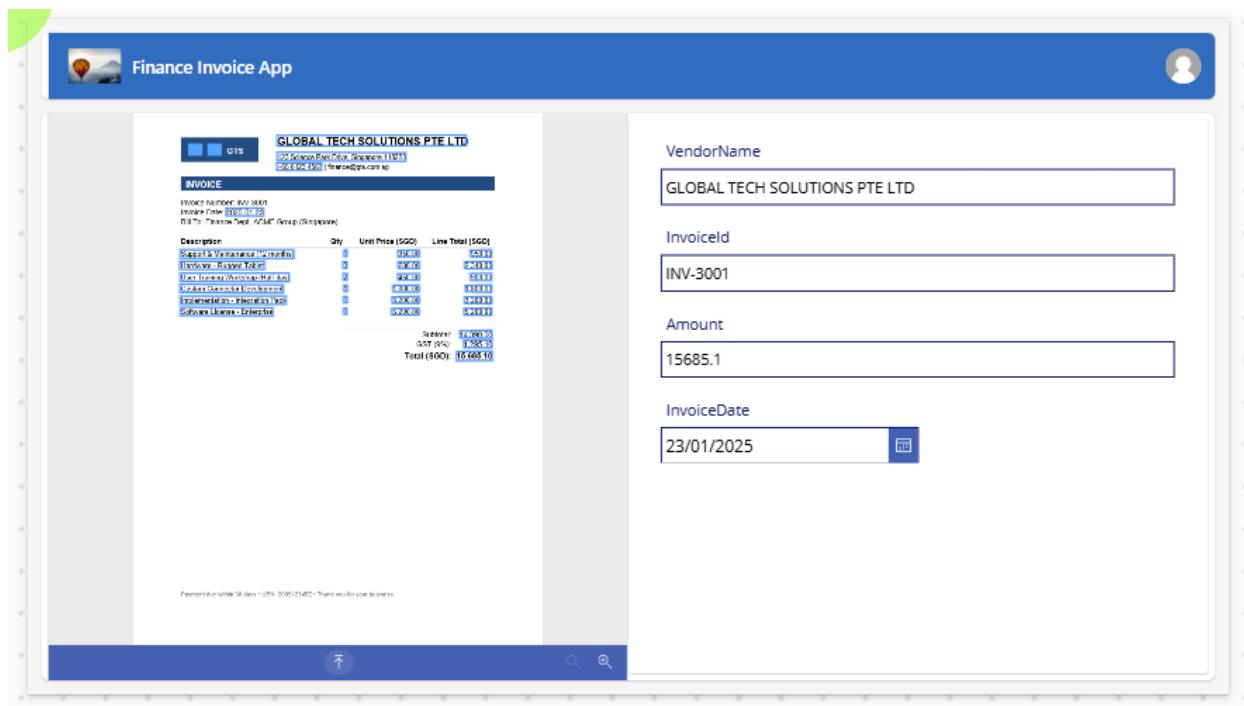
5. Change OnChange properties of the Receipt Processor to below.

`NewForm(Form1);`



The screenshot shows the 'Edit fields' screen in the Power Apps studio. At the top, there are navigation icons: Back, Undo, Redo, Insert, Add data, New screen, Edit fields, and more. Below these are two dropdown menus: 'OnChange' set to `NewForm(Form1);` and 'OnSelect' (partially visible). The main area is titled 'Tree view' with tabs for 'Screens' and 'Components'. A purple header bar at the top says 'Power Apps | Finance Invoice App'.

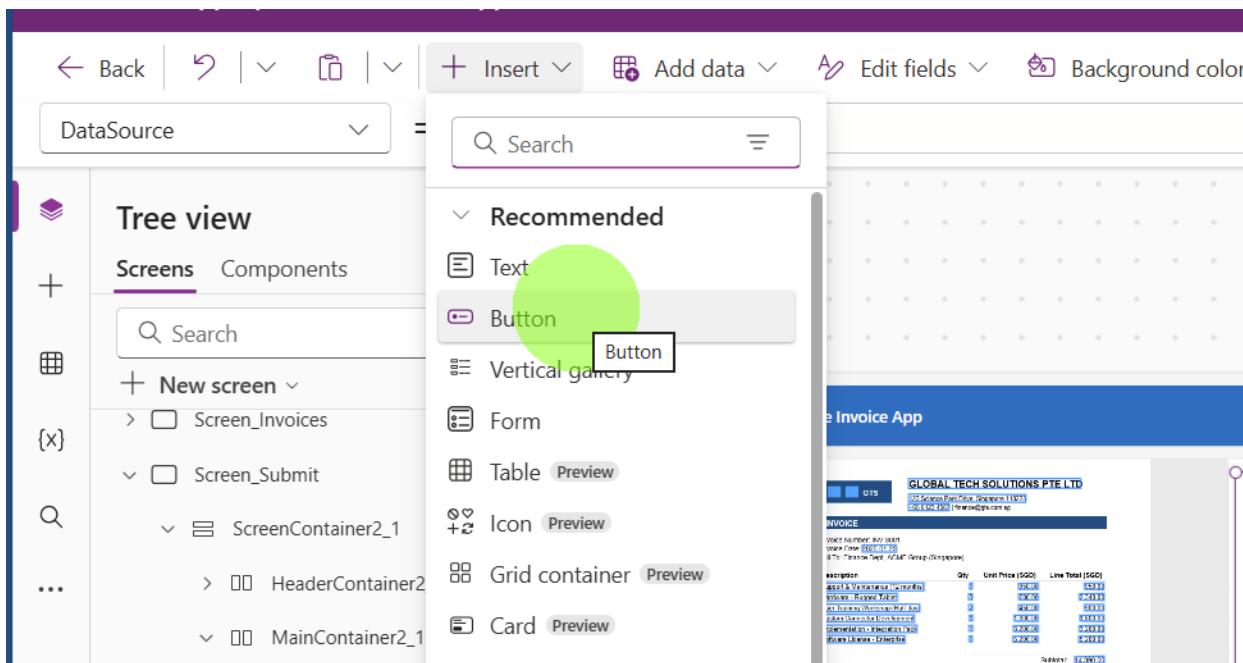
You can test with the sample invoices provided to you and the form should auto-populate the values based on invoice uploaded.



----- End of Task -----

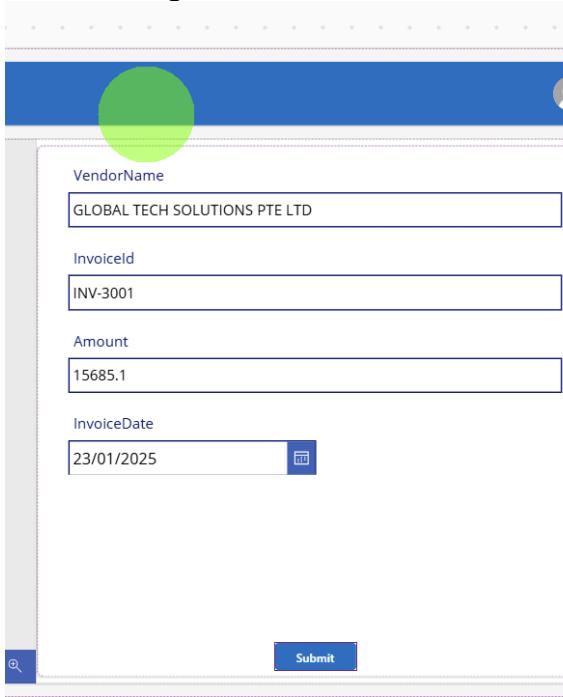
## Task 4: Submitting Extracted Values

1. Add a button below the form.



The screenshot shows the Microsoft Power Platform canvas editor interface. On the left, the "Tree view" pane is open, showing "Screens" selected. Below it, there is a search bar and a "+ New screen" dropdown. In the center, the "Components" pane is open, showing a list of recommended components: Text, Button, Vertical gallery, Form, Table, Icon, Grid container, and Card. A green circle highlights the "Button" component. On the right, the "Invoice App" screen is displayed, showing the same invoice grid and form as the previous screenshot, but with a new "Button" component added to the bottom of the form area.

## 2. Change Text to Submit



**Properties**

BUTTON  
ButtonCanvas3

**Display** Advanced

**General**

- Text: Submit
- Icon: None
- Layout: Icon before
- Accessible label: No value
- Visible: On (switch)

**Behavior**

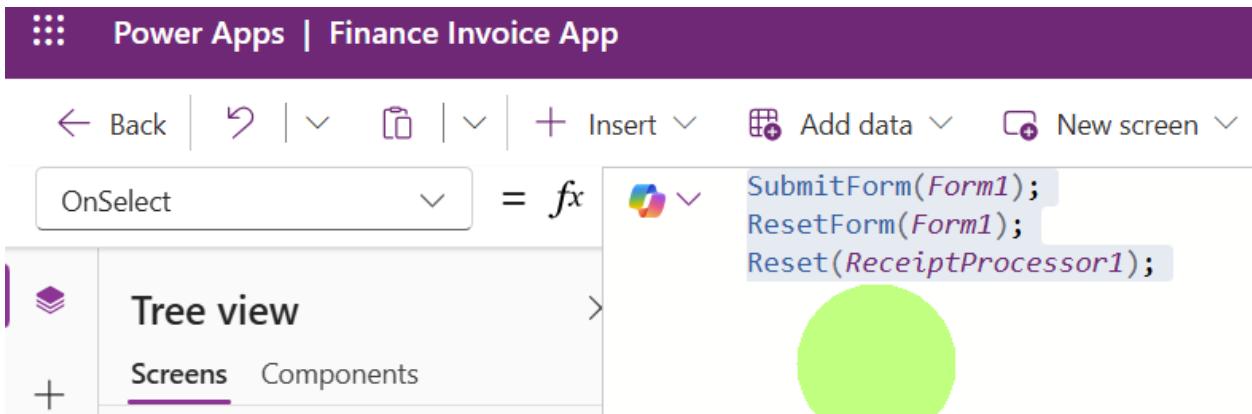
- Display mode: Edit

**Size and position**

- Align: Left
- Vertical align: Top
- Icon rotation: 0

## 3. Set OnSelect property of Submit button to this.

```
SubmitForm(Form1);
ResetForm(Form1);
Reset(ReceiptProcessor1);
```



OnSelect = *fx*

```
SubmitForm(Form1);
ResetForm(Form1);
Reset(ReceiptProcessor1);
```

## 4. Set DisplayMode property of Submit button to this.

```
If(Form1.Valid, DisplayMode.Edit, DisplayMode.Disabled)
```

... Power Apps | Finance Invoice App

Back | ↻ | ⌂ | ⌂ | + Insert | Add data | New screen | Settings | ...

DisplayMode = fx If(Form1.Valid, DisplayMode.Edit, DisplayMode.Disabled)

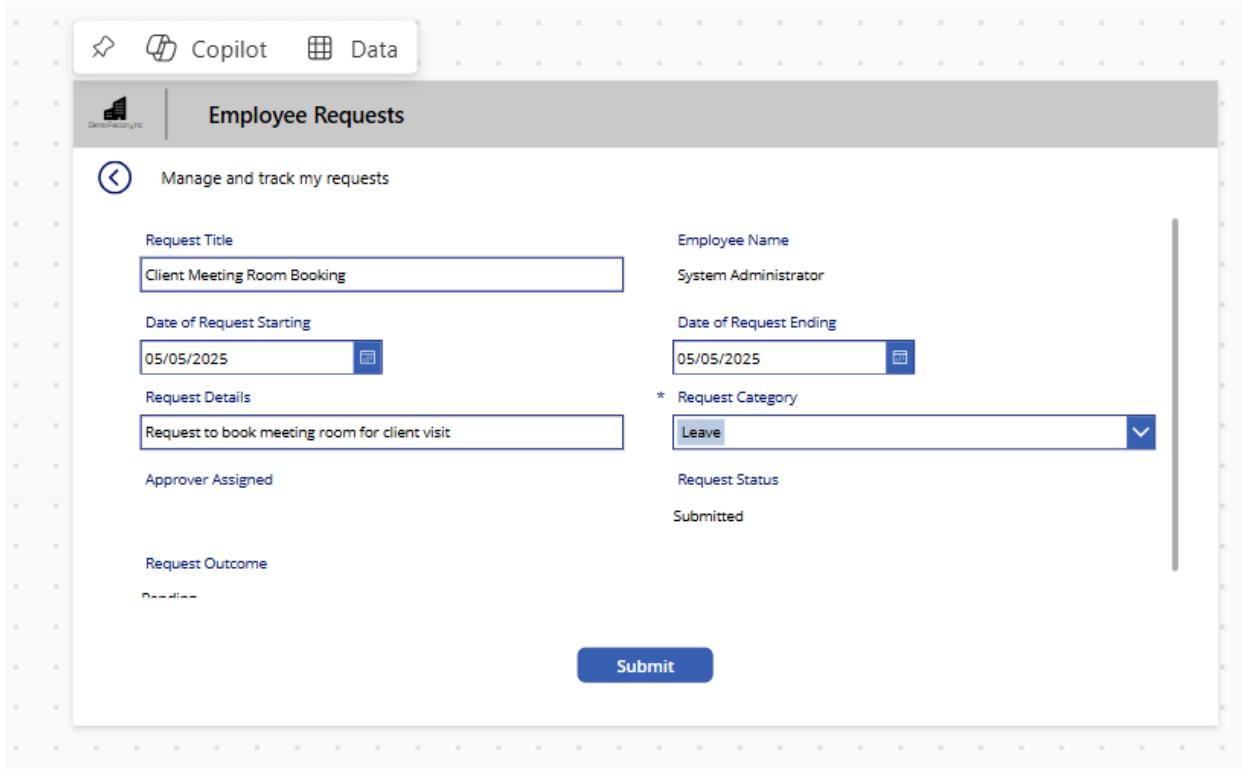
Tree view >

Screens Components



# Exercise 6: Creating Message Popups using Components

Utilising components to create reusable message popups.



The screenshot shows a Microsoft Power Platform canvas with a "Employee Requests" form component. The form includes fields for Request Title (Client Meeting Room Booking), Date of Request Starting (05/05/2025), Request Details (Request to book meeting room for client visit), Employee Name (System Administrator), Date of Request Ending (05/05/2025), Request Category (Leave), Approver Assigned, Request Status (Submitted), and Request Outcome (Pending). A "Submit" button is at the bottom.

## Objectives

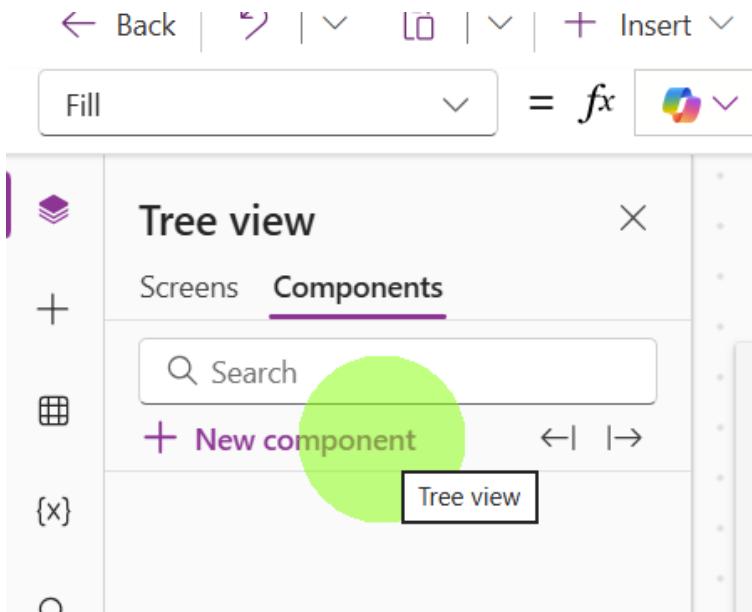
- Understand the use of components and custom properties

## Estimated time to complete this lab

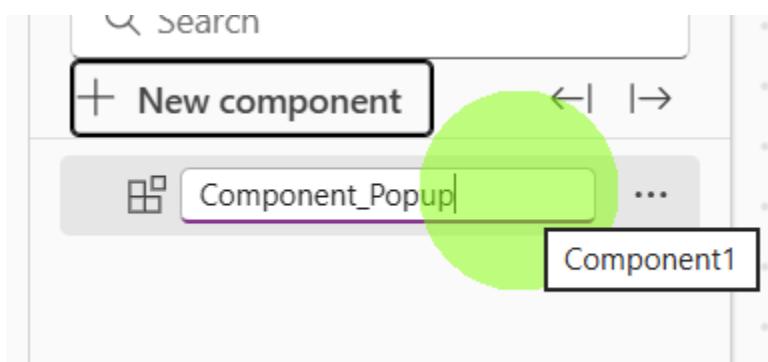
20 mins

## Task 1: Adding a New Component with Custom Properties

1. Switch to Components tab and click +New Component.



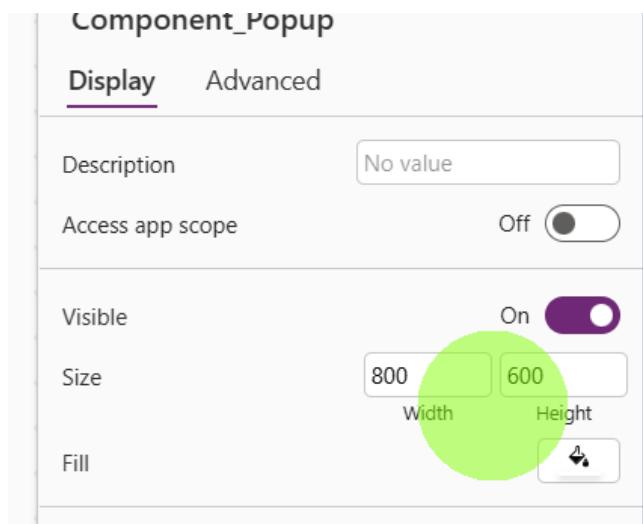
Rename it to Componenet\_Popup.



Set Width and Height of the component as below.

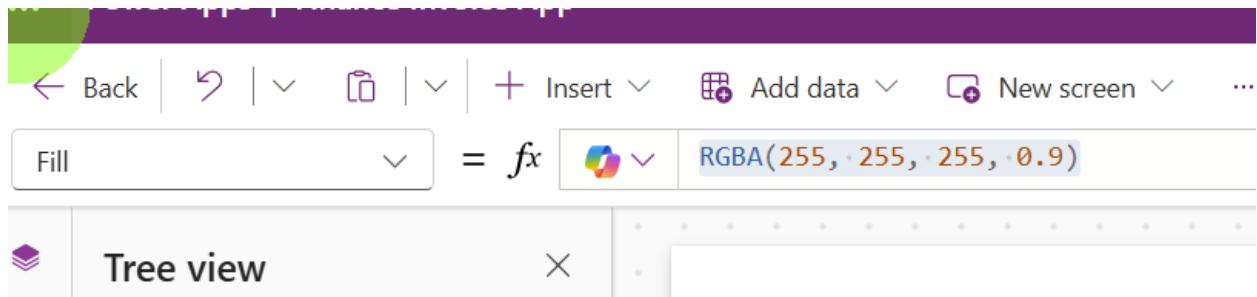
Width: 800

Height: 600

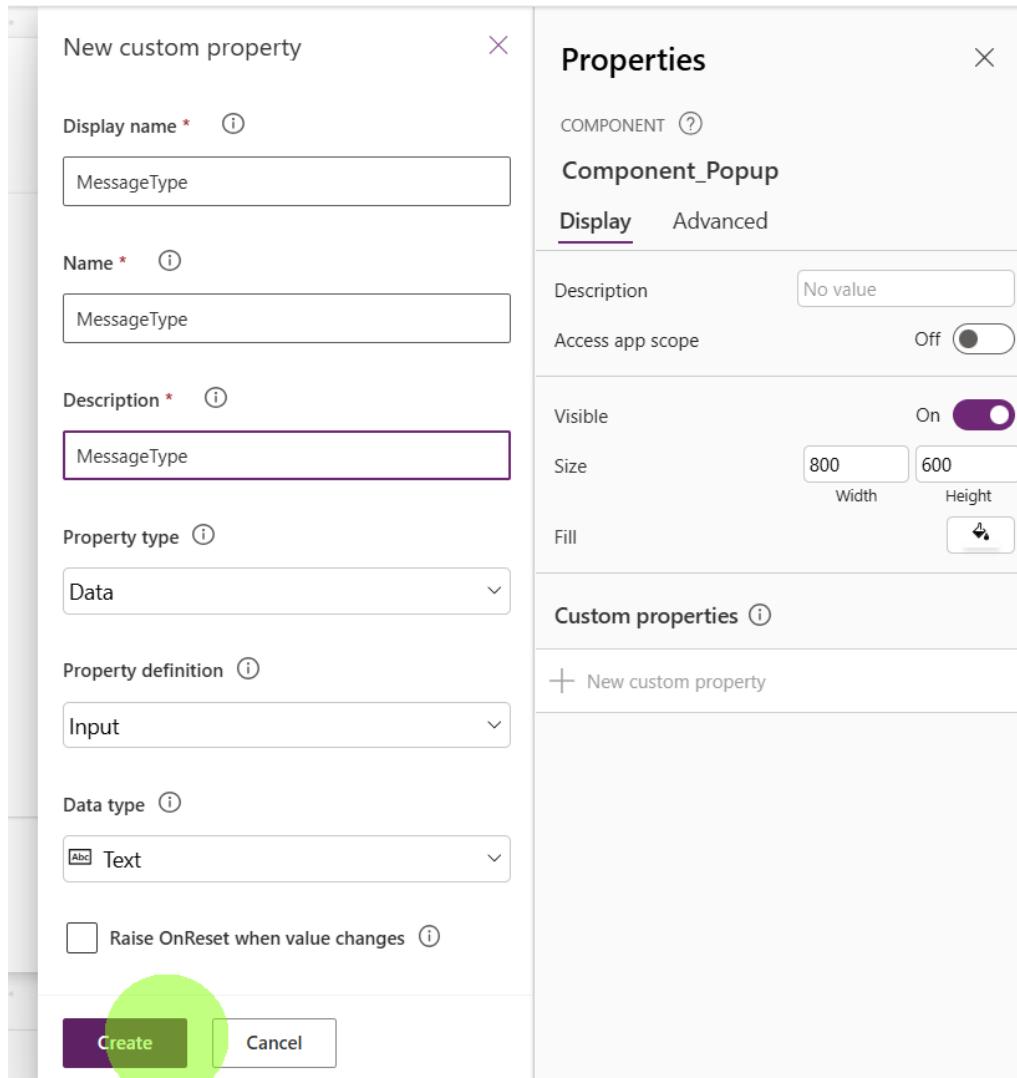


Set Fill Property to the component as below.

RGB(255, 255, 255, 0.9)



2. Add a new Custom Property (MessageType) to the Component as below.



The screenshot shows two side-by-side panels in the Microsoft Power Platform interface.

**New custom property (Left Panel):**

- Display name:** MessageType
- Name:** MessageType
- Description:** MessageType
- Property type:** Data (Input)
- Data type:** Text
- Checkboxes:** Raise OnReset when value changes (unchecked)

**Properties (Right Panel):**

- COMPONENT:** Component\_Popup
- Display** tab selected.
- Description:** No value
- Access app scope:** Off (switch off)
- Visible:** On (switch on)
- Size:** Width: 800, Height: 600
- Fill:** (button)
- Custom properties:** New custom property

3. Add one more Custom Property (Message) as below.

### New custom property

Display name \*

Name \*

Description \*

Property type

Property definition

Data type

Raise OnReset when value changes

**Create** **Cancel**

### Properties

COMPONENT

**Display** **Advanced**

Description	<input type="text" value="No value"/>
Access app scope	<input checked="checked" type="checkbox"/>
Visible	<input checked="checked" type="checkbox"/>
Size	<input type="text" value="800"/> Width <input type="text" value="600"/> Height
Fill	<input type="checkbox"/>

Custom properties

+ New custom property

4. Add one more Custom Property (ClickEvent) as below.

New custom property ×

Display name \* (i)

Name \* (i)

Description \* (i)

Property type (i)

Return data type (i)

**Parameters**  
+ New parameter

Create Cancel

**Properties** ×

COMPONENT (?)  
**Component\_Popup**

**Display** Advanced

Description   
Access app scope Off (On)

Visible Off On (On)

Size 800 600  
Width Height (Fill)

**Custom properties** (i)

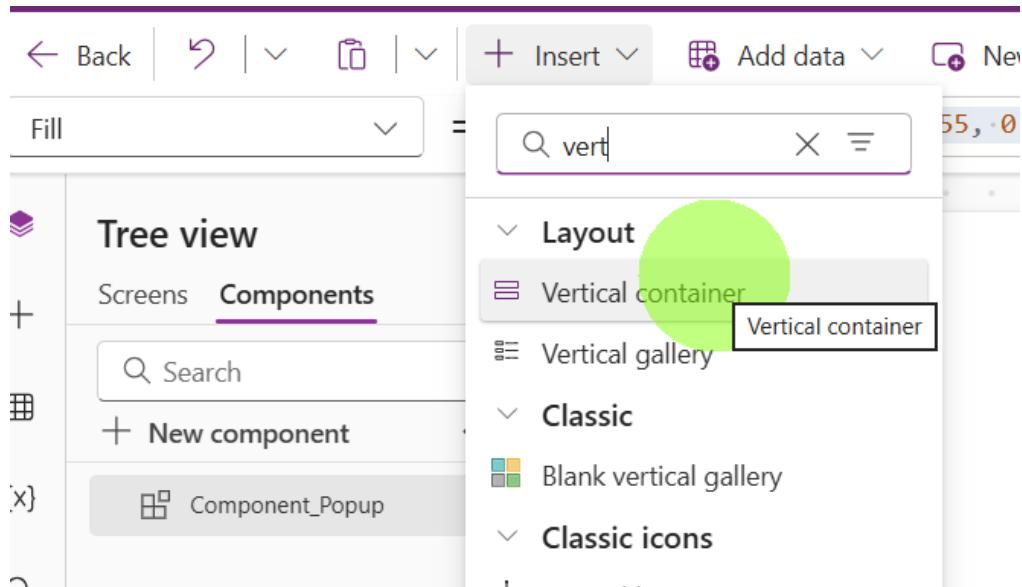
+ New custom property

Custom property	Type	Value	...
MessageType	Text	Text	...
Message	Text	Text	...

----- End of Task -----

## Task 2: Adding Controls into the Component

1. Add a new Vertical Container inside the component



Change below properties of the Vertical Container.

Width: 500

Height: 400

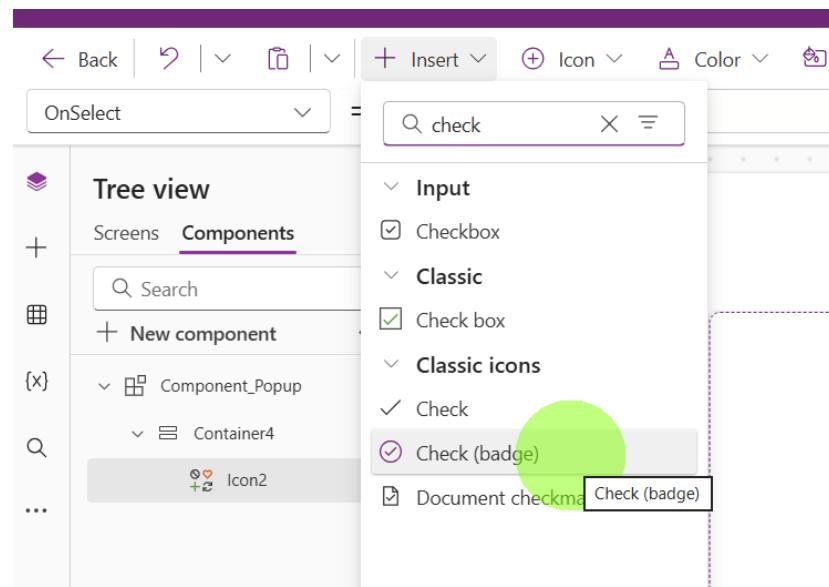
LayoutGap: 30

LayoutJustifyContent: `LayoutJustifyContent.Center`

X: `Parent.Width/2 - Self.Width/2`

Y: `Parent.Height/2 - Self.Height/2`

## 2. Insert a Check (badge) icon into the container.



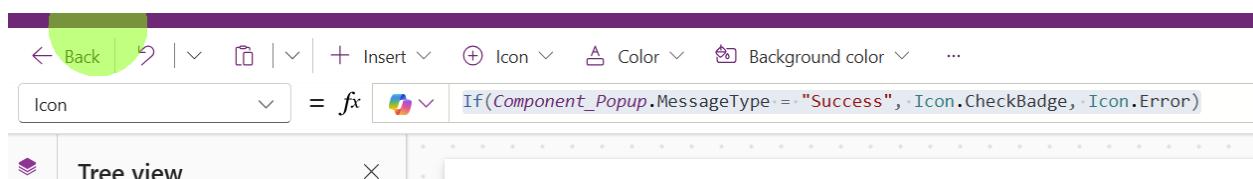
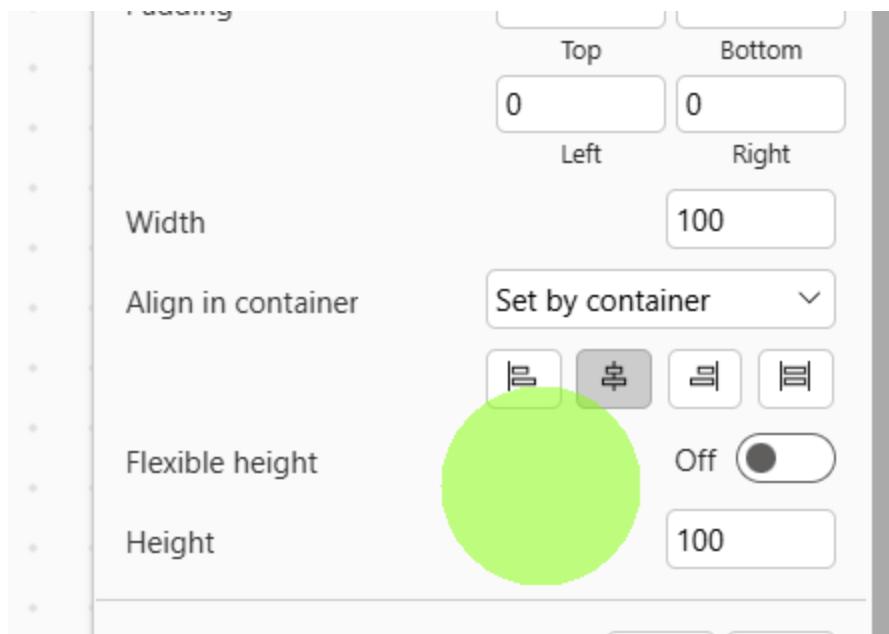
Change below properties of the Icon.

Width: 100

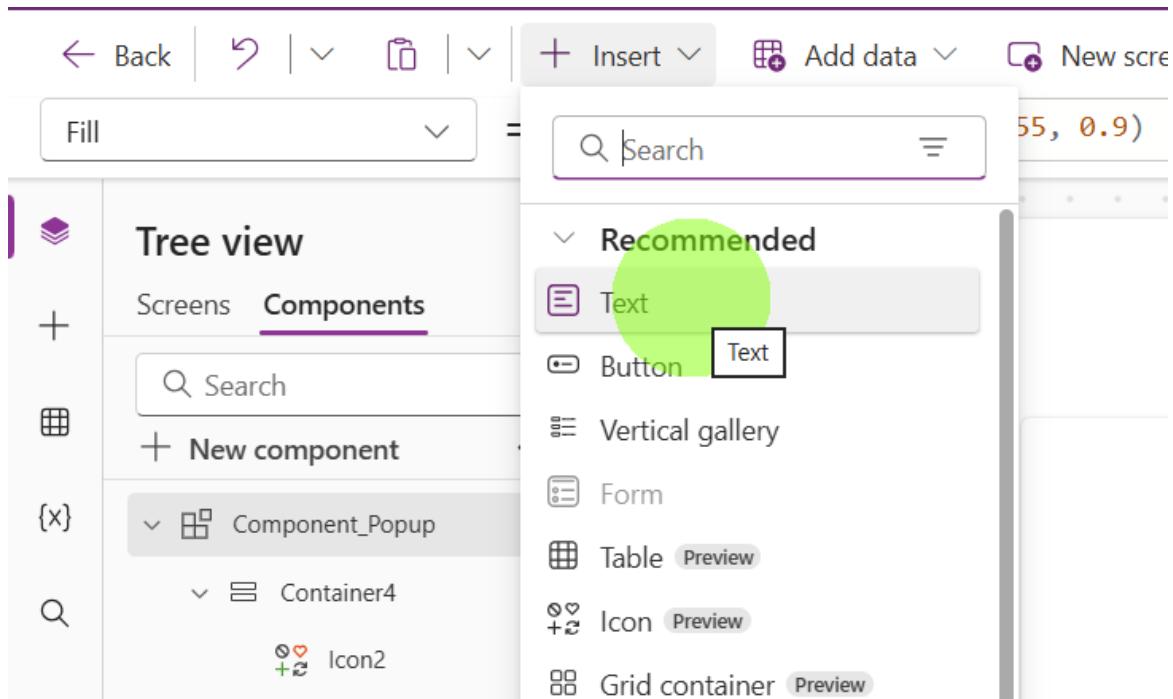
Height: 100

Color: `If(Component_Popup.MessageType = "Success", Color.DarkGreen, Color.DarkRed)`

Icon: `If(Component_Popup.MessageType = "Success", Icon.CheckBadge, Icon.Error)`



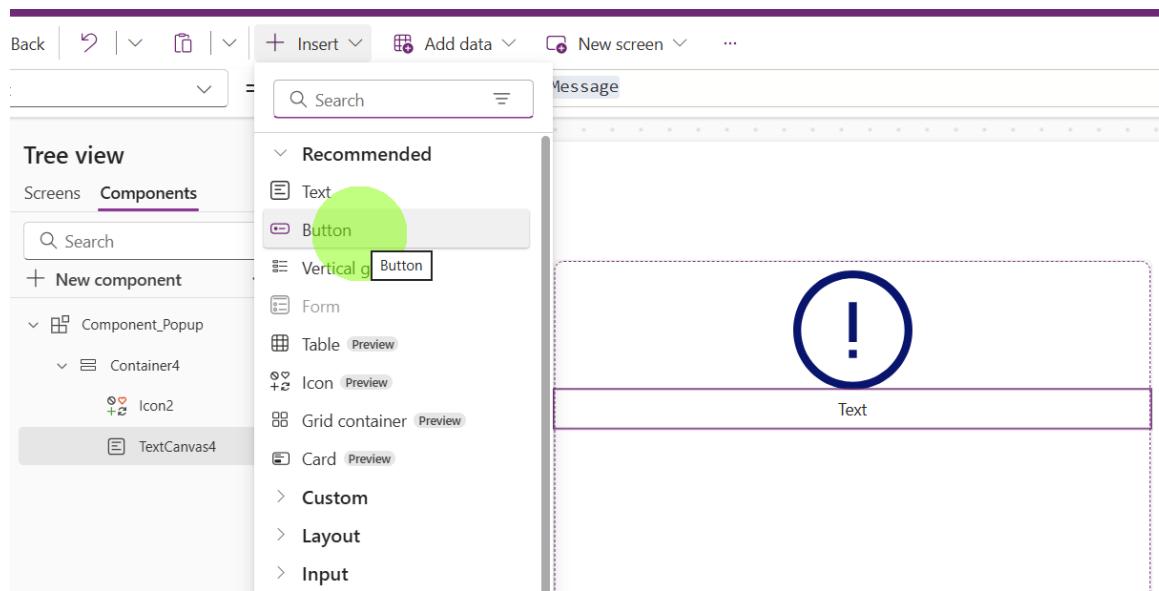
3. Insert a Text control into the container.



Change below properties of the Text.

Width: `Parent.Width`  
Align: `'TextCanvas.Align'.Center`  
VerticalAlign: `VerticalAlign.Middle`  
Icon: `Component_Popup.Message`

## 3. Insert a Button control into the container.

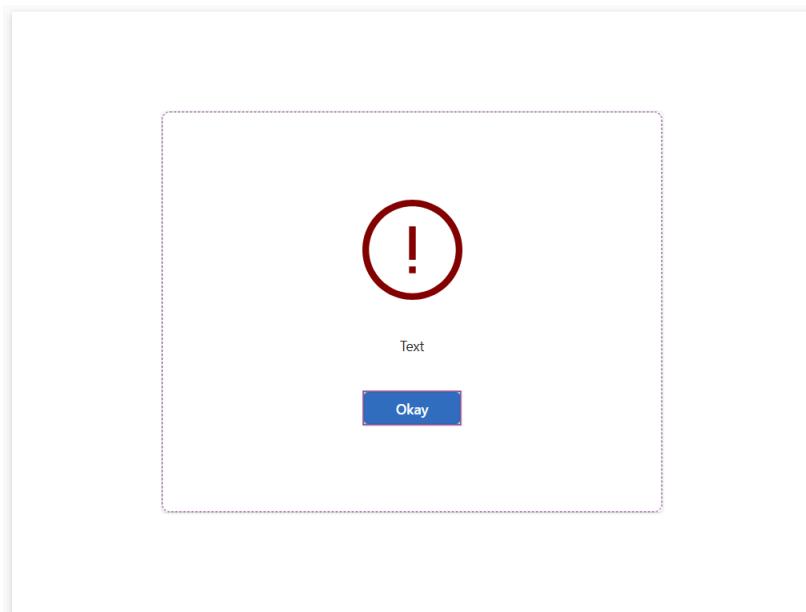


Change below properties of the Button.

Text: “**Okay**”

OnSelect: **Component\_Popup.ClickEvent()**

Your Component should look something like below.

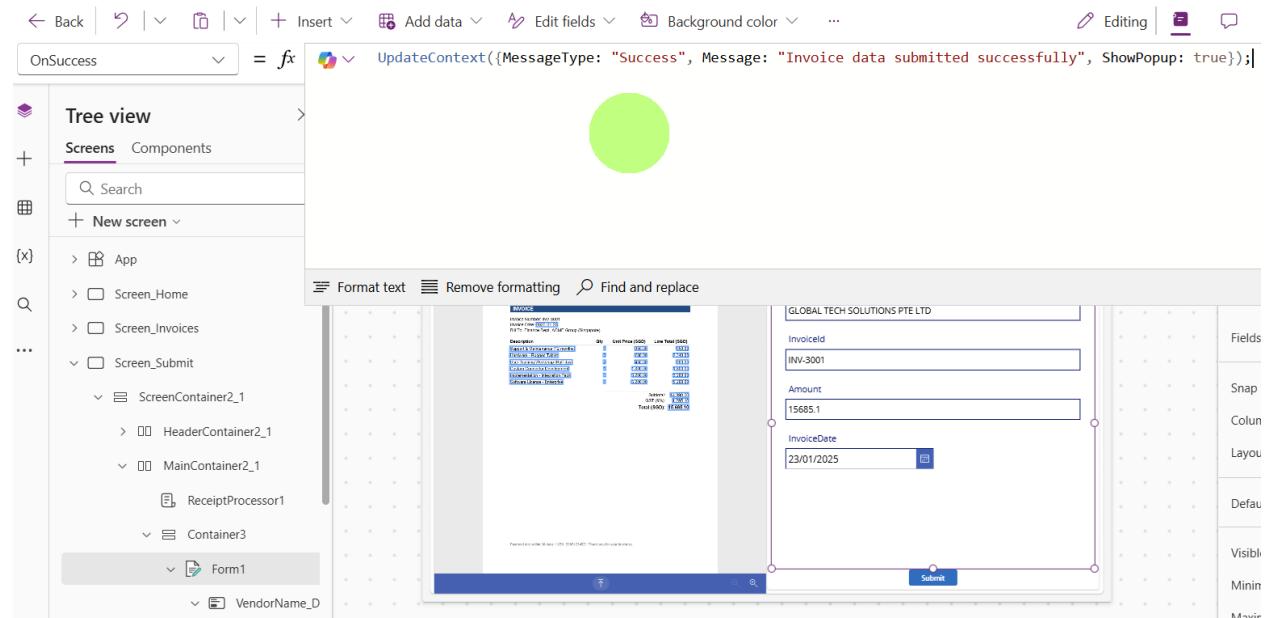


**----- End of Task -----**

## Task 3: Adding Popup Component to Submission Screen

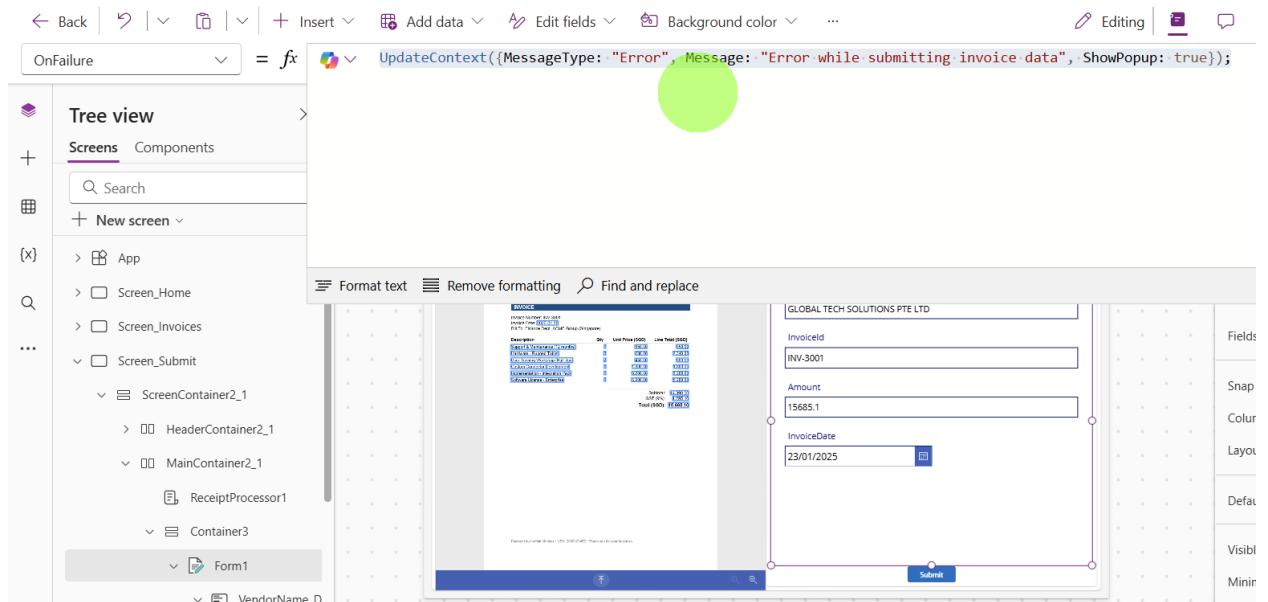
2. Select Form control and change OnSuccess proper to below.

```
UpdateContext({MessageType: "Success", Message: "Invoice data submitted successfully", ShowPopup: true});
```

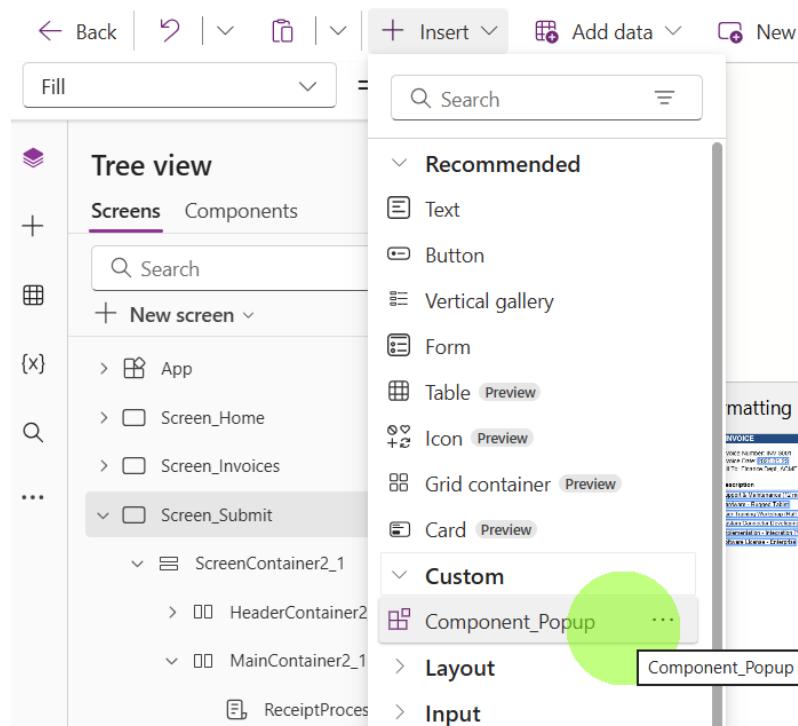


2. Select Form control and change OnFailure property to below.

```
UpdateContext({MessageType: "Error", Message: "Error while submitting invoice data", ShowPopup: true});
```



3. Insert Component\_Popup into Screen\_Submit.

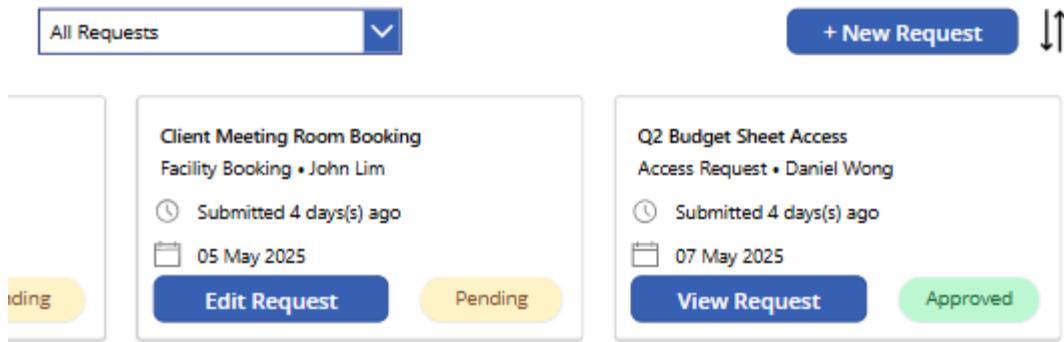


Change below properties of Component.

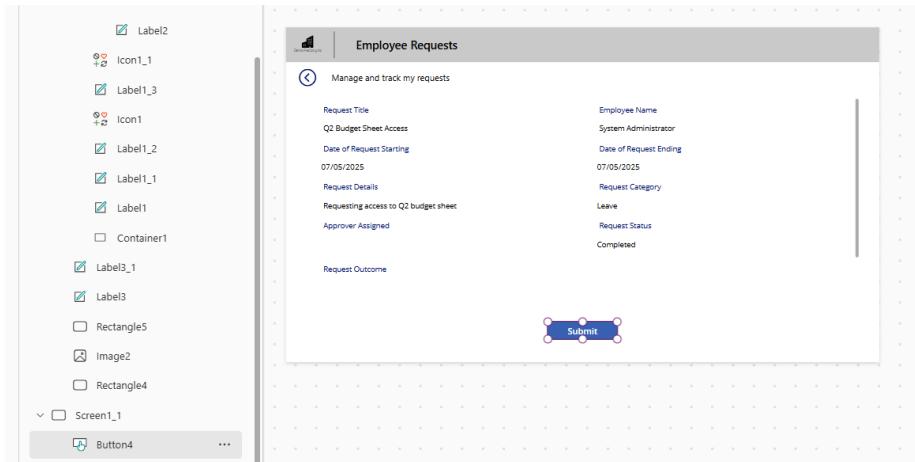
Width: `Parent.Width`  
Height: `Parent.Height`  
Visible: `ShowPopup`  
ClickEvent: `UpdateContext({ShowPopup:false});`  
MessageType: `MessageType`  
Message: `Message`

4. Change the text property of the button to the following value

```
If(ThisItem.'Request Status'.Value = "Completed", "View Request", "Edit Request")
```

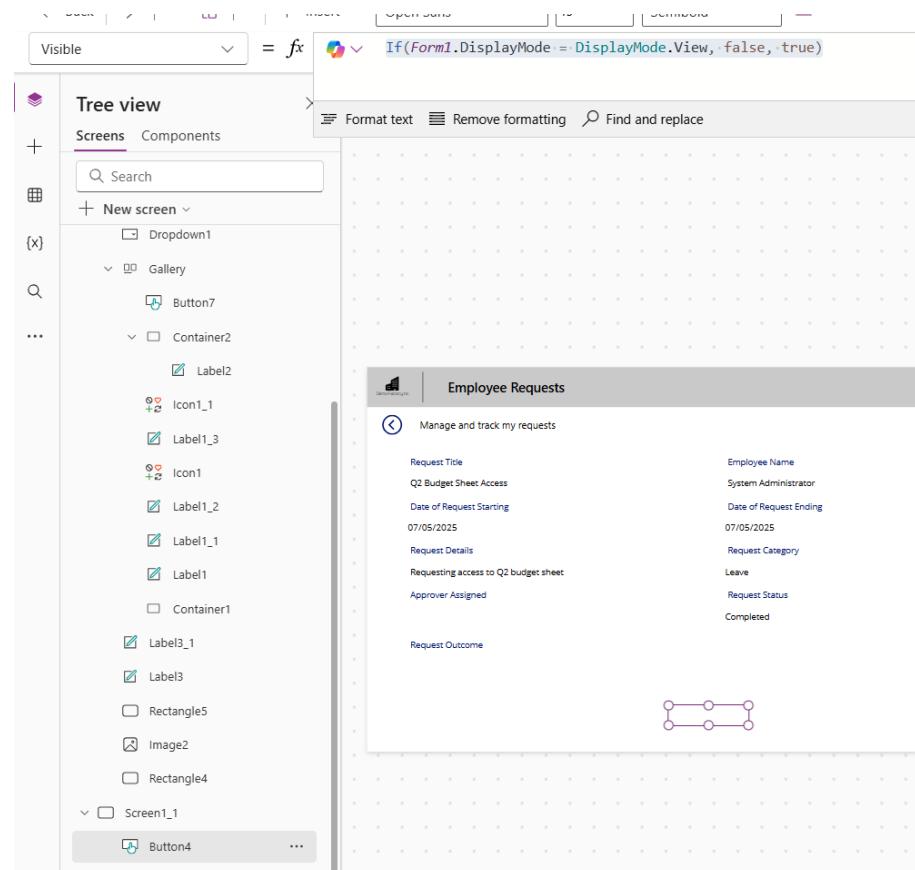


5. Select the "submit" button on the Form page.



Change the visible property of the button to the following value

```
If(Form1.DisplayMode = DisplayMode.View, false, true)
```



**End of Task**