Power Apps Documentation

This document will primarily describe how the navigation aspect of the application functions, and how it can be configured. For context, all members of the Creative Department—across teams such as Digital Creative, Copywriting, VM & Regional Visual Design, Visual Design, and Graphic Design—will be automatically redirected to their respective team's report dashboards. This ensures that access remains exclusive to each team, maintaining confidentiality without any overlap between different teams. Secondly, the application is designed to ensure that each team member can access only their own individual reports. This feature upholds exclusivity within the Creative Department, preventing unauthorized access and ensuring that no one can view another member's reports.

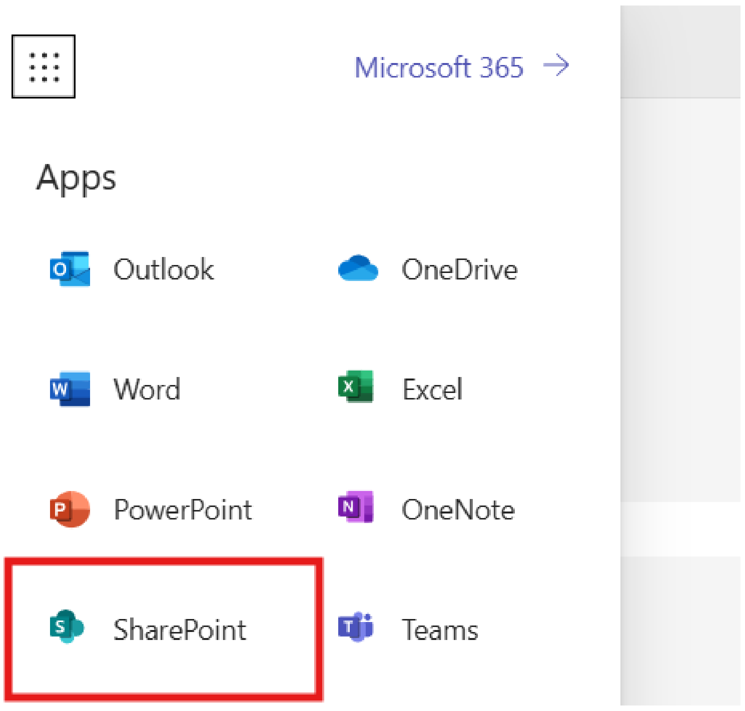
Creating Automatic Navigation Within Power Apps

Creating a SharePoint List

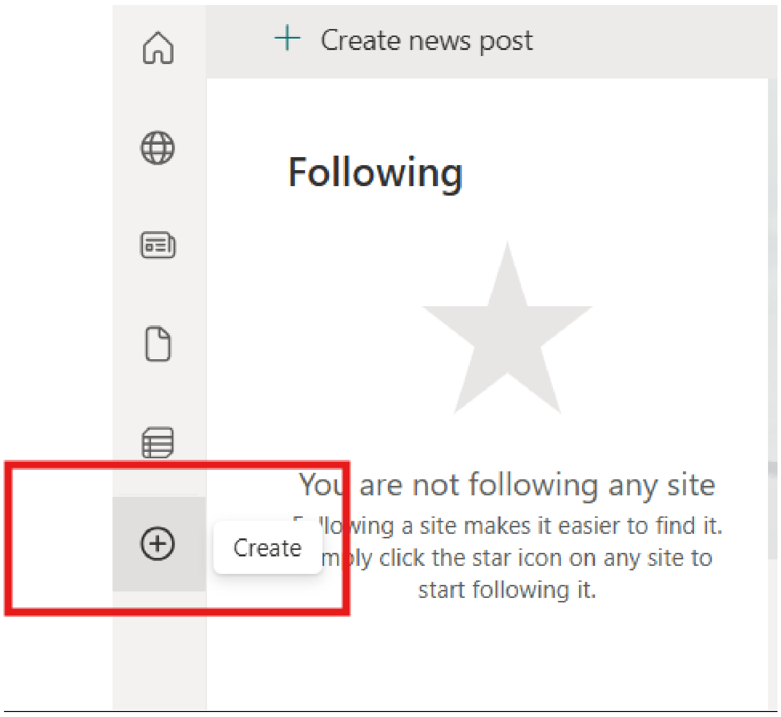
A SharePoint list is a collection of data that can be shared across members of your organisation. For automatic navigation, it will be used to associate the unique emails of each member within the Creative Department to their respective teams.

Steps:

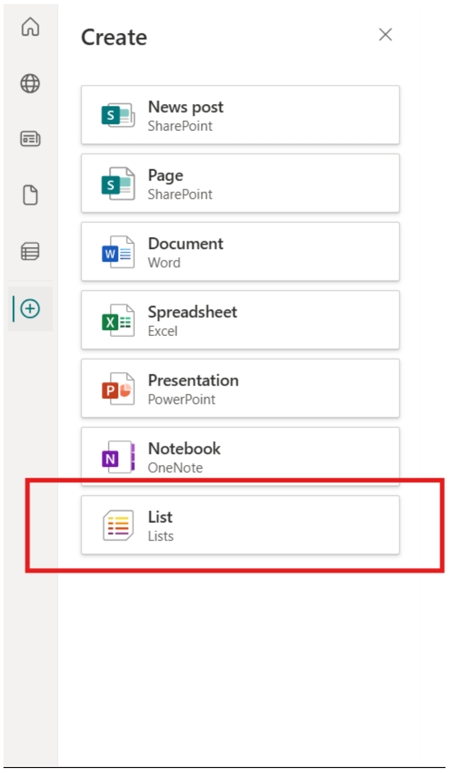
1. Access Microsoft SharePoint. You can either do this online or through Microsoft Teams.



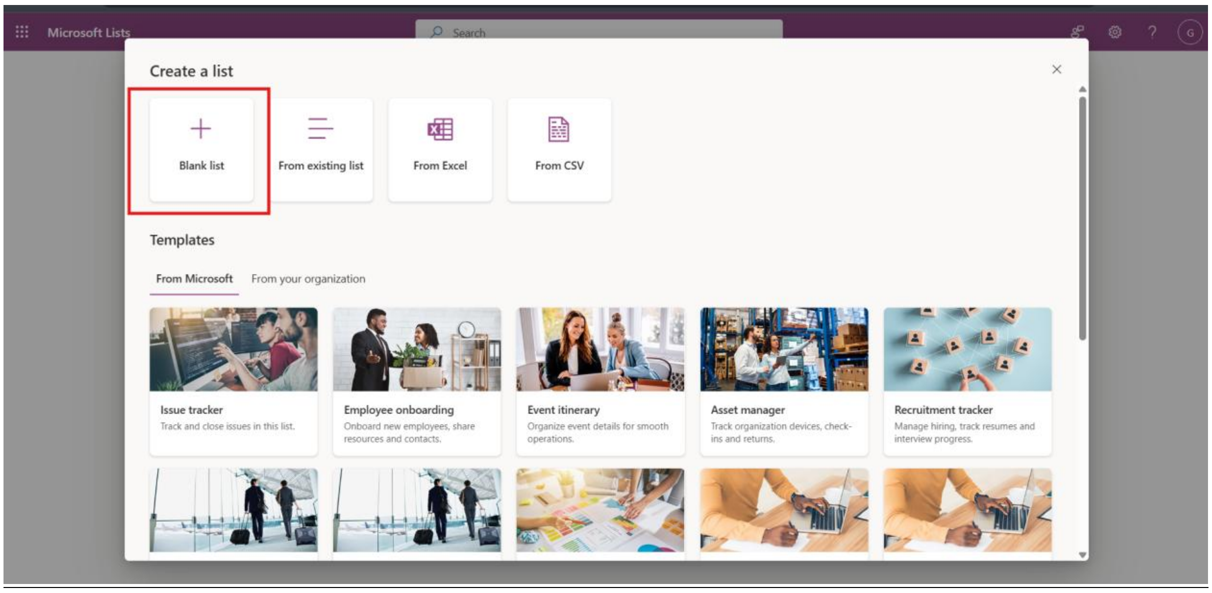
2. On the bar on the far left, click on 'Create'.


Create news post
Following
You are not following any site
Following a site makes it easier to find it. Simply click the star icon on any site to start following it.
Create

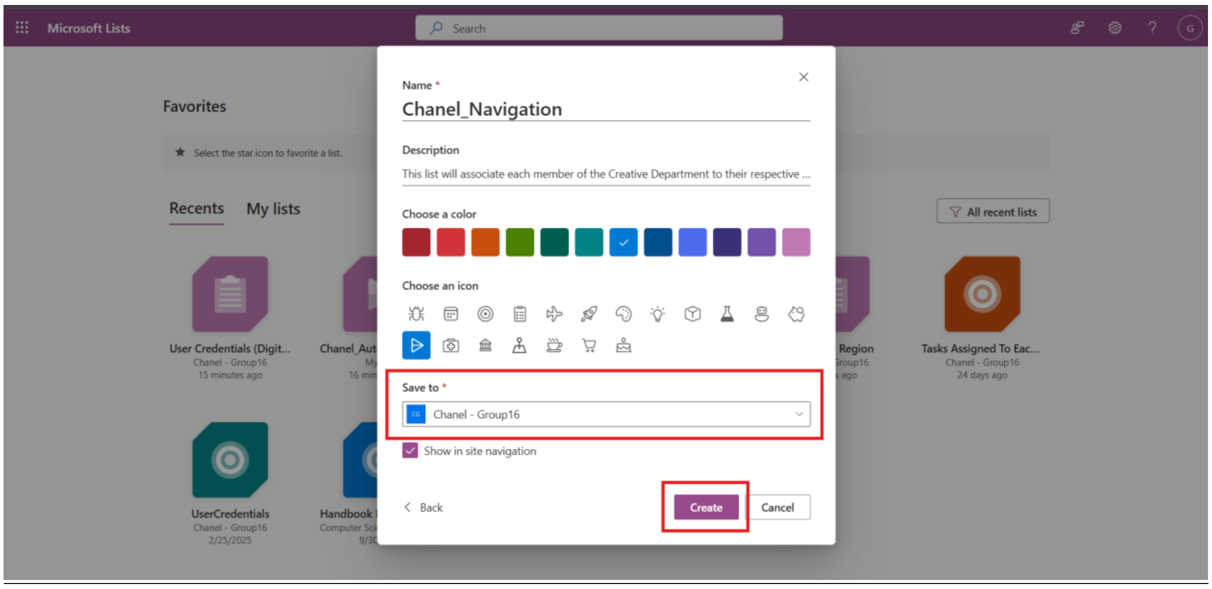
3. After pressing 'Create', click on 'List'.



4. After being redirected to a new page, select 'Blank List'.


Microsoft Lists
Search
Create a list
Blank list
From existing list
From Excel
From CSV
Templates
From Microsoft
From your organization
Issue tracker
Track and close issues in this list.
Employee onboarding
Onboard new employees, share resources and contacts.
Event itinerary
Organize event details for smooth operations.
Asset manager
Track organization devices, check-ins and returns.
Recruitment tracker
Manage hiring, track resumes and interview progress.

5. Enter the required details for your new SharePoint List. Save the list to a secure cloud drive—such as *Chanel-Group16* in this example—before selecting ‘Create’.

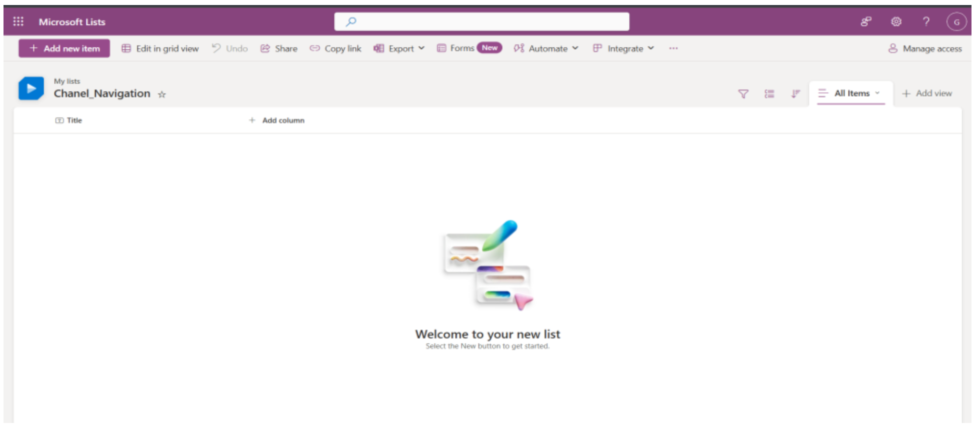


Name *
Chanel_Navigation

Description
This list will associate each member of the Creative Department to their respective ...

Save to *
Chanel - Group16

Your new list should look something like this:



Populating the SharePoint List

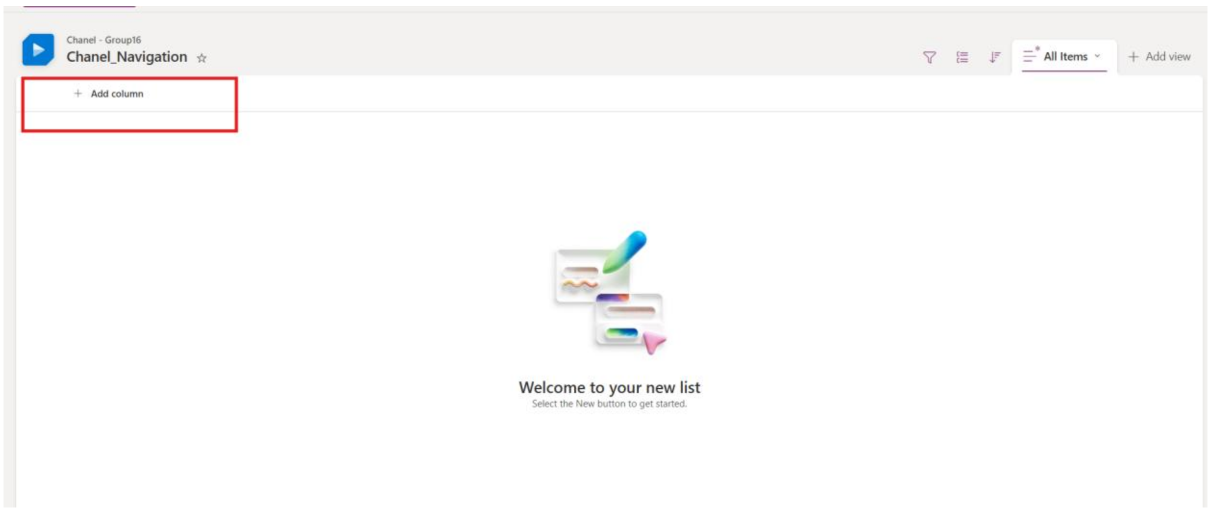
Once the SharePoint List has been created, each column must be populated with relevant information about the members of the Creative Department. To facilitate seamless navigation, the following details are required:

- Each member's unique organisational email
- Their team
- The name of their Individual Report Screen (more elaboration later).

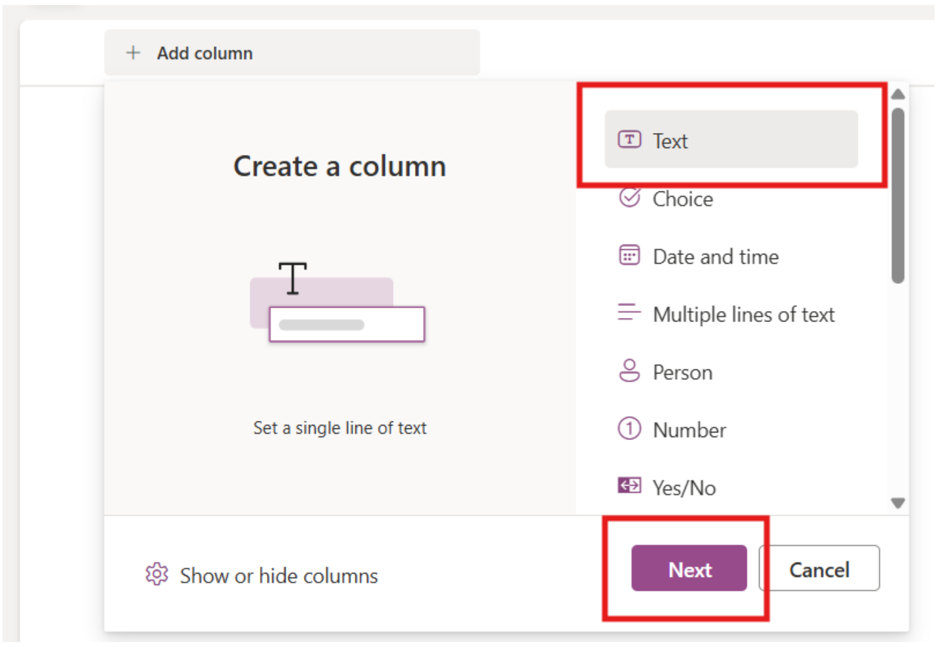
Ensure that all information is collected and on hand before populating the SharePoint List.

<u>Steps:</u>

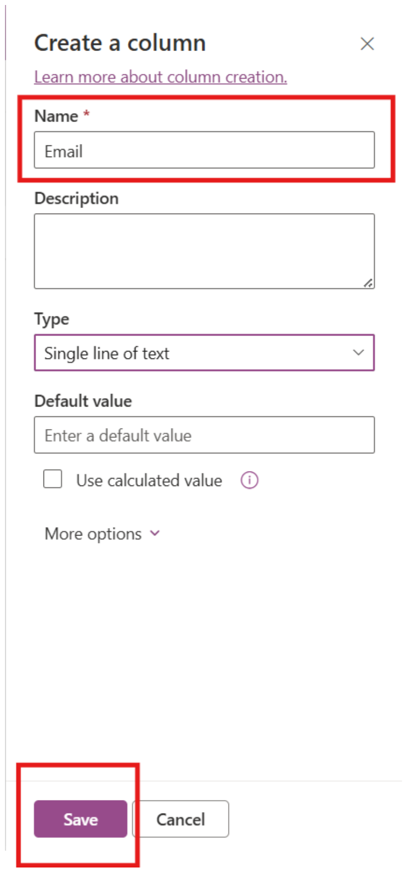
1. Firstly, the relevant columns in the SharePoint List must be created. Select 'Add column' in the header of the SharePoint List.



2. Select 'Text' when the pop-up is toggled. Select 'Next' when done.



3. A new pop-up to the right of the screen will appear. Name the new column as ‘Email’, the type of input to be ‘Single line of text’. Then press ‘Save’.



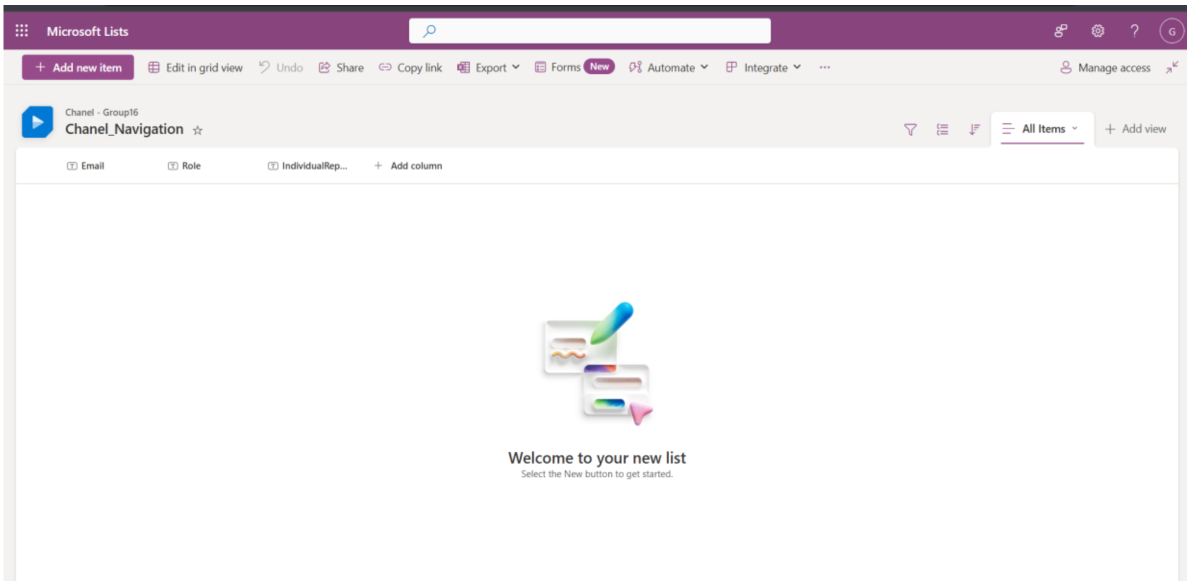
Name *
Email

Type
Single line of text

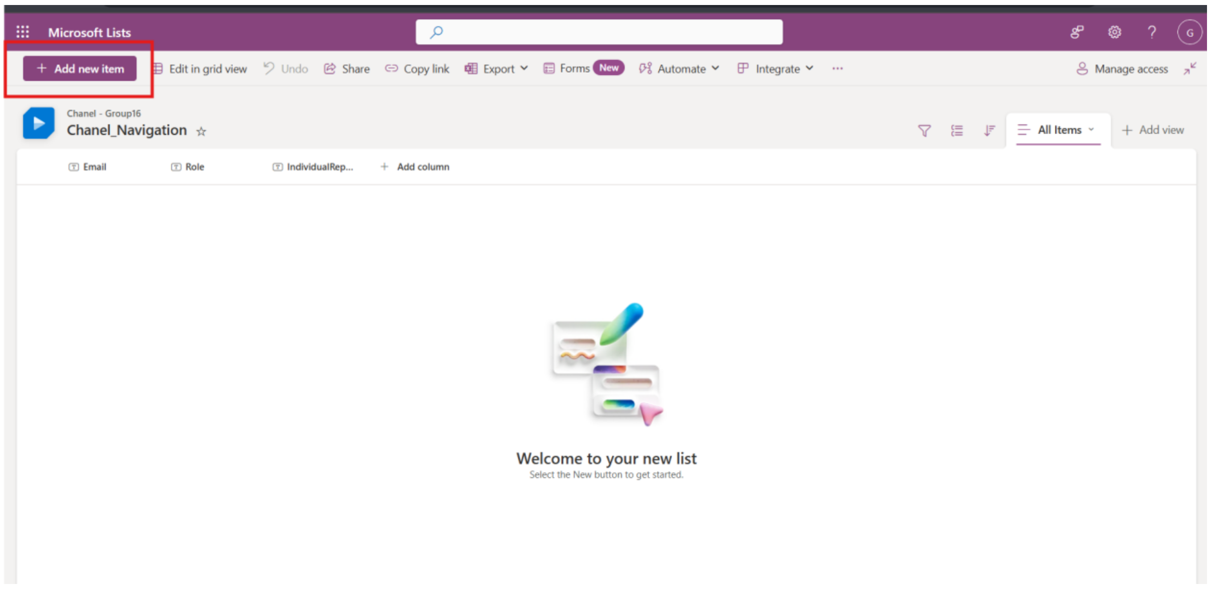
Default value

Use calculated value ☐

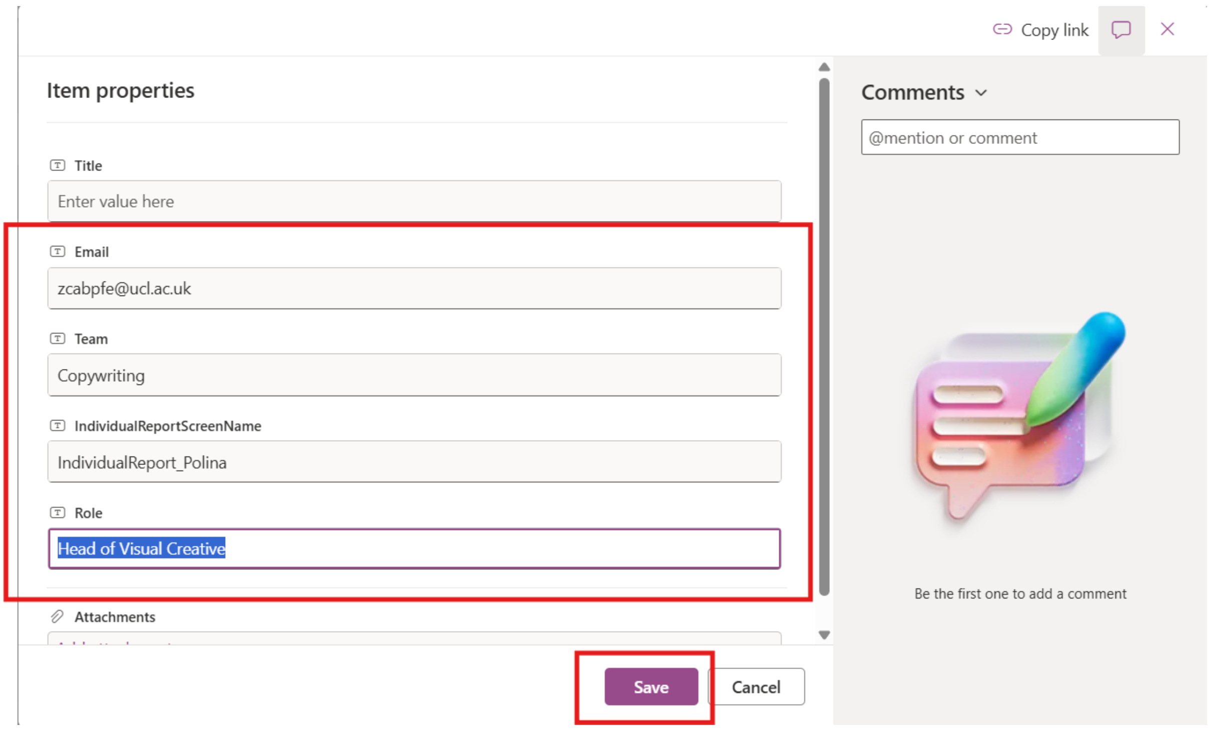
4. Repeat Steps 1 to 3 to create three more columns called ‘Team’, ‘Role’ and ‘IndividualReportScreenName’. The SharePoint List should look like this:



5. To populate the SharePoint List with information, select the ‘Add new item’ button in the top left corner.



6. When the pop-up appears, enter the necessary details for each member of the Creative Department. While this example uses UCL emails, the official Chanel organisation emails should be used instead. For the ‘Role’ column, enter the member’s role within their respective team. For the ‘IndividualReportScreenName’ column, enter the value in the format: ‘IndividualReport_[Member’sName]’ (e.g., *IndividualReport_Polina*). Click on ‘Save’ when done.

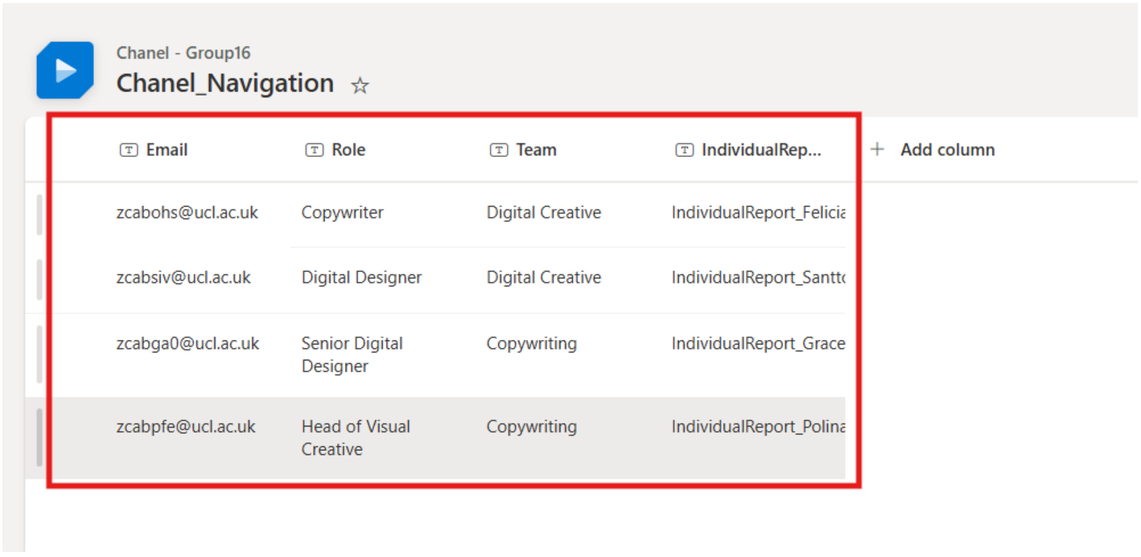


Item properties

Title	Enter value here
Email	zcabpfe@ucl.ac.uk
Team	Copywriting
IndividualReportScreenName	IndividualReport_Polina
Role	Head of Visual Creative
Attachments	

@mention or comment

7. Repeat Step 6 for all members of the Creative Department. In this example, only members of the Digital Creative & Copywriting team are included.



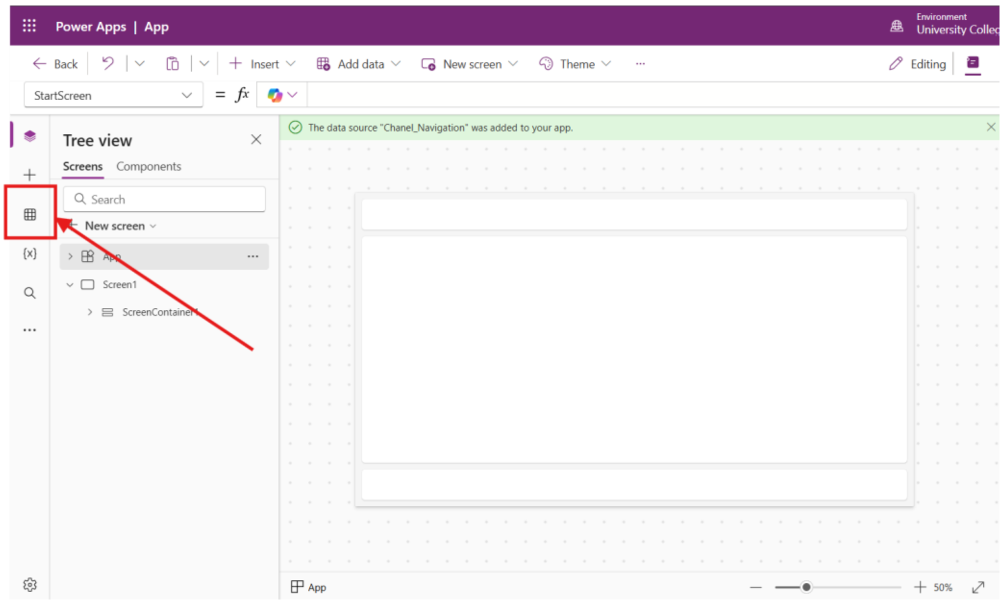
Email	Role	Team	IndividualRep...
zcabohs@ucl.ac.uk	Copywriter	Digital Creative	IndividualReport_Felicia
zcabsiv@ucl.ac.uk	Digital Designer	Digital Creative	IndividualReport_Santt
zcabga0@ucl.ac.uk	Senior Digital Designer	Copywriting	IndividualReport_Grace
zcabpfe@ucl.ac.uk	Head of Visual Creative	Copywriting	IndividualReport_Polina

Connecting the SharePoint List to Power Apps

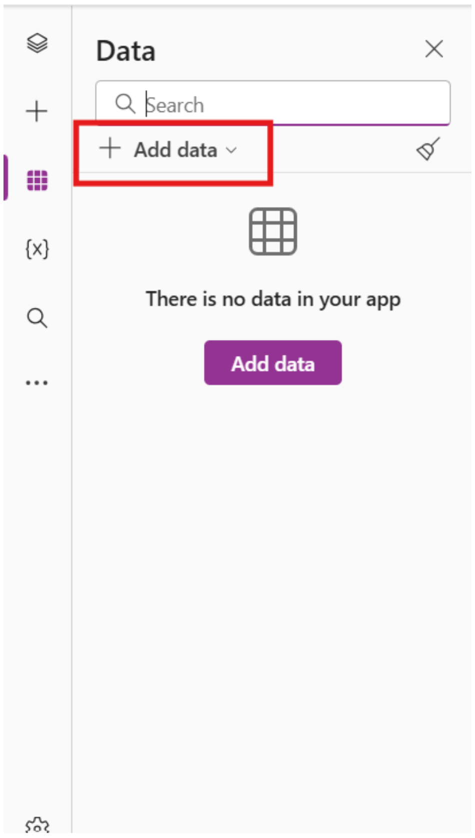
To utilise the information within the SharePoint List in the Power App, it must be connected as a data source. Ensure that a new, blank Power App canvas has been created before connecting your SharePoint List.

Steps:

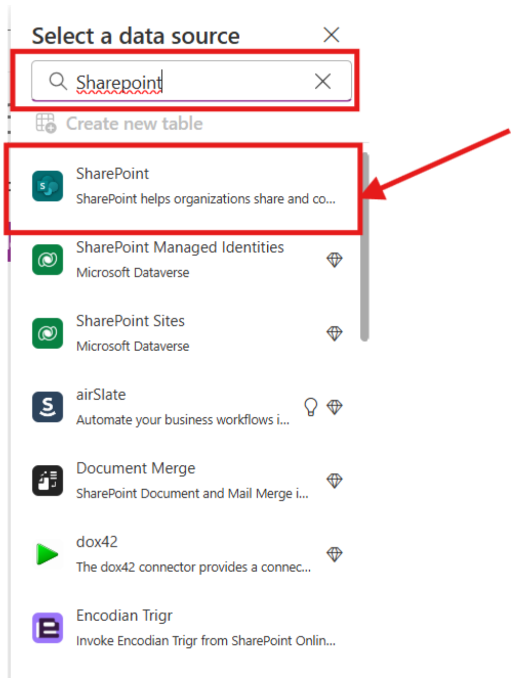
1. Select the ‘Data’ button located on the far-left sidebar of the Power Apps interface.



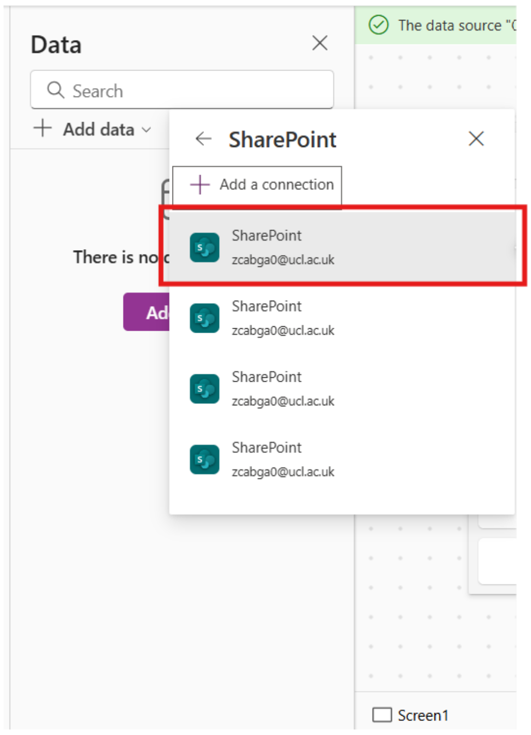
2. Select ‘Add data’ when the sidebar expands.



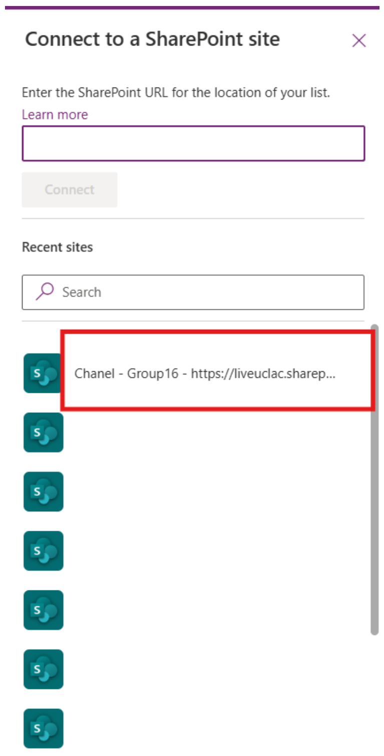
3. When the pop-up appears, type in ‘SharePoint’ in the search bar. Select ‘SharePoint’ when the option appears.


Select a data source
Sharepoint
Create new table
SharePoint
SharePoint helps organizations share and co...
SharePoint Managed Identities
Microsoft Dataverse
SharePoint Sites
Microsoft Dataverse
airSlate
Automate your business workflows i...
Document Merge
SharePoint Document and Mail Merge i...
dox42
The dox42 connector provides a connec...
Encodian Trigr
Invoke Encodian Trigr from SharePoint Onlin...

4. Select the first option that appears when the pop-up is toggled.



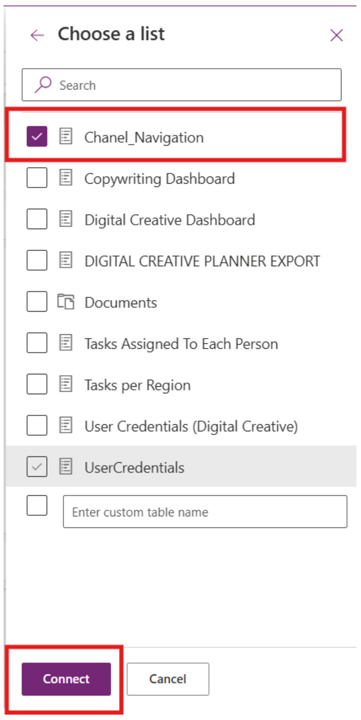
5. Choose the drive where the SharePoint list is stored. In this example, select *Chanel–Group 16*, as specified in Step 5 of the SharePoint List creation process.


Connect to a SharePoint site

Enter the SharePoint URL for the location of your list.
Learn more

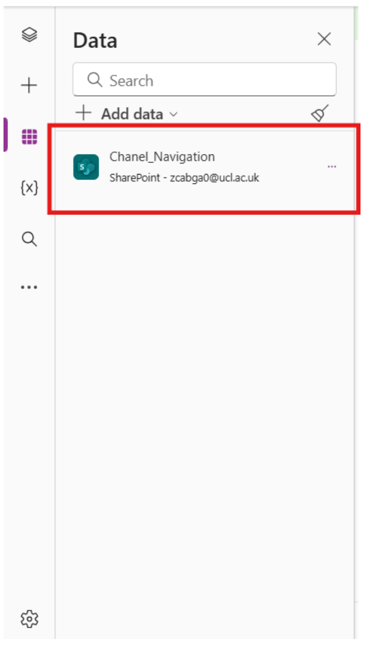
Chanel - Group16 - https://liveuclac.sharep...

6. Select the populated SharePoint that has each member's details for navigation. Select 'Connect'.



- [✓ Chanel_Navigation]
- Copywriting Dashboard
- Digital Creative Dashboard
- DIGITAL CREATIVE PLANNER EXPORT
- Documents
- Tasks Assigned To Each Person
- Tasks per Region
- User Credentials (Digital Creative)
- [✓ UserCredentials]
- Enter custom table name

7. When the SharePoint list has been successfully connected, it should show up in the Data bar on the left.



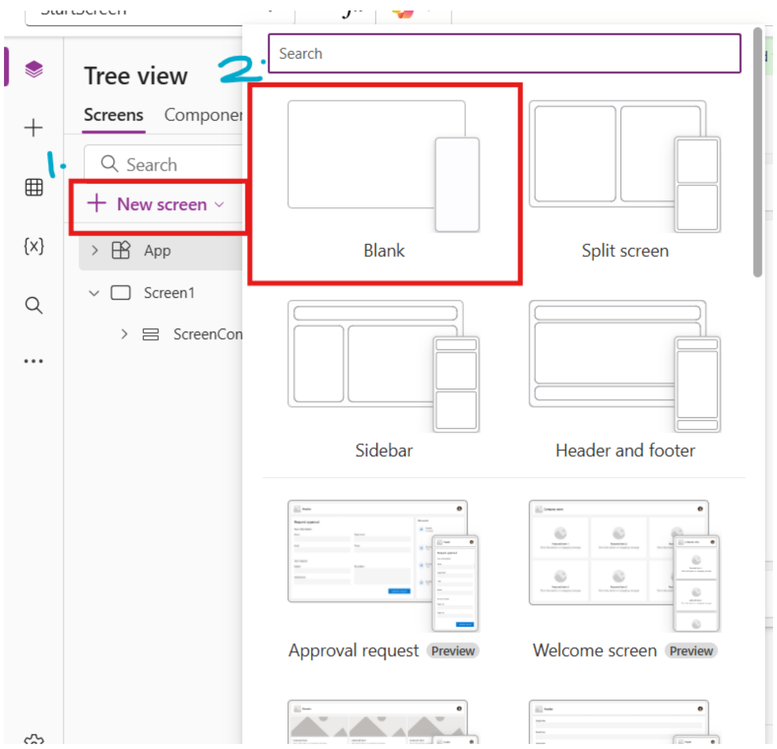
Implementing Automatic Navigation in Power Apps

This feature ensures that upon logging into the application, users are automatically redirected to their respective team's report dashboard based on their email. The navigation process will be facilitated using two key formulas:

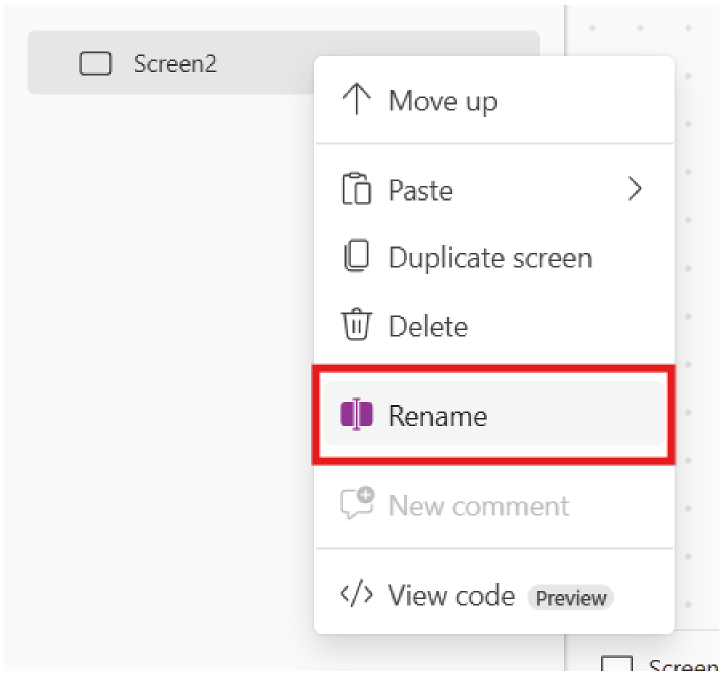
- LookUp: this function fetches the user's team based on their email. It searches the data source for a record where the 'Email' column matches the currently logged-in user's email (retrieved with User().Email). It then retrieves the value from the Team column associated with that user. This effectively determines which team the user belongs to.
- Switch: the Switch function evaluates the team name retrieved from LookUp and determines which dashboard page to navigate to.

<u>Steps:</u>

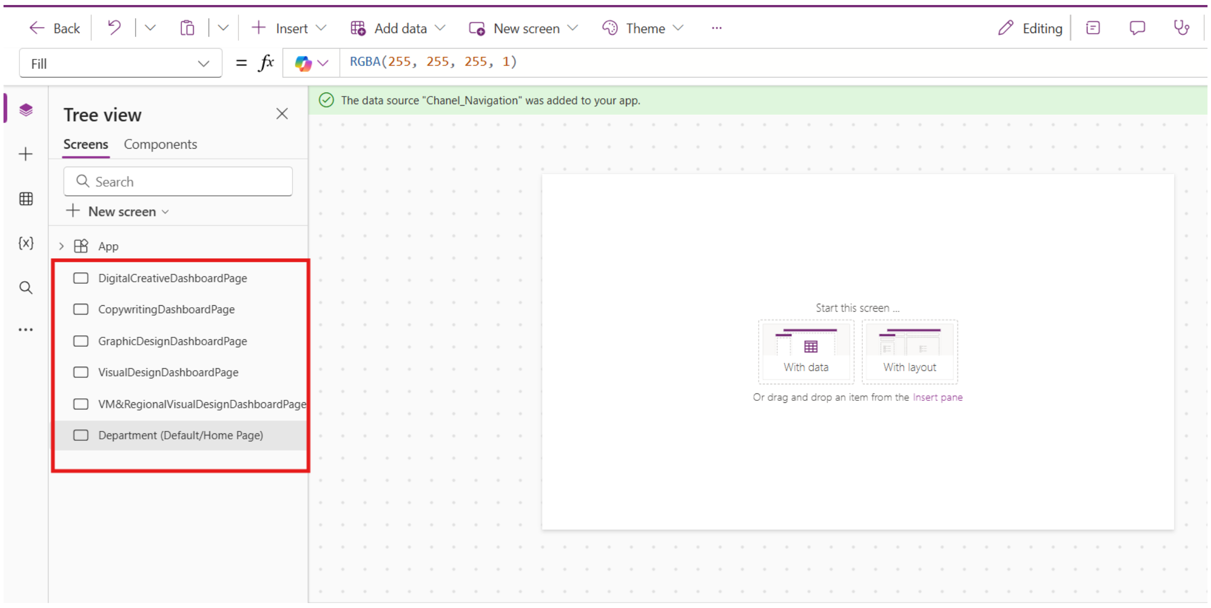
1. Create a new screen. Do this by selecting 'New Screen' in the tree view, and selecting a Blank canvas.



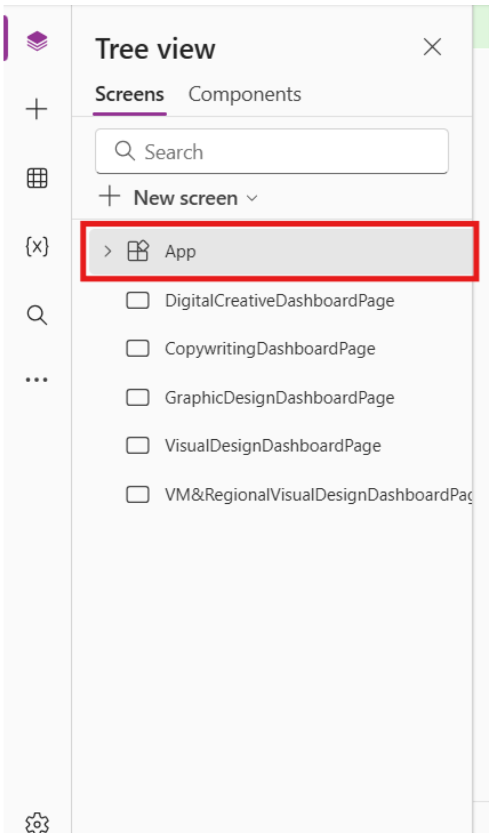
2. Right click the new screen created in the Tree view. Select ‘Rename’. Rename the screen to ‘DigitalCreativeDashboardPage’.



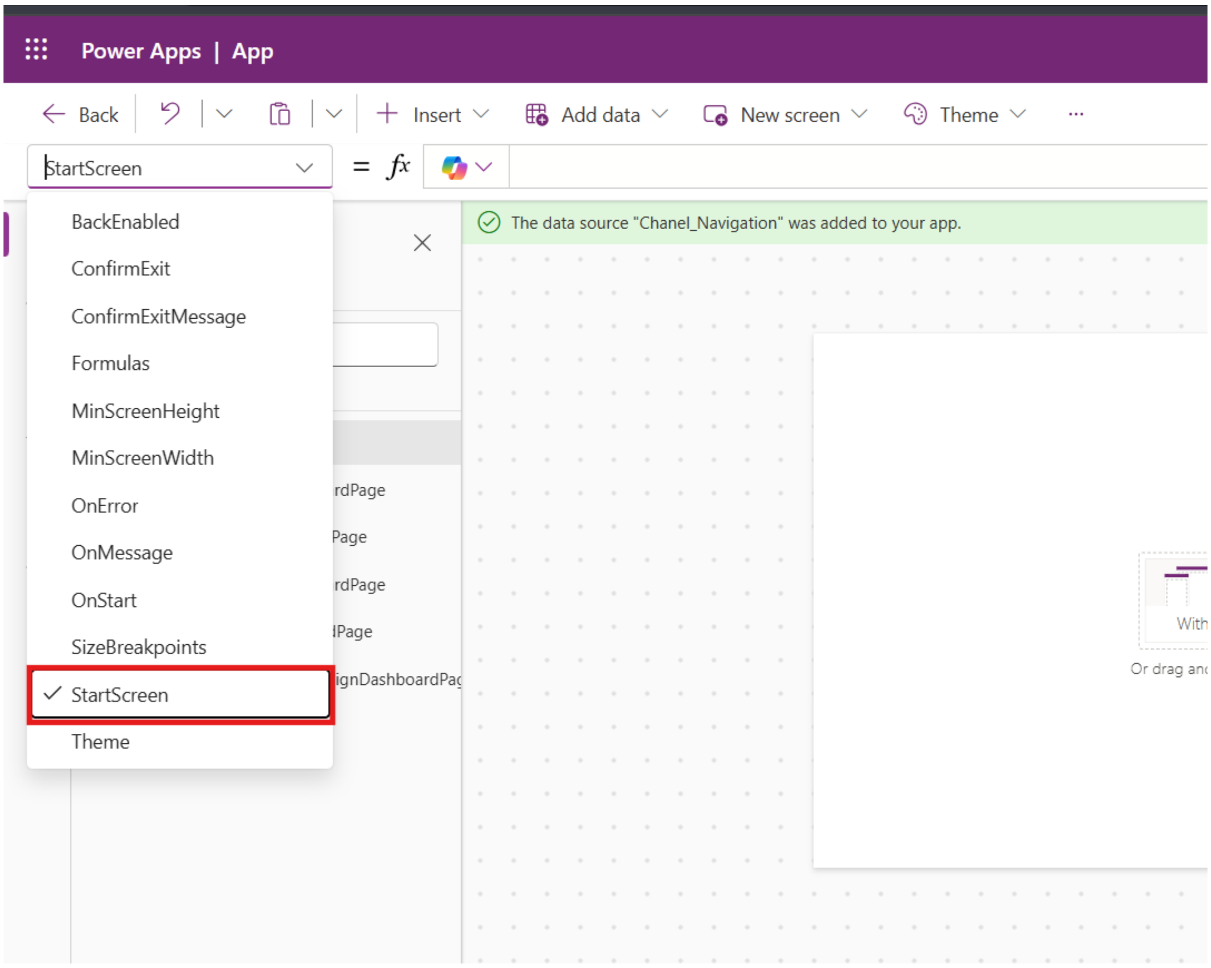
3. Repeat Steps 1 and 2 for each required team dashboard. Since there are five teams, this process should be completed for five separate screens. Rename each screen using the format ‘[TeamName]DashboardPage’ to maintain consistency. Create an additional screen named 'Department (Default/Home Page)'. The final screens within the Tree view should look something like this:


Back
Insert
Add data
New screen
Theme
Editing
Fill
RGBA(255, 255, 255, 1)
The data source "Chanel_Navigation" was added to your app.
Tree view
Screens
Components
Search
New screen
App
DigitalCreativeDashboardPage
CopywritingDashboardPage
GraphicDesignDashboardPage
VisualDesignDashboardPage
VM&RegionalVisualDesignDashboardPage
Department (Default/Home Page)
Start this screen ...
With data
With layout
Or drag and drop an item from the Insert pane

4. To add the navigation formula to Power Apps, select 'App' within the Tree view.


Tree view
Screens
Components
Search
New screen
App
DigitalCreativeDashboardPage
CopywritingDashboardPage
GraphicDesignDashboardPage
VisualDesignDashboardPage
VM&RegionalVisualDesignDashboardPag

5. Select the 'Start Screen' property in the formula bar near the top of the Power Apps.



The data source "Chanel_Navigation" was added to your app.

6. In the formula bar for the 'Start Screen' property, enter the following formula:

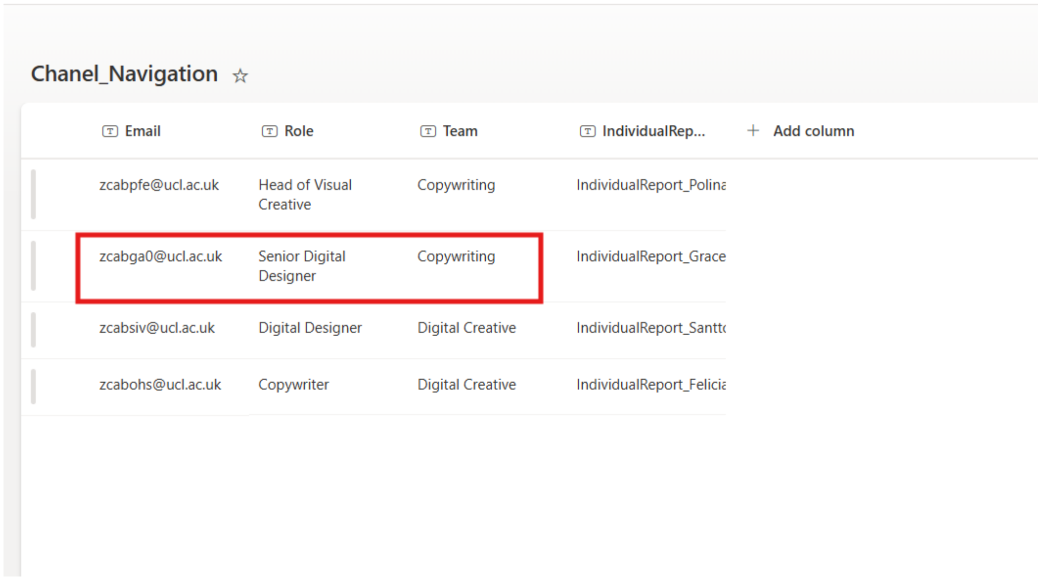
```
Switch(
LookUp('Chanel_Navigation', 'Email' = User().Email, Team),
"Digital Creative", DigitalCreativeDashboardPage,
"Copywriting", CopywritingDashboardPage,
"Graphic Design", GraphicDesignDashboardPage,
"Visual Design",VisualDesignDashboardPage,
"VM & Regional Visual Design",
'VM&RegionalVisualDesignDashboardPage',
'Department (Default/Home Page)'
)
```

Explanation:

- 'Chanel_Navigation': the name of the SharePoint list created initially. This is the table used to search for a specific value.
- 'Email' = User().Email: attributes the email to be the user's email address when logging in. This formula checks if there exists a value within the 'Email' column is equal to the user's email.

- Team: for the first record of the user's email that is found, return the Team of that record.

For example, if User().Email is equal to zcabga0@ucl.ac.uk, the value of Team returned will be equal to 'Copywriting'.



Chanel_Navigation ☆

Email	Role	Team	IndividualRep...	Add column
zcabpfe@ucl.ac.uk	Head of Visual Creative	Copywriting	IndividualReport_Polina	
zcabga0@ucl.ac.uk	Senior Digital Designer	Copywriting	IndividualReport_Grace	
zcabsiv@ucl.ac.uk	Digital Designer	Digital Creative	IndividualReport_Santt	
zcabohs@ucl.ac.uk	Copywriter	Digital Creative	IndividualReport_Felicia	

Creating Profile Page in Power Apps

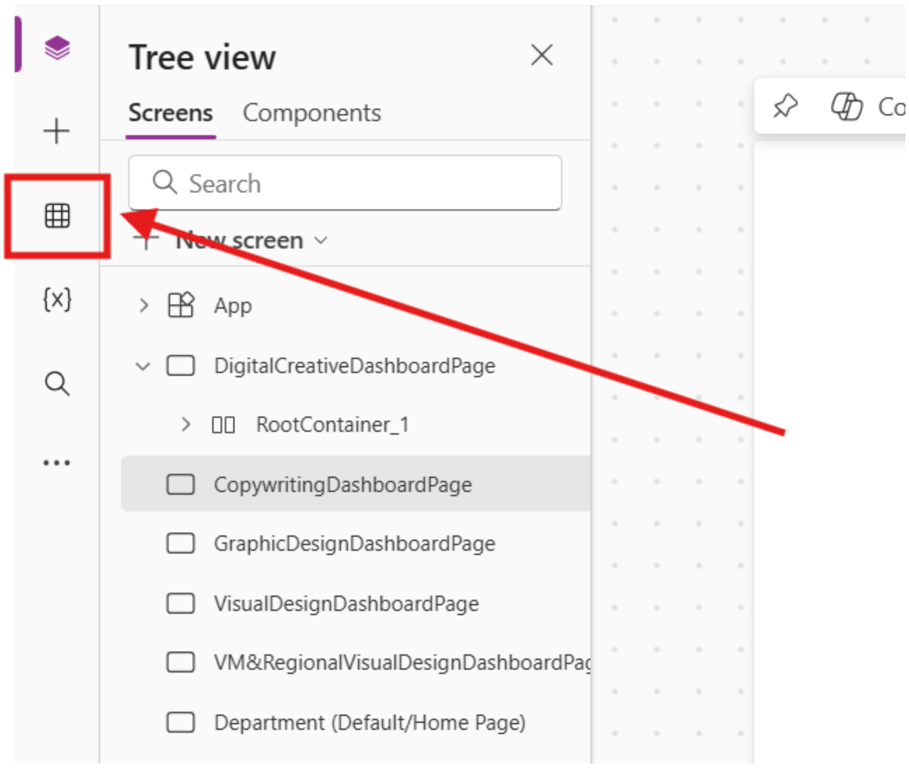
The profile page is a specific screen within the application to collate information about the user. By utilising the organisation's database that contains each user's information, only one screen is required to display accurate information about each user.

Connecting Office365Users as a Data Source

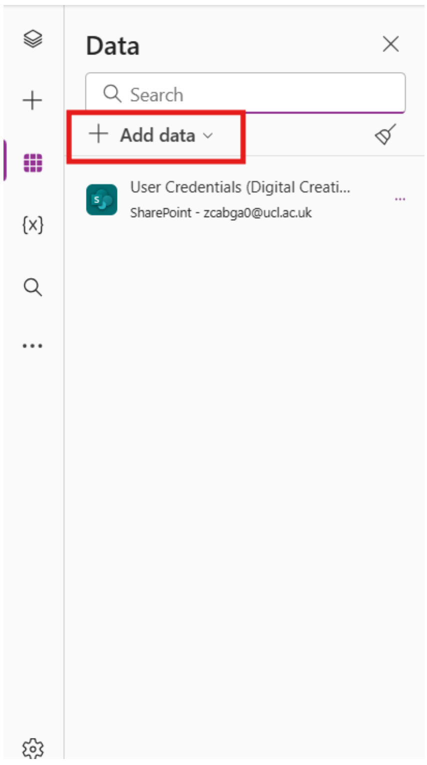
To access each user's information, Office365Users needs to be connected to the Power App as a data source.

Steps:

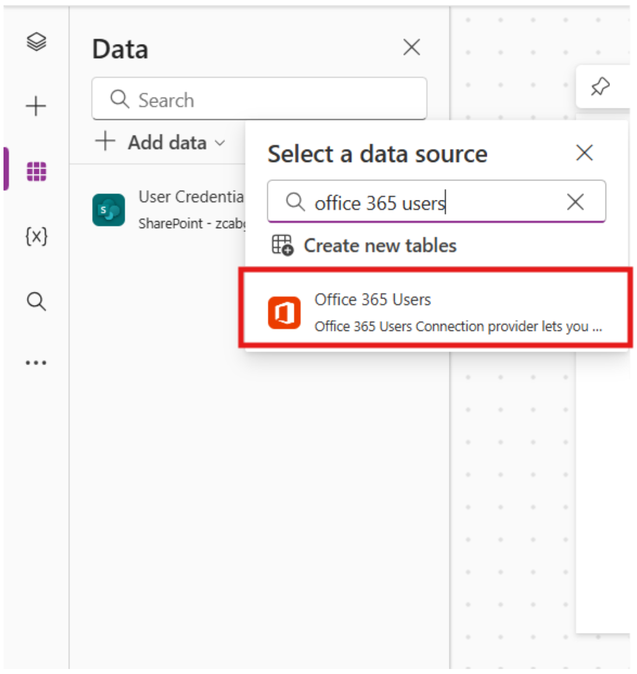
1. Select the 'Data' button located on the far-left sidebar of the Power Apps interface.


Tree view
Screens
Components
Search
New screen
App
DigitalCreativeDashboardPage
RootContainer_1
CopywritingDashboardPage
GraphicDesignDashboardPage
VisualDesignDashboardPage
VM&RegionalVisualDesignDashboardPag
Department (Default/Home Page)

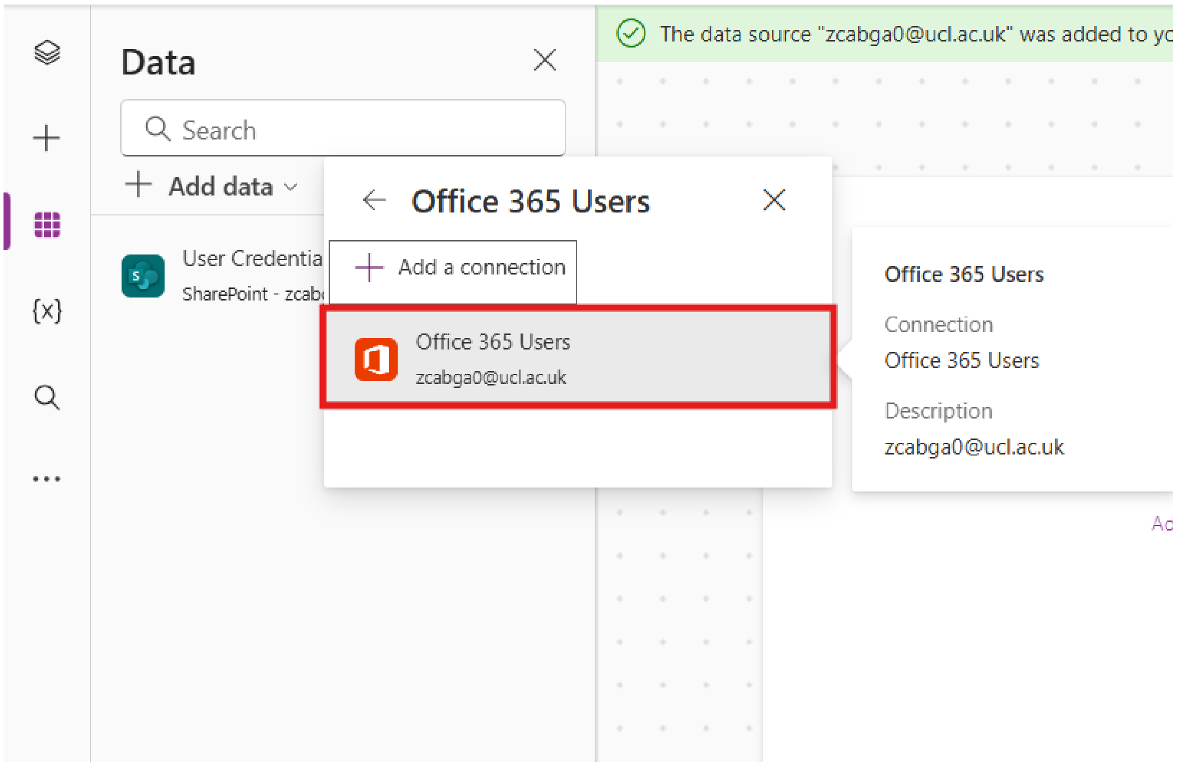
2. When the ‘Data’ page is expanded, select the ‘Add data’ button at the top.


Data
Search
Add data
User Credentials (Digital Creati...
SharePoint - zcabga0@ucl.ac.uk

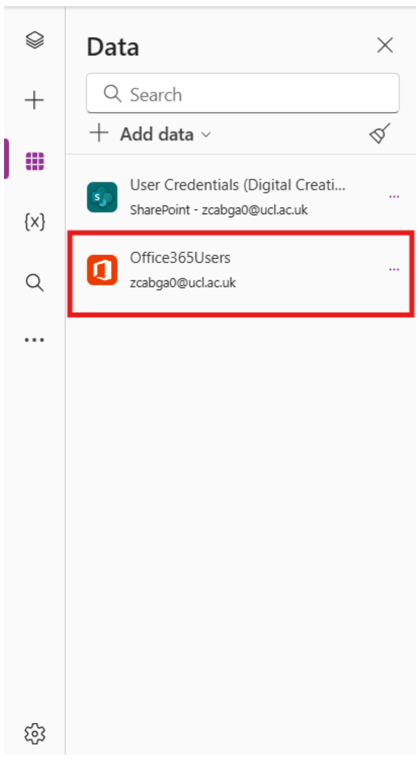
3. Search for Office365Users in the search bar. Select ‘Office 365 Users’.



4. Select Office 365 Users as a data source.



5. Upon selection, Office365Users should be successfully added to the Power App as a data source.

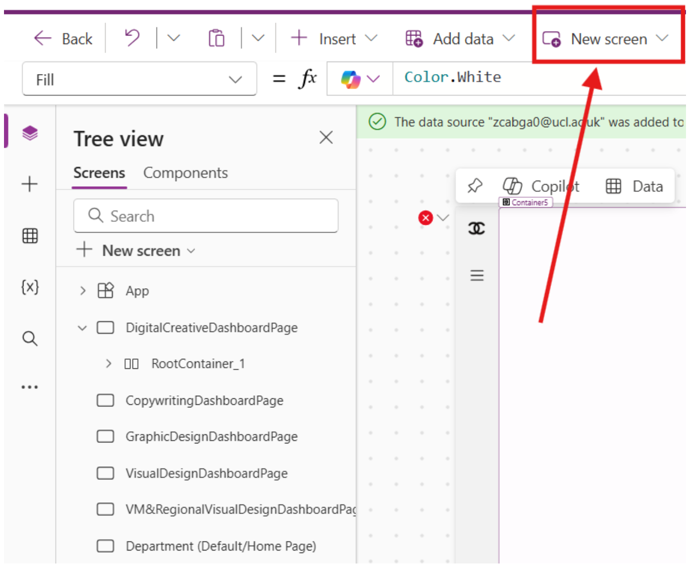

Data
Search
Add data
User Credentials (Digital Creati...
SharePoint - zcabga0@ucl.ac.uk
Office365Users
zcabga0@ucl.ac.uk

Creating a New Profile Page Screen

Users will be automatically directed to this screen upon selecting the ‘Profile’ button from the left-hand navigation menu. Thus, a new screen will need to be created. This section will detail all aspects of the Profile Page, from UI design to how information will be retrieved and displayed.

<u>Steps:</u>

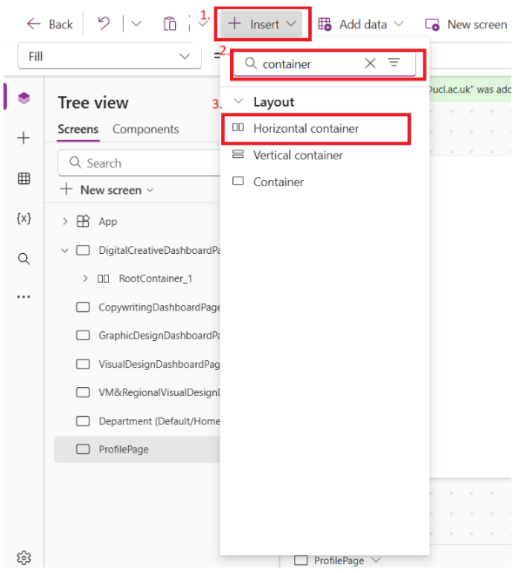
1. Create a new Profile screen. Select the ‘New screen’ button in the tool bar at the top of the Power App. Rename the screen to ‘ProfilePage’.


Back
Insert
Add data
New screen
Fill
Tree view
Screens
Components
Search
New screen
App
DigitalCreativeDashboardPage
RootContainer_1
CopywritingDashboardPage
GraphicDesignDashboardPage
VisualDesignDashboardPage
VM&RegionalVisualDesignDashboardPag
Department (Default/Home Page)
Copilot
Data

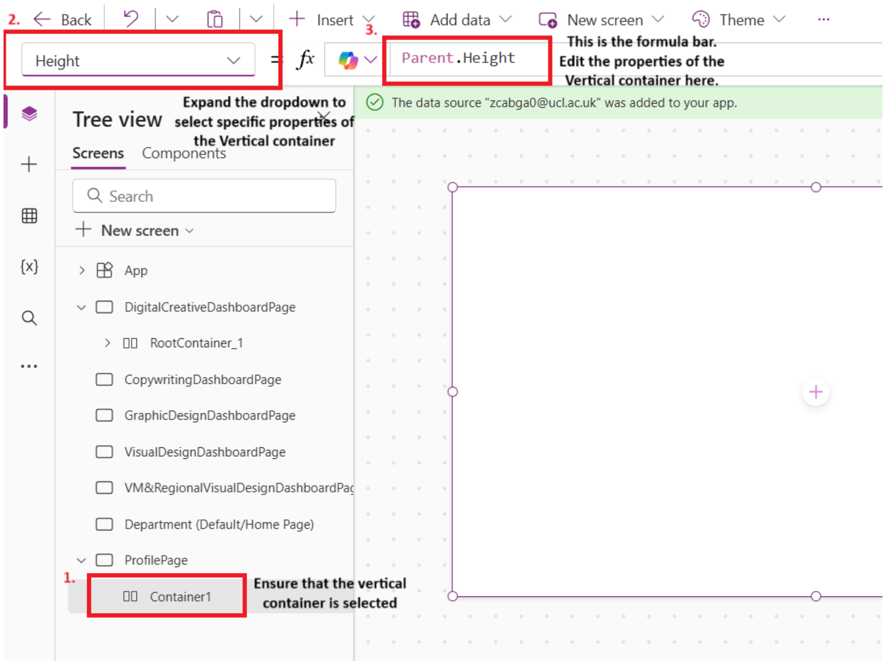
```
= fx Color.White
```

The data source "zcabga0@ucl.ac.uk" was added to

2. Add a horizontal container within the ProfilePage screen. Select the ‘Insert’ button in the tool bar at the top of the Power App. Search for ‘Container’, and select ‘Horizontal container’. This will serve as the top-level container for all other controls and elements within the ProfilePage screen, enabling easier design and management for the rest of the nested containers within the screen.



3. After adding the horizontal container to the ProfilePage screen, edit the following properties of the container. These properties can be set within the formula bar of the Power App.



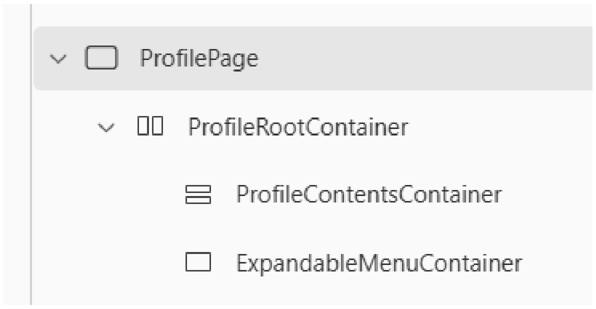
2.
Back
Insert
3.
Add data
New screen
Theme
Height
Parent.Height
This is the formula bar. Edit the properties of the Vertical container here.
Expand the dropdown to select specific properties of the Vertical container
The data source "zcabga0@ucl.ac.uk" was added to your app.
Tree view
Screens
Components
Search
New screen
App
DigitalCreativeDashboardPage
RootContainer_1
CopywritingDashboardPage
GraphicDesignDashboardPage
VisualDesignDashboardPage
VM&RegionalVisualDesignDashboardPag
Department (Default/Home Page)
ProfilePage
1.
Container1
Ensure that the vertical container is selected

Edit the following properties of the Horizontal container:

- Set Height to Parent.Height within the formula bar.

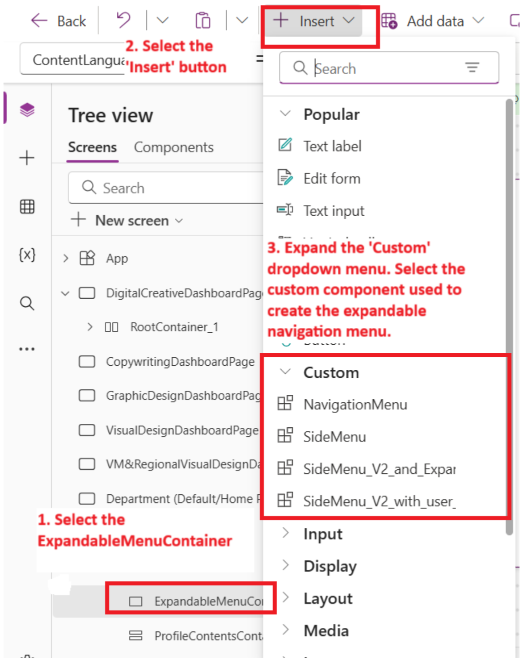
- Set Width to Parent.Width within the formula bar.
- Set DropShadow to DropShadow.None within the formula bar.
- Set X to 0.
- Set Y to 0.

4. Repeat Step 2 to nest one Container and one Vertical container inside the root container created in Step 3. After creation, rename the Vertical container as 'ProfileContentsContainer' and the Container as 'ExpandableMenuContainer'. The nested tree view of the ProfilePage should now look like this:



ProfileContentsContainer and **ExpandableMenuContainer** is nested within **ProfileRootContainer**.

5. The custom component used to design and create the navigation menu will now be added to the ExpandableMenuContainer. Ensure that the container is selected before inserting the custom component.

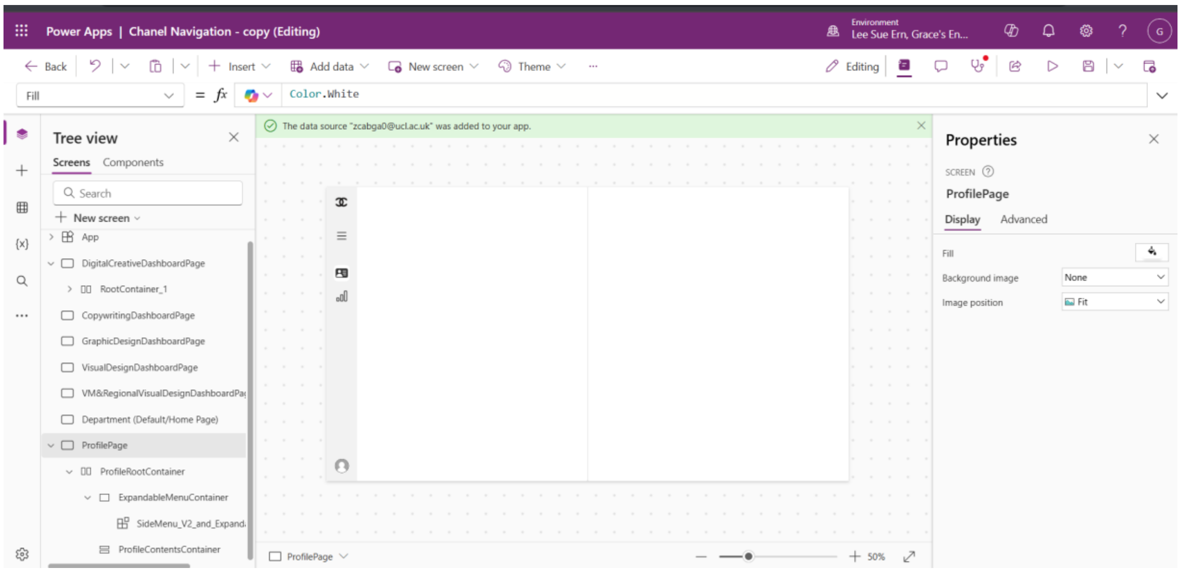


1. Select the ExpandableMenuContainer

2. Select the 'Insert' button

3. Expand the 'Custom' dropdown menu. Select the custom component used to create the expandable navigation menu.

The ProfilePage screen should now appear as follows. Please note that the navigation menu may differ from the final version.


Fill = Color.White

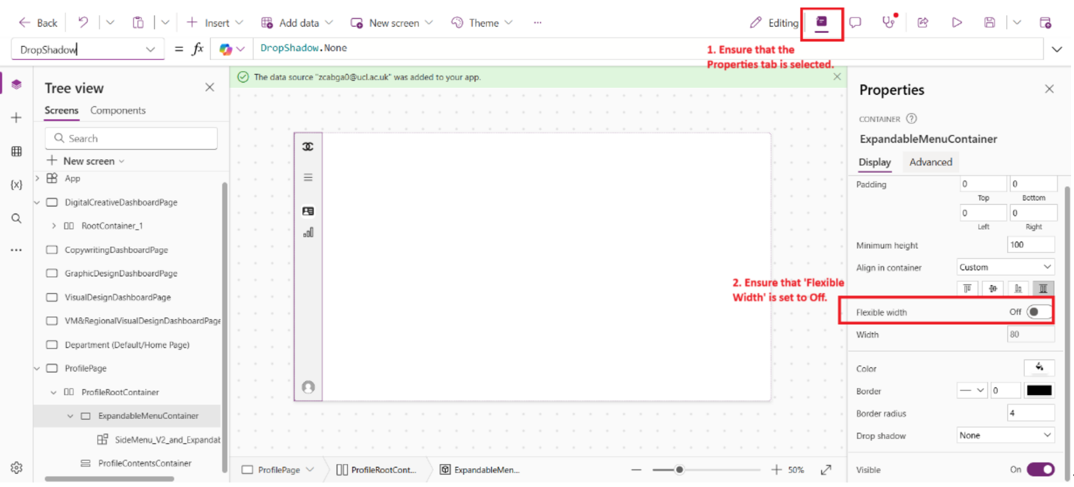
The data source "zcabga0@ucl.ac.uk" was added to your app.

Properties — SCREEN — ProfilePage

Background image	None
Image position	Fit

6. Utilising the method in Step 3, edit the following properties of the ExpandableMenuContainer:
    - Set DropShadow to DropShadow.None within the formula bar.
    - Set Width to [‘NavigationBarComponentName’].Width within the formula bar. In this example, Width will be set to SideMenu_V2_and_Expandable_2.Width.

In the ‘Properties’ Menu to the right, ensure that Flexible width is toggled off.


DropShadow = DropShadow.None

1. Ensure that the Properties tab is selected.

2. Ensure that 'Flexible Width' is set to Off.

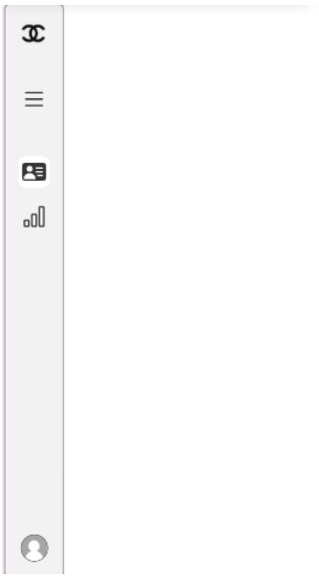
The data source "zcabga0@ucl.ac.uk" was added to your app.

Properties — CONTAINER — ExpandableMenuContainer

Padding	Top 0, Bottom 0, Left 0, Right 0
Minimum height	100
Align in container	Custom
Flexible width	Off
Width	80
Border	0
Border radius	4
Drop shadow	None
Visible	On

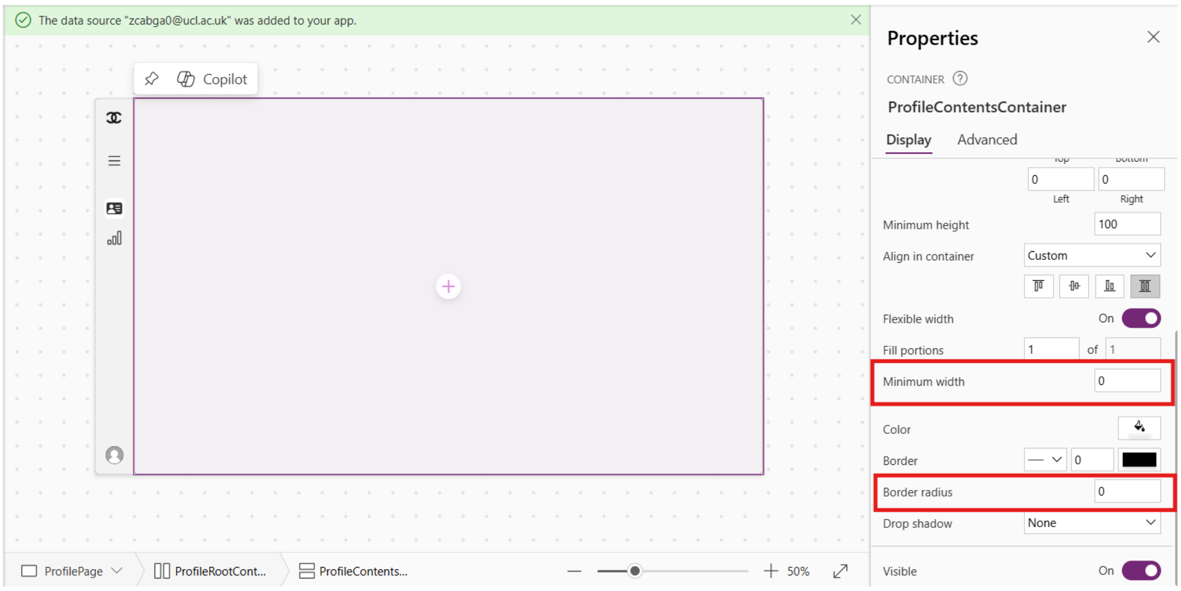
#

Your ExpandableMenuContainer should now look like this:



7. Utilising the method as highlighted in Step 3, edit the following properties of the ProfileContentsContainer:
    - Set DropShadow to DropShadow.None within the formula bar.
    - Set Fill to RGBA(242, 242, 242, 1). This sets the background colour to a light grey.
    - Set LayoutDirection to LayoutDirection.Vertical

    In the 'Properties' Menu to the right, ensure that 'Border Radius' and 'Minimum Width' is set to 0.



The data source "zcabga0@ucl.ac.uk" was added to your app.

Properties

CONTAINER

ProfileContentsContainer

Display | Advanced

Left 0 | Right 0

Minimum height 100

Align in container Custom

Flexible width On

Fill portions 1 of 1

Minimum width 0

Color

Border 0

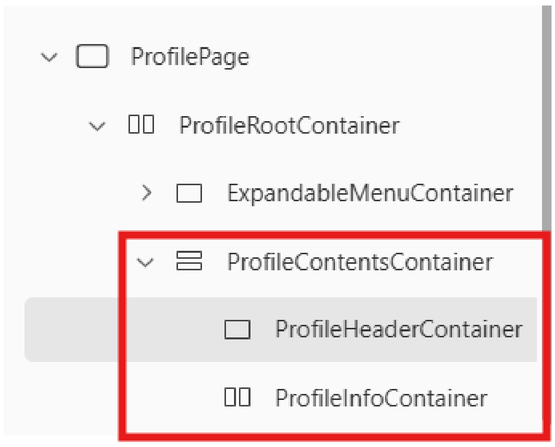
Border radius 0

Drop shadow None

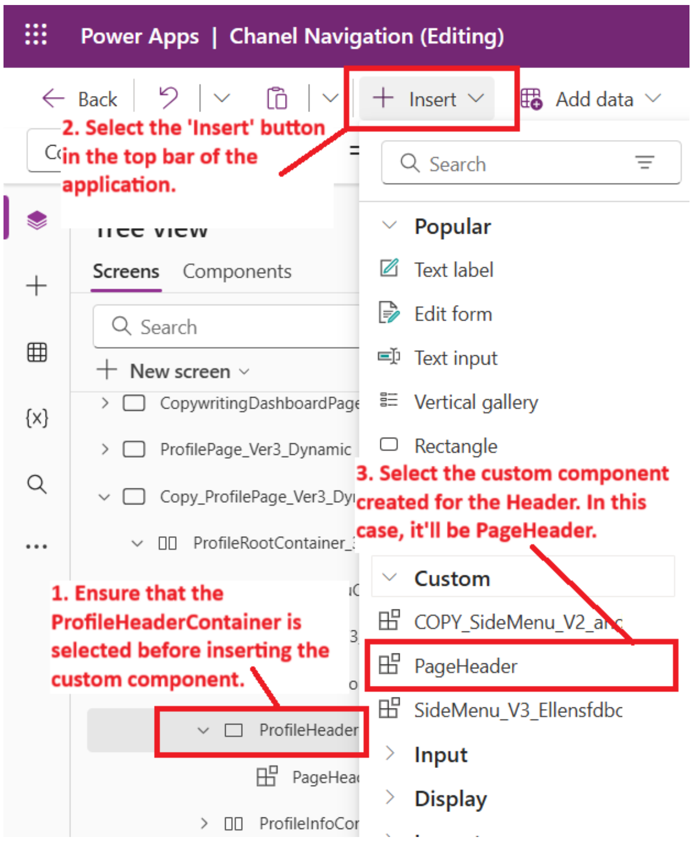
Visible On

8. Repeat Step 2 to nest one Container and one Horizontal container inside ProfileContentsContainer created in Step 4. After creation, rename the Horizontal container to 'ProfileInfoContainer' and the Container as

'ProfileHeaderContainer'. The nested tree view of the ProfilePage should now look like this:



9. The 'ProfileHeaderContainer' will be used to display the title of the screen. The custom component used to design and create the Header will now be added to this container. Ensure that the container is selected before inserting the custom component.

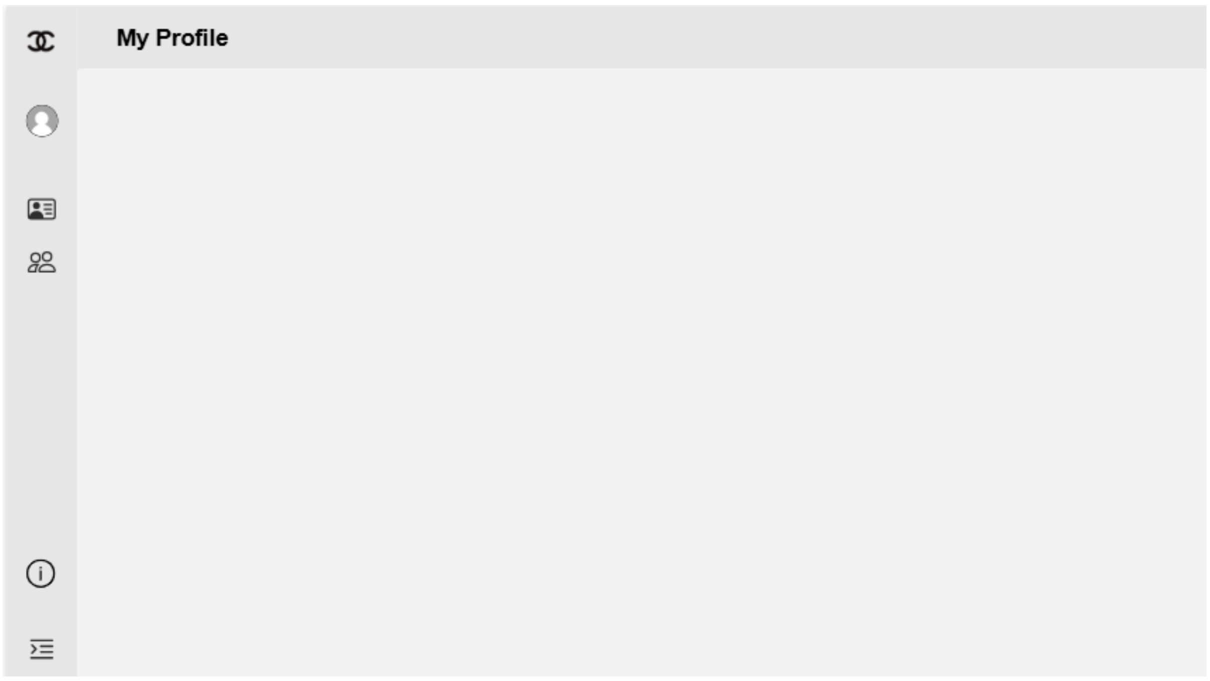


1. Ensure that the ProfileHeaderContainer is selected before inserting the custom component.

2. Select the 'Insert' button in the top bar of the application.

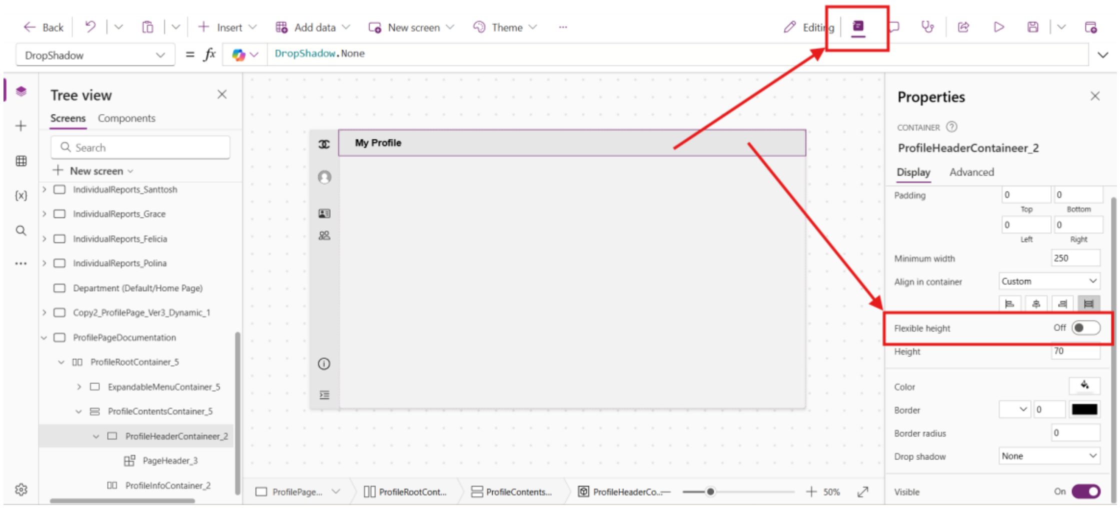
3. Select the custom component created for the Header. In this case, it'll be PageHeader.

The ProfilePage should now display the header and the title of the screen, as follows:



10. Utilising the method in Step 3, edit the following properties of the ProfileHeaderContainer:
    - Set DropShadow to DropShadow.None within the formula bar.
    - Set BorderStyle to BorderStyle.None.
    - Set Height to 70.

    In the 'Properties' Menu to the right, ensure that Flexible width is toggled off.

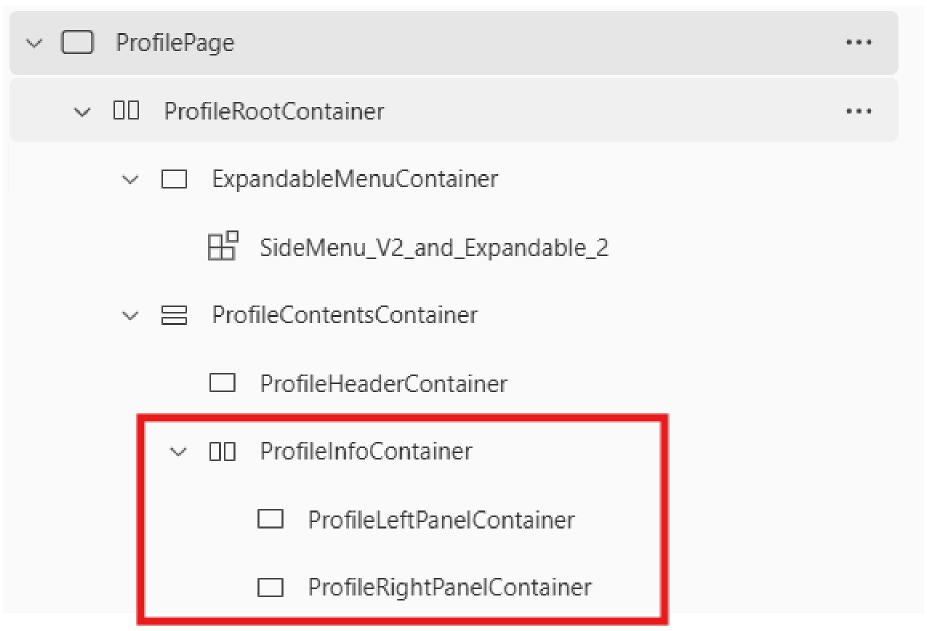

DropShadow = fx DropShadow.None

Properties
CONTAINER
ProfileHeaderContaineer_2
Display | Advanced

Padding	Top 0, Bottom 0, Left 0, Right 0
Minimum width	250
Align in container	Custom
Flexible height	Off
Height	70
Border	0
Border radius	0
Drop shadow	None
Visible	On

11. Repeat Step 2 to nest two Containers inside the ProfileInfoContainer created in Step 8. After creation, rename the containers as ProfileLeftPanelContainer and ProfileRightPanelContainer respectively. These containers will be used to store information about the user in a neater, more organised way.

Your nested containers should look like this:

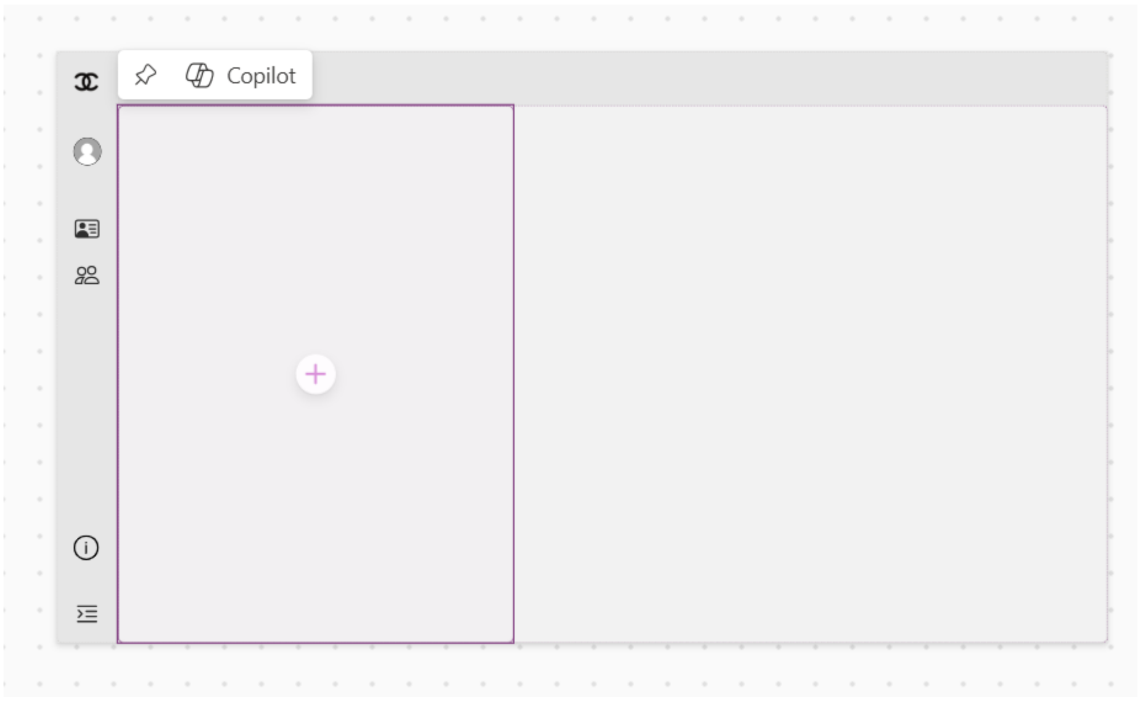


12. Using the method described in Step 3, edit the following properties of ProfileLeftPanelContainer to have these values:
    - Set Width to Parent.Width * 40%. In the 'Properties' Menu to the right, ensure that Flexible width is toggled off.
    - Set BorderStyle to BorderStyle.None.
    - Set DropShadow to DropShadow.None.
    - Set Fill to RGBA(237, 237, 237, 1)

    Edit the following properties of ProfileRightPanelContainer to have the following values:

    - Set BorderStyle to BorderStyle.None.
    - Set DropShadow to DropShadow.None.
    - Set Fill to RGBA(237, 237, 237, 1)

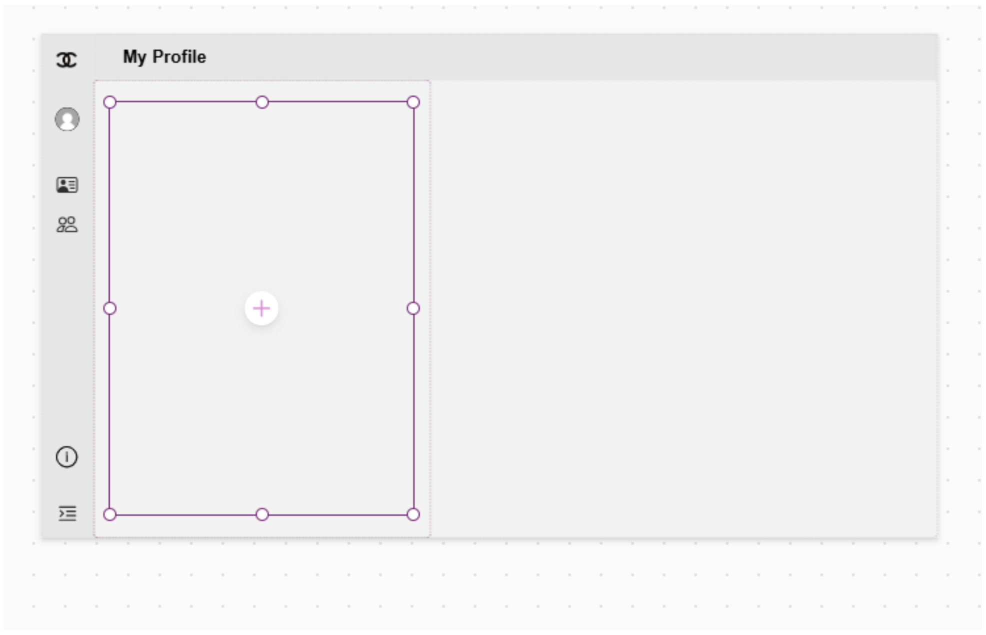
Your ProfilePage screen should now look like this, with the left container being smaller in width than the right container:



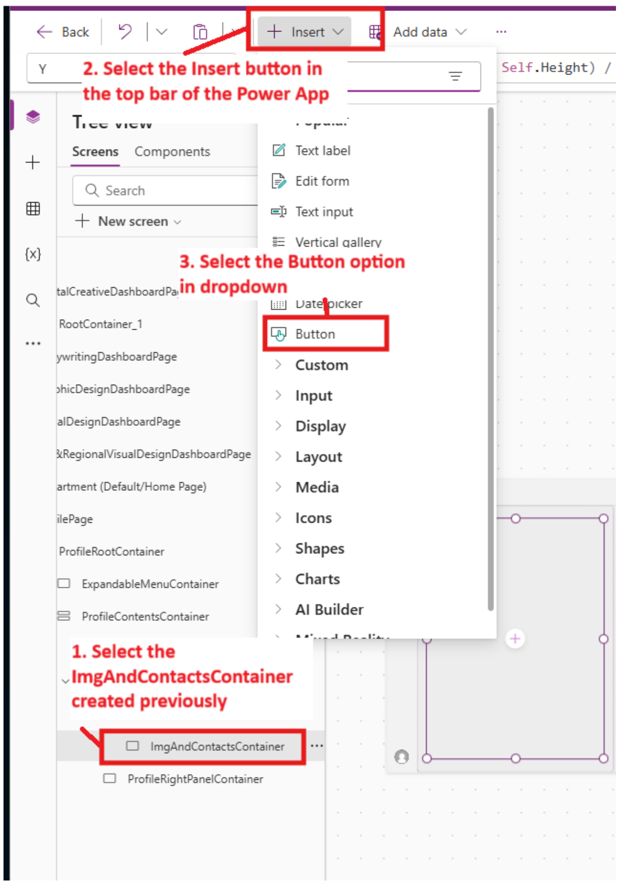
We will now start designing ProfileLeftPanelContainer.

13. Nest a normal Container within ProfileLeftPanelContainer. This container will act as the parent container that manages UI of the nested child containers. Rename this container to ImgAndContactsContainer. Set the properties of ImgAndContactsContainer to the following:
    - Set BorderStyle to BorderStyle.None.
    - Set DropShadow to DropShadow.None.
    - Set Height to 90% * Parent.Height.
    - Set Width to 90% * Parent.Width.
    - Set X to (Parent.Width - Self.Width)/2
    - Set Y to (Parent.Height - Self.Height) / 2

    ImgAndContactsContainer should look something like this:

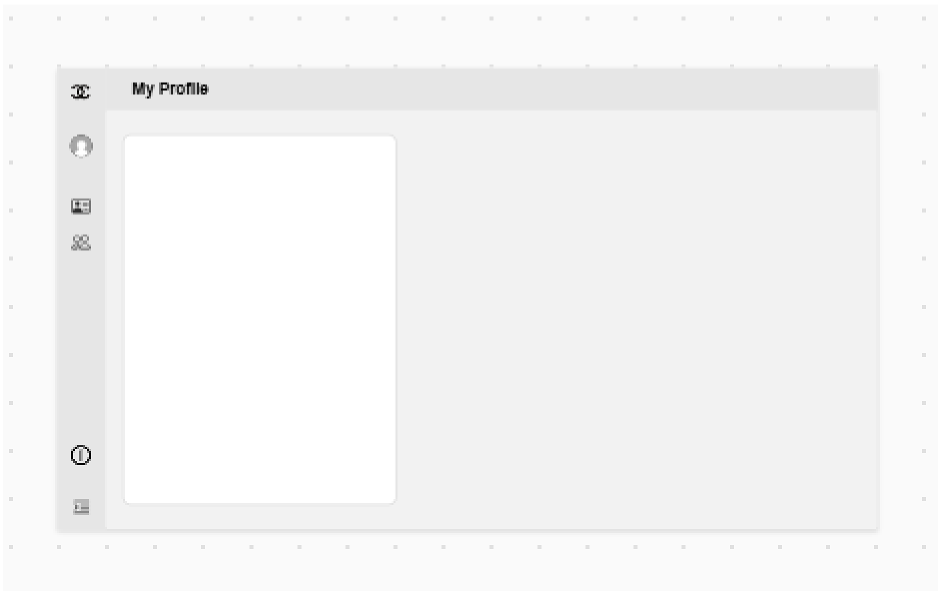


14. After selecting ImgAndContactsContainer, insert a button into the container. Rename the button to ProfileLeftPanelBackground. Ensure that the button is nested within the container.



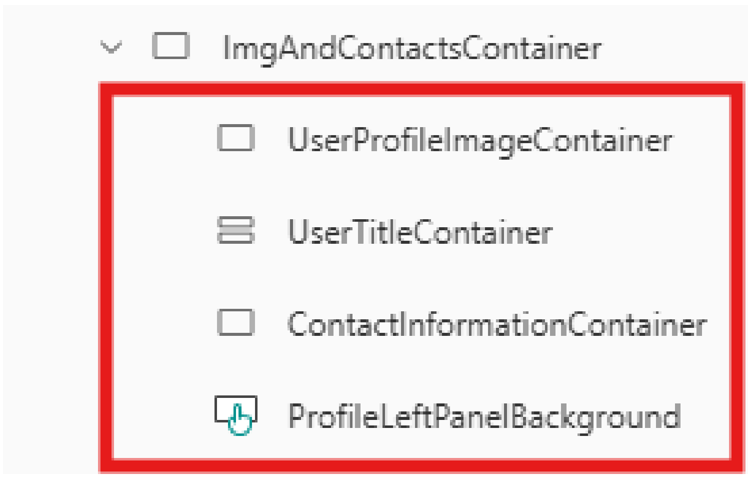
15. This button will be used purely for design purposes. Edit the following properties of ProfileLeftPanelBackground to design and disable all functionalities of the button:
    - Set BorderStyle to BorderStyle.Solid.
    - Set Fill to RGBA(255, 255, 255, 1).
    - Set Height to Parent.Height * 98%.
    - Set Width to Parent.Width * 98%.
    - Set RadiusBottomLeft, RadiusBottomRight, RadiusTopLeft and RadiusTopRight to 14.
    - Set X to (Parent.Width - Self.Width)/2
    - Set Y to (Parent.Height - Self.Height)/2
    - Make sure HoverColor, HoverFill and Text is set to nothing/have no value.

    The ProfilePage screen should now look something like this:



16. Insert two Containers and one Vertical Container as child elements within the ImgAndContactsContainer created in Step 13. These containers will store the user's primary details, including their name, role, profile picture and contact information on the profile page. Name the two Containers as UserProfileImageContainer and ContactInformationContainer respectively. Name the Vertical Container as UserTitleContainer.

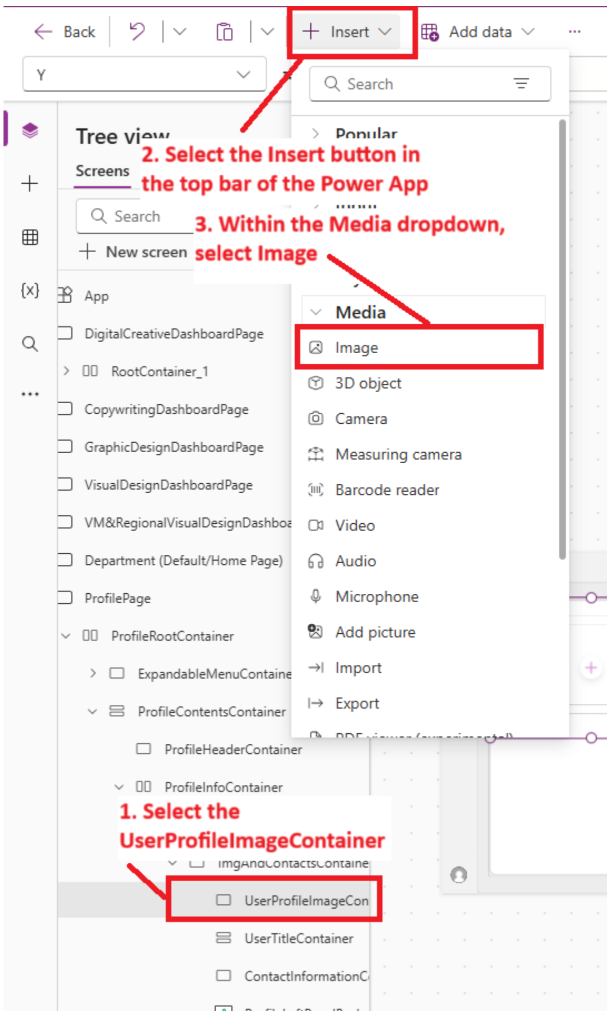
    After inserting and nesting the three containers, the Tree view of ImgAndContactsContainer should look like this:


ImgAndContactsContainer
UserProfileImageContainer
UserTitleContainer
ContactInformationContainer
ProfileLeftPanelBackground

Ensure that ProfileLeftPanelBackground is reordered towards the back of the container.

17. Select UserProfileImageContainer and set it's properties to the following:
    - Set BorderColor to RGBA(0, 0, 0, 0).
    - Set BorderStyle to BorderStyle.None.
    - Set DropShadow to DropShadow.None.
    - Set Height to Parent.Height * 50%.
    - Set Width to ['WhiteButtonBackground'].Width. In this case, it will be ProfileLeftPanelBackground.Width.
    - Set X to ['WhiteButtonBackground'].X. In this case, it will be ProfileLeftPanelBackground.X.
    - Set Y to 0.

18. The user's profile picture will now be inserted into UserProfileImageContainer. This can be done by inserting an image as a child within the UserProfileImageContainer. Rename this image to UserProfileImage



19. Select UserProfileImage and set it’s properties to the following:

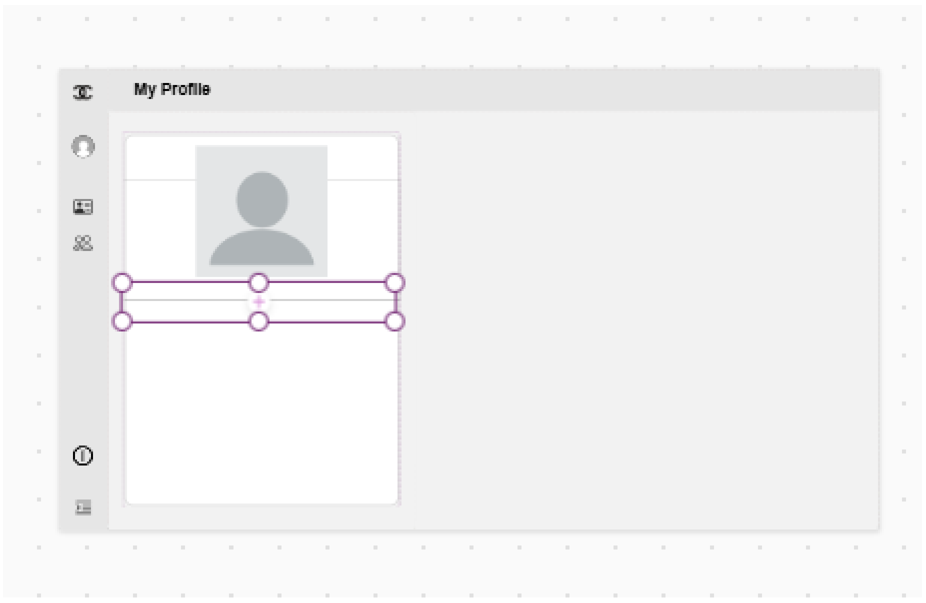
- Set Height to 70% * Parent.Height.
- Set Width to Self.Height
- Set X to Parent.Width/2 - Self.Width/2.
- Set Y to 50% * (Parent.Height/2 - Self.Height/2)
- Set ImagePosition to ImagePosition.Fit.
- Set Image to the following formula:

```
If(
    Office365Users.UserPhotoMetadata(User().Email).HasPhoto,
    Office365Users.UserPhotoV2(User().Email),
    'blank-profile-picture' // Replace with your uploaded image name
)
```

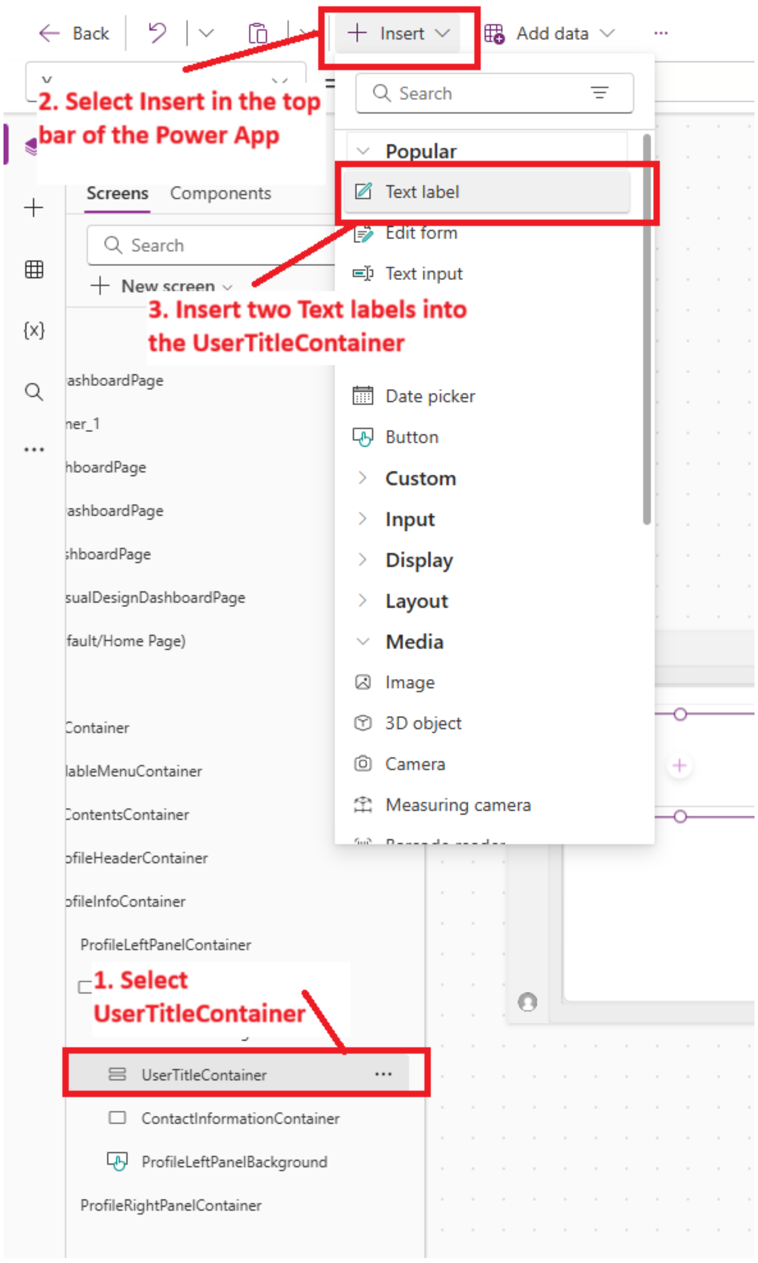
This formula is used to dynamically display the user’s profile picture from the Office 365 Users data source. It ensures a profile picture is always displayed, and that if an Office 365 profile picture exists, it is shown. If no profile picture exists, a default placeholder image (blank-profile-picture) is displayed instead.

20. Select UserTitleContainer created in Step 16 and set it's properties to the following:
    - Set BorderStyle to BorderStyle.None.
    - Set DropShadow to DropShadow.None.
    - Set Height to Parent.Height * 10%
    - Set Width to ['WhiteButtonBackground'].Width. In this case, it will be ProfileLeftPanelBackground.Width.
    - Set LayoutJustifyContent to LayoutJustifyContent.Start
    - Set Y to UserProfileImage * 115%.

By now, your ProfilePage screen should look like this:



21. The UserTitleContainer will now be used to display the name and role of the user. Insert two labels and nest it within the UserTitleContainer. Name the two labels as FullNameLabel and JobTitleLabel respectively.


Back
Insert
Add data
2. Select Insert in the top bar of the Power App
Search
Popular
Text label
Edit form
Text input
3. Insert two Text labels into the UserTitleContainer
Date picker
Button
Custom
Input
Display
Layout
Media
Image
3D object
Camera
Measuring camera
Screens
Components
New screen
1. Select UserTitleContainer
UserTitleContainer
ContactInformationContainer
ProfileLeftPanelBackground
ProfileLeftPanelContainer
ProfileRightPanelContainer

22. Select the FullNameLabel text label created previously and set it's properties to the following:

- Set Align to Align.Center.
- Set DisplayMode to DisplayMode.View.
- Set Font to Font.Arial.
- Set Height to Parent.Height * 50%.
- Set Width to Parent.Width.
- Set Y to 0
- Set Text to the following formula:

```
Office365Users.MyProfileV2().givenName & " " &
    Office365Users.MyProfileV2().surname
```

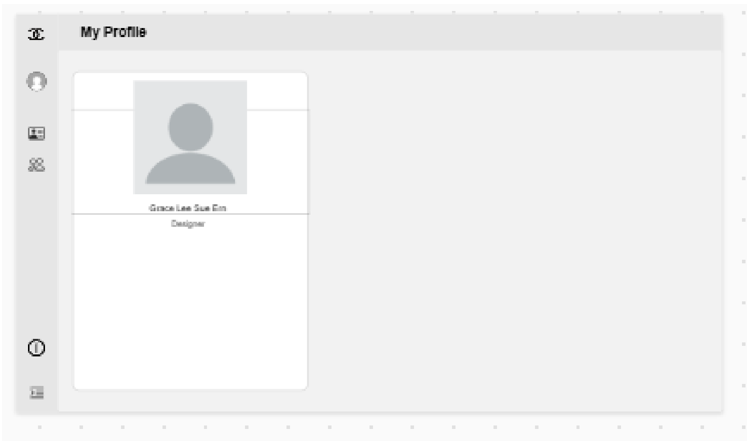
This formula is used to display the logged-in user's name. This is done by concatenating the user's first name (Office365Users.MyProfileV2().givenName) with the user's last name (Office365Users.MyProfileV2().surname), as found within the Office365User database.

23. Select the JobTitleLabel text label created previously and set it's properties to the following:
    - Set Align to Align.Center.
    - Set Color to RGBA(77, 77, 77, 1).
    - Set DisabledBorderColor to RGBA(116, 116, 116, 1).
    - Set DisplayMode to DisplayMode.View.
    - Set Font to Font.Arial.
    - Set Height to Parent.Height - FullNameLabel.Height.
    - Set Width to Parent.Width.
    - Set Size to 12.
    - Set Text to the following formula:

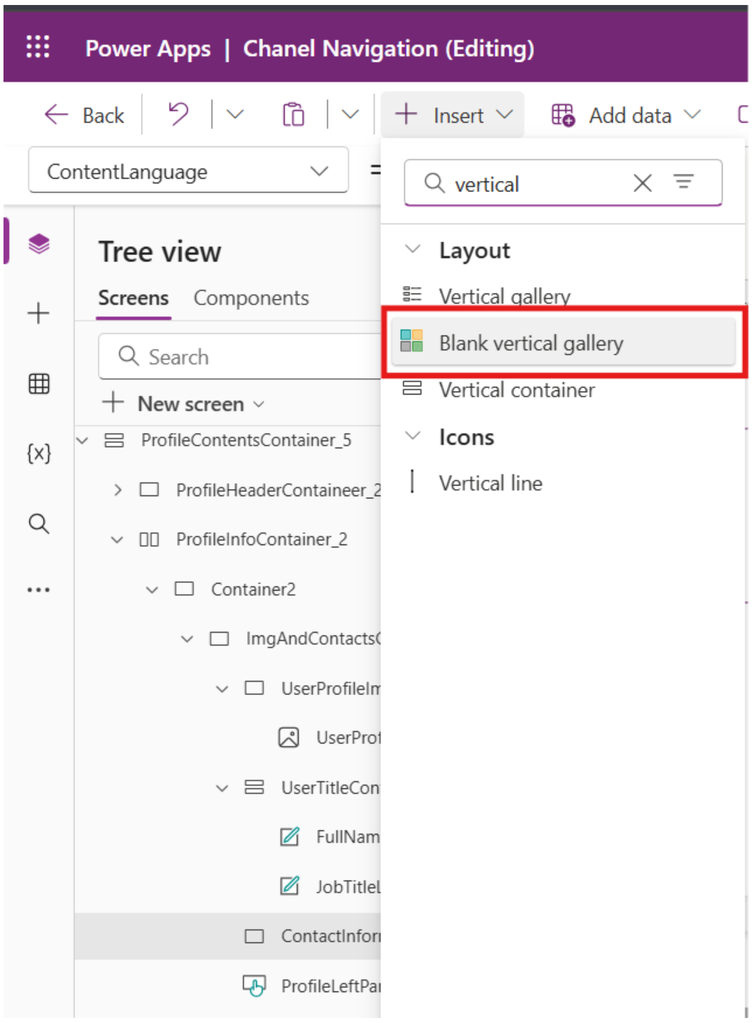
```
If (
        IsBlank(Office365Users.MyProfileV2().jobTitle),
        "Designer", // This is default value
        Office365Users.MyProfileV2().jobTitle
)
```

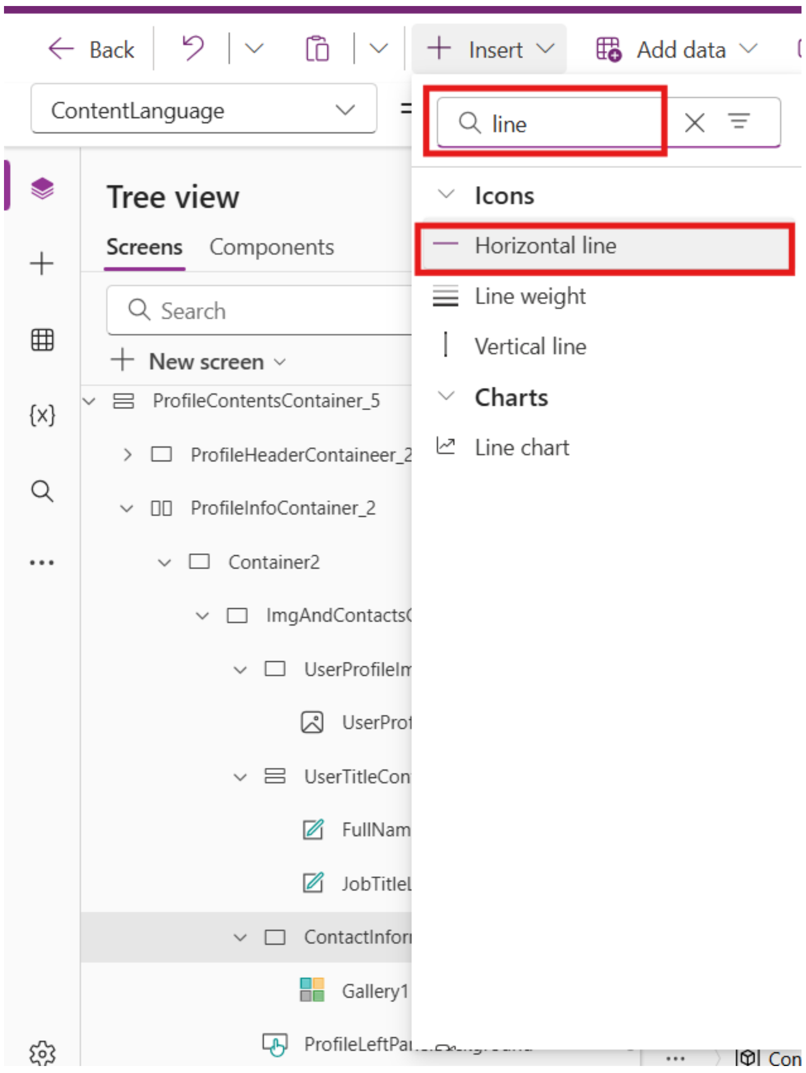
This formula is used to display the logged-in user's role within the organisation. This is done by displaying the user's job title as saved within the Office365Users database.

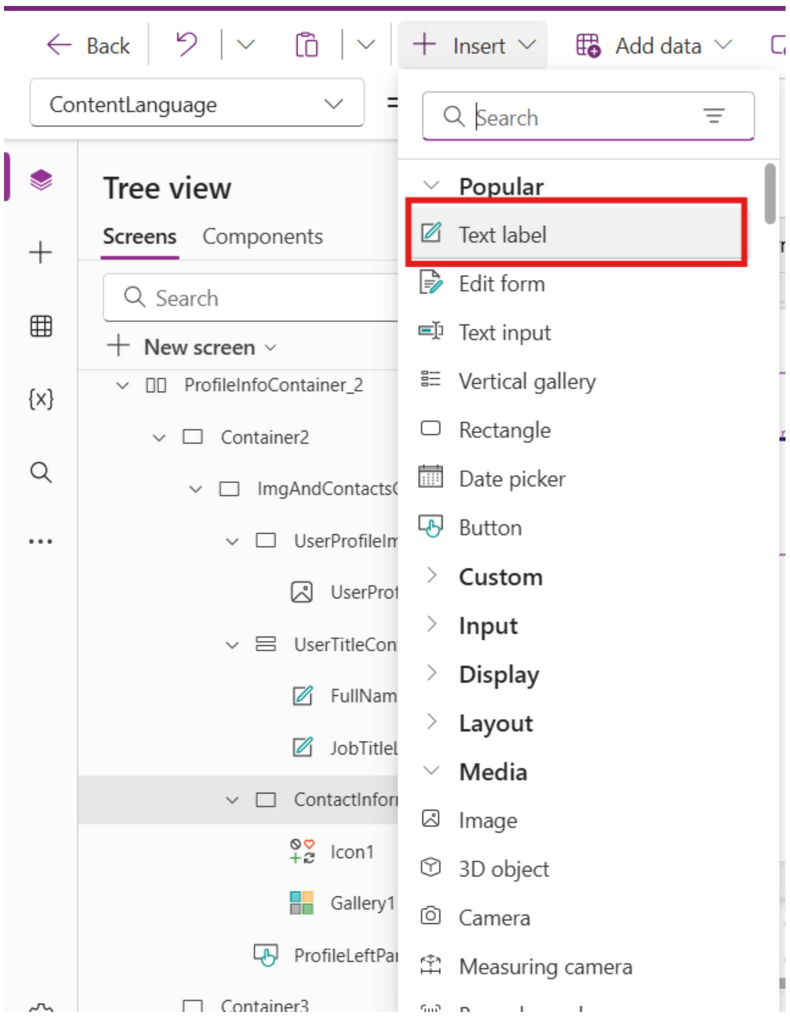
By now, your ProfilePage screen should look something like this:



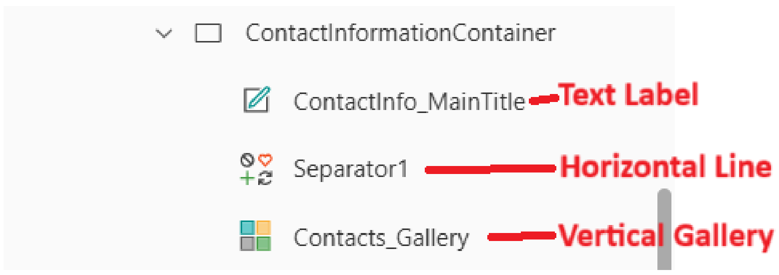
24. Set the properties of the ContactInformationContainer as the following:
    - Set BorderStyle to BorderStyle.None.
    - Set DropShadow to DropShadow.None.
    - Set Height to (ProfileLeftPanelBackground.Y + ProfileLeftPanelBackground.Height - Self.Y).
    - Set Width to ProfileLeftPanelBackground.Width.
    - Set X to ProfileLeftPanelBackground.X.
    - Set Y to 110% * (UserTitleContainer.Y + UserTitleContainer.Height)
25. The contact details of the logged-in user will be stored in the ContactInformationContainer. Insert a Text Label (renamed to ContactInfo_MainTitle), a Blank Vertical Gallery (renamed to Contacts_Gallery) and a horizontal line (named as Separator1) as children within the container.




Back
Insert
Add data
ContentLanguage
line
Icons
Horizontal line
Line weight
Vertical line
Charts
Line chart
Tree view
Screens
Components
Search
New screen
ProfileContentsContainer_5
ProfileHeaderContaineer_2
ProfileInfoContainer_2
Container2
ImgAndContacts
UserProfileIm
UserProf
UserTitleCon
FullNam
JobTitleL
ContactInfor
Gallery1
ProfileLeftPa
Con



The nested tree view should look something like this:



26. Set the properties of Separator1 to the following:
    - Set BorderColor to RGBA(202, 202, 202, 1).
    - Set BorderThickness to 1.
    - Set Color to RGBA(237, 237, 237, 1).
    - Set Fill to RGBA(237, 237, 237, 1).

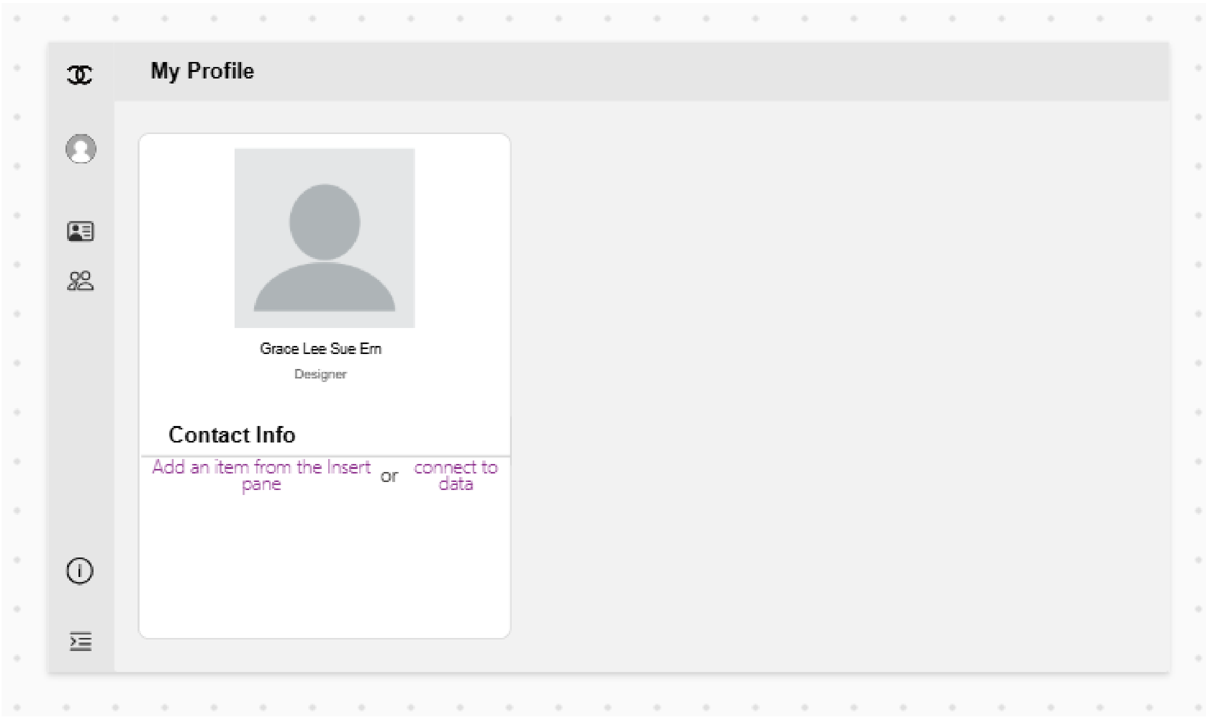
- Set Width to ProfileLeftPanelBackground.Width.
- Set Height to 0.
- Set X to ProfileLeftPanelBackground.X.
- Set Y to 15% * Parent.Y

27. Set the properties of ContactInfo_MainTitle as the following:
    - Set Font as Font.Arial.
    - Set FontWeight as FontWeight.Semibold.
    - Set PaddingTop, PaddingBottom, PaddingRight and PaddingLeft to 0.
    - Set Text to “Contact Info”.
    - Set Width to Parent.Width - Self.X.
    - Set X to ContactsGallery.X.
    - Set Y to (Separator1.Y)/2 - Self.Height/2.
    - Set size to the following formula:

```
If(
    App.Width > 1200, 20,    // Large screens
    App.Width > 800, 16,     // Medium screens
    14                       // Small screens
)
```

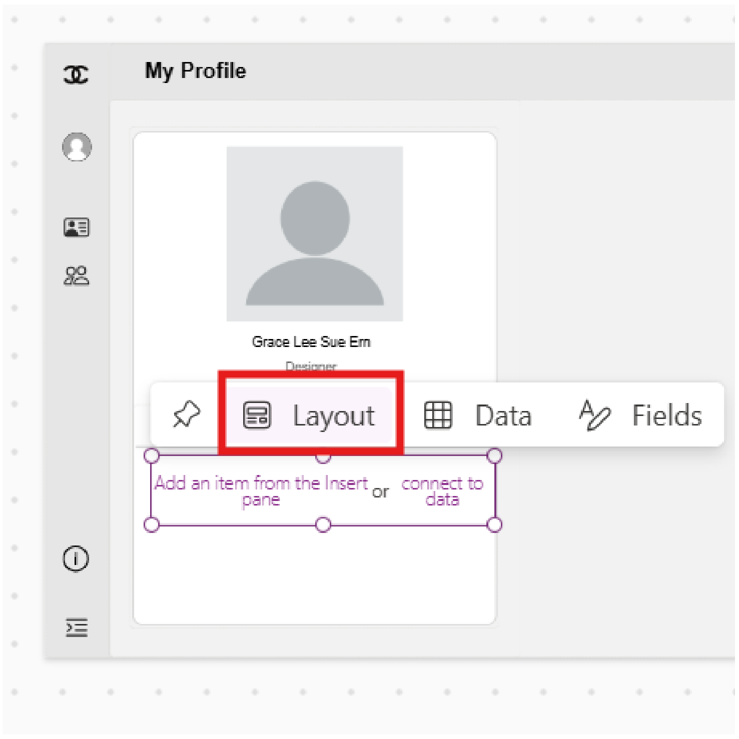
This ensures that the label can still be seen, regardless of screen size.

After following Step 24 to Step 27, the ProfilePage screen should look like this:

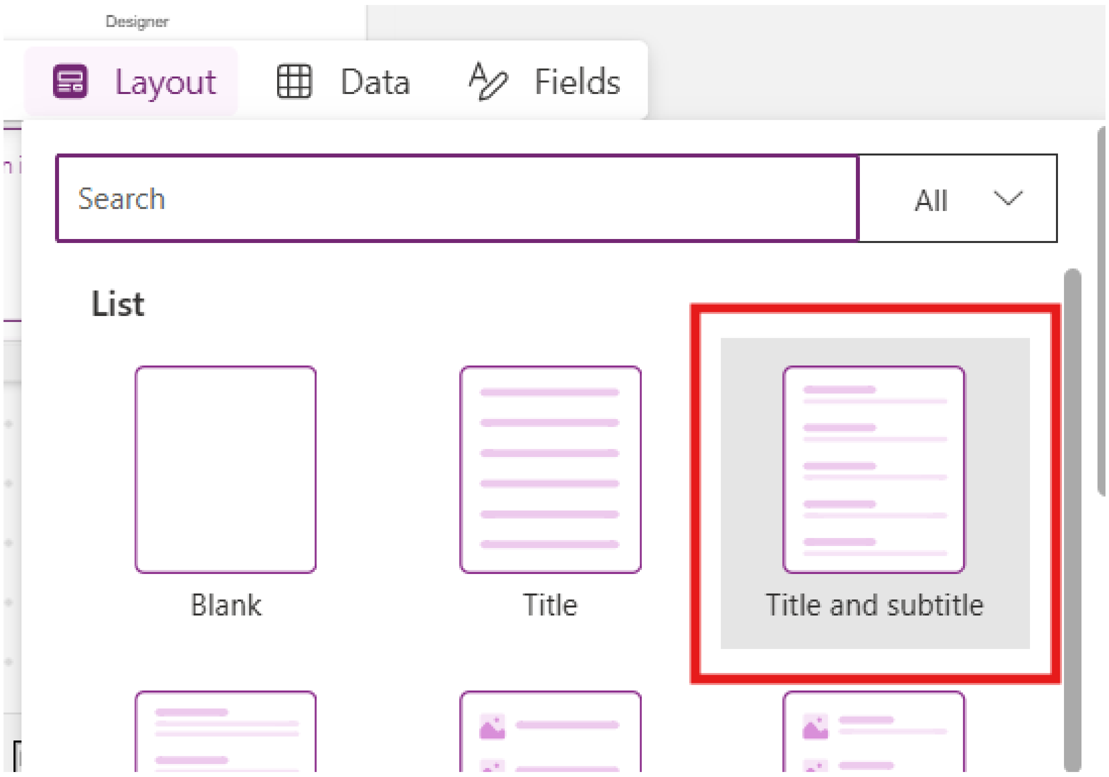


28. Set the properties of the Contacts_Gallery container to the following:
    - Set Height to 70% * Parent.Height
    - Set TemplateSize to 67.
    - Set Width to Parent.Width - Self.X.
    - Set X to 5% * ProfileLeftPanelBackground.Width.
    - Set Y to 120% * Separator1.Y.
    - Set ShowScrollBar to false.

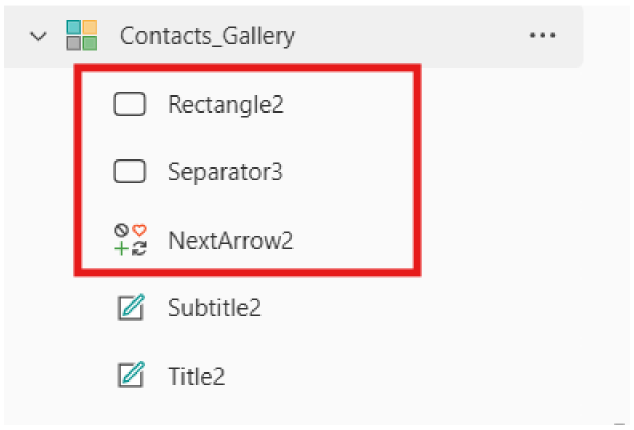
29. The layout of the Contacts_Gallery must now be adjusted. After hovering over Contacts_Gallery displayed on the screen, select the ‘Layout’ option displayed on the pop-up.



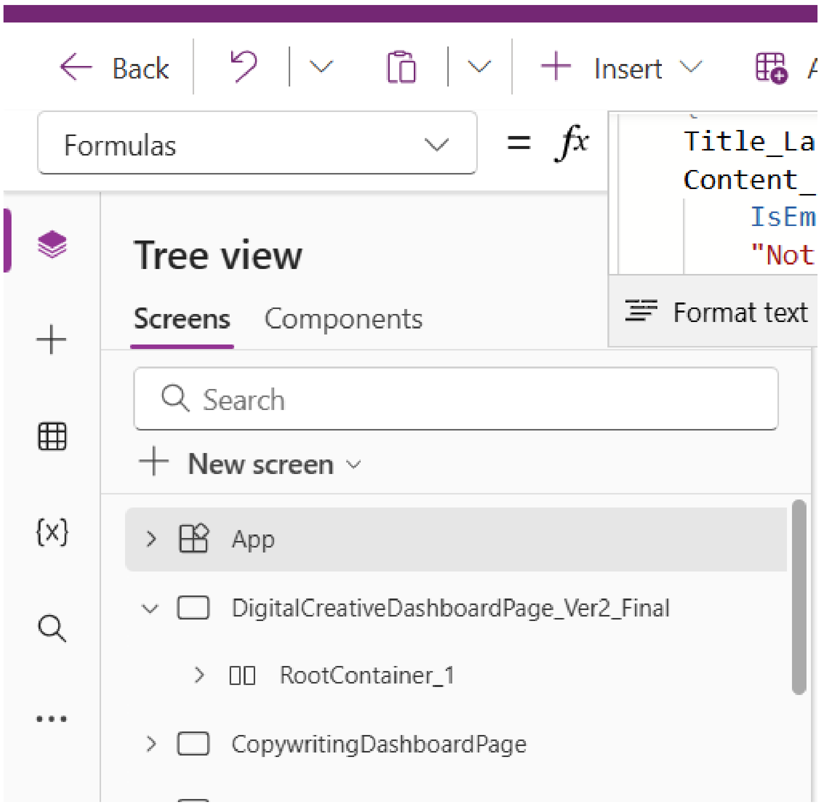
30. Select the ‘Title and Subtitle’ option displayed on the pop-up.



31. A number of items will be nested within the Contacts_Gallery. Select, right-click, and delete the following items encircled in red. Only the Title and Subtitle should remain within the gallery.



32. The contact information of the user shall now be added to Contacts_Gallery. Return to the top of the Tree View, select ‘App’, and select ‘Formulas’ within the drop-down menu.

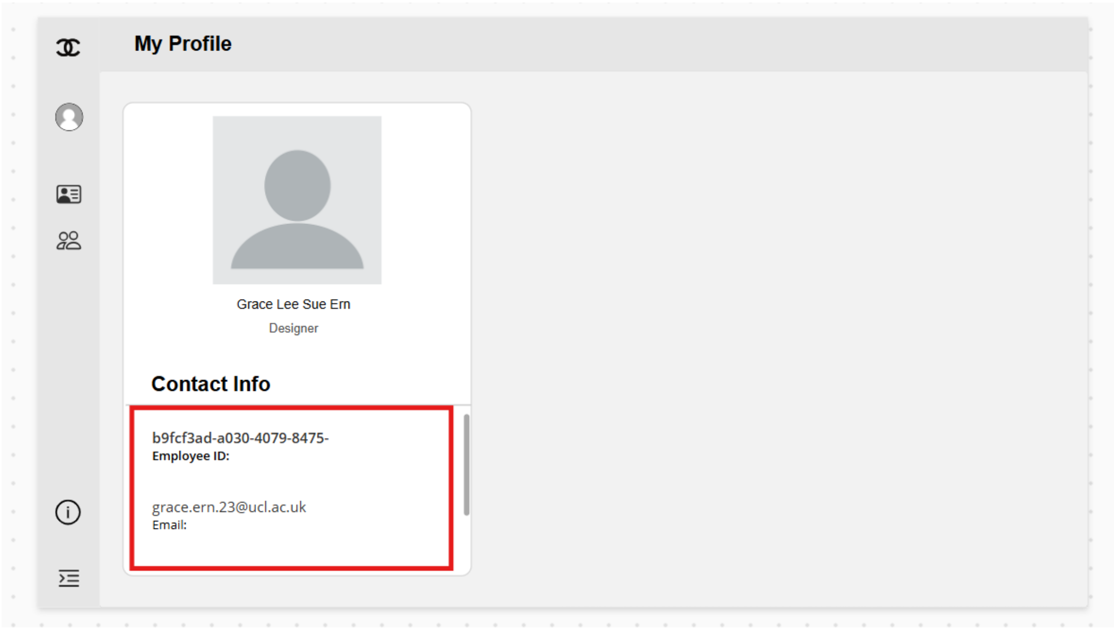


33. Within the formula bar of the newly selected Formula property, add the following table inside the formula bar:

```
ContactInfo_Table = Table(
    {
        Title_Label: "Employee ID:",
        Content_Label: If(
            IsBlankOrError(Office365Users.MyProfileV2().id),
            "Not Available",
            Office365Users.MyProfileV2().id
        )
    },
    {
        Title_Label: "Email:",
        Content_Label: Office365Users.MyProfileV2().mail
    },
    {
    Title_Label: "Phone Number:",
    Content_Label: If(
        IsEmpty(Office365Users.MyProfileV2().businessPhones), // Use
IsEmpty for tables
        "Not Available",
        Concat(Office365Users.MyProfileV2().businessPhones, Text(Value)
& ", ")
    )
    }
);
```

This table utilises data from the Office365User database to retrieve the employee ID, email and phone number of the logged-in user. It specifies the name of the title (i.e., Title_Label = "Employee ID") and the data to be displayed below the title (i.e., Content_Label: Office365Users.MyProfileV2().mail).

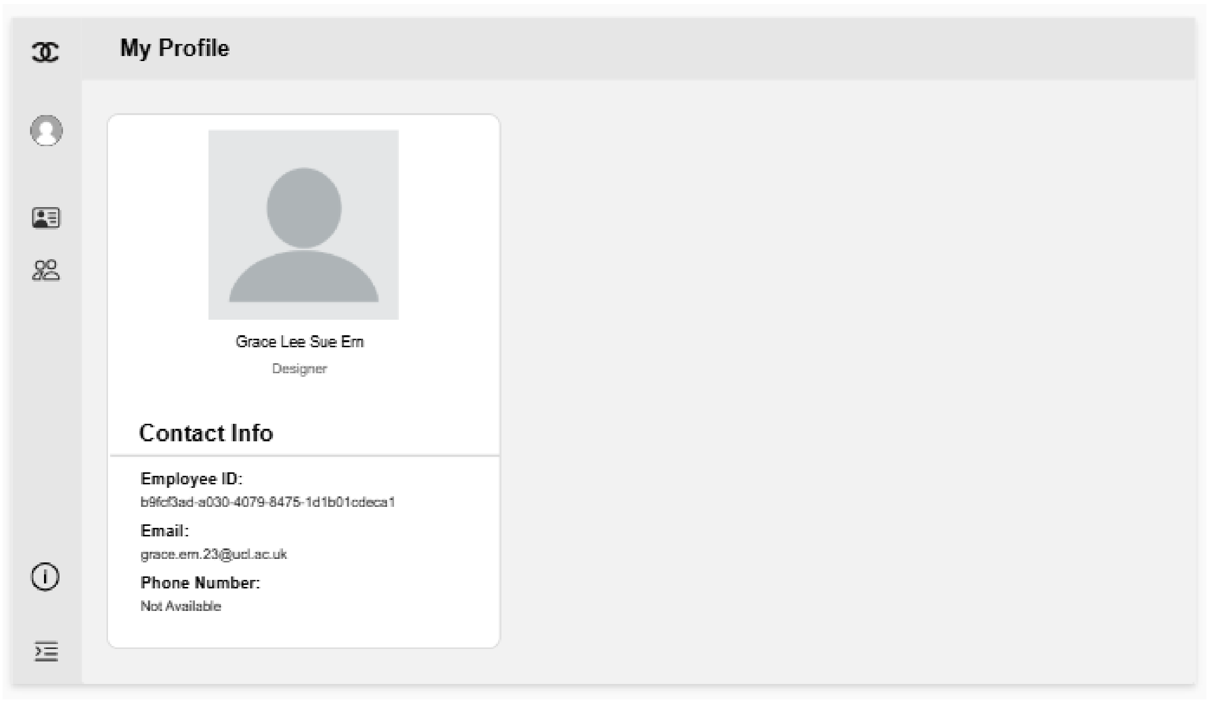
34. Select Contacts_Gallery within the Tree view. Now, set the Items property of the gallery to ContactInfo_Table, as specified in the previous step. Now, the Contacts_Gallery should display information, like the following:



35. Set the properties of Subtitle2 within the Contacts_Gallery to the following:
- Set Color to RBGA(0, 0, 0, 1).
- Set Font to Font.Arial.
- Set FontWeight to FontWeight.Normal.
- Set PaddingBottom to 0.
- Set PaddingLeft to 0.
- Set PaddingTop to 0.
- Set VerticalAlign to VerticalAlign.Middle.
- Set Width to Parent.TemplateWidth.
- Set Text to ThisItem.Content_Label.

36. Set the properties of Title2 within the Contacts_Gallery to the following:
- Set Color to RBGA(0, 0, 0, 1).
- Set Font to Font.Arial.
- Set FontWeight to FontWeight.Bold.
- Set PaddingBottom to 10.
- Set Width to Parent.TemplateWidth.
- Set Text to ThisItem.Title_Label.

The ProfilePage screen should now look like this:

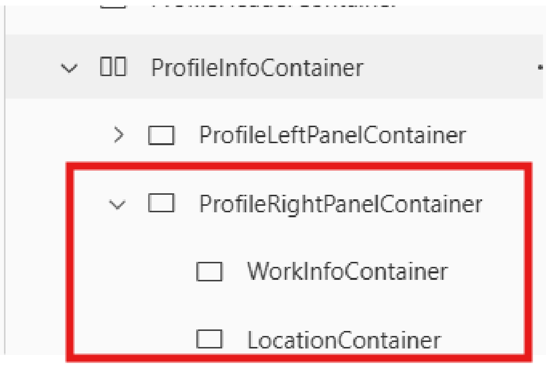

My Profile

Grace Lee Sue Em
Designer

Contact Info

Employee ID:
b9fcf3ad-a030-4079-8475-1d1b01cdeca1
Email:
grace.em.23@ucl.ac.uk
Phone Number:
Not Available

37. Insert two Containers within the ProfileRightPanelContainer. Name them as WorkInfoContainer and LocationContainer. As specified in the name, these containers will store the work information and work location/address of the logged-in user. The nested containers should look like this:



38. Edit the following properties of WorkInfoContainer:
- Set BorderStyle to BorderStyle.None.
- Set DropShadow to DropShadow.None.
- Set Height to (LocationContainer.Y - ImgAndContactsContainer.Y) * 70%.
- Set Width to LocationContainer.Width.
- Set X to Parent.Width/2 - Self.Width/2.
- Set Y to ImgAndContactsContainer.Y.

39. Nest the following items within the WorkInfoContainer:

    i. A button. Refer to Step 14 to add a button into WorkInfoContainer. Rename this button to WorkInformationBackground.
    ii. A text label. Refer to Step 21 to add a text label into WorkInfoContainer. Rename this label to WorkInformationLabel.
    iii. A horizontal line. Refer to Step 25 to add the horizontal line into WorkInfoContainer. Rename this to WorkInformationSeparator.

40. Edit the following properties of WorkInformationBackground so that it loses it's functionalities as a button and is redesigned to become the background of the Work Information container:
    - Set AutoDisableOnSelect to false.
    - Set Fill to RGBA(255, 255, 255, 1).
    - Set Height to 98% * Parent.Height
    - Set HoverColor, HoverFill and Text to nothing/null.
    - Set RadiusBottomLeft, RadiusTopLeft, RadiusBottomRight, RadiusTopRight to 14.
    - Set Width to 98% * Parent.Width.
    - Set X to Parent.Width/2 - Self.Width/2.
    - Set Y to Parent.Height/2 - Self.Height/2.

41. Edit the following properties of WorkInformationSeparator:
    - Set BorderColor to RGBA(202, 202, 202, 1).
    - Set BorderThickness to 1.
    - Set Color to RGBA(237, 237, 237, 1).
    - Set Fill to RGBA(237, 237, 237, 1).
    - Set Width to *WorkInformationBackground*.Width.
    - Set Height to 0.
    - Set X to WorkInformationBackground.X.
    - Set Y to 17% * Parent.Height.

42. Edit the following properties of WorkInformationLabel:
    - Set Font to Font.Arial.
    - Set FontWeight to FontWeight.Semibold.
    - Set Height to Locations_MainTitle.Height.
    - Set PaddingLeft, PaddingRight, PaddingTop and PaddingBottom to 0.
    - Set Text to 'Work Information'.
    - Set Width to 90% * Parent.Width.
    - Set X to Locations_MainTitle.X.
    - Set Y to (WorkInformationSeparator.Y - WorkInformationBackground.Y)/2 - Self.Height/2

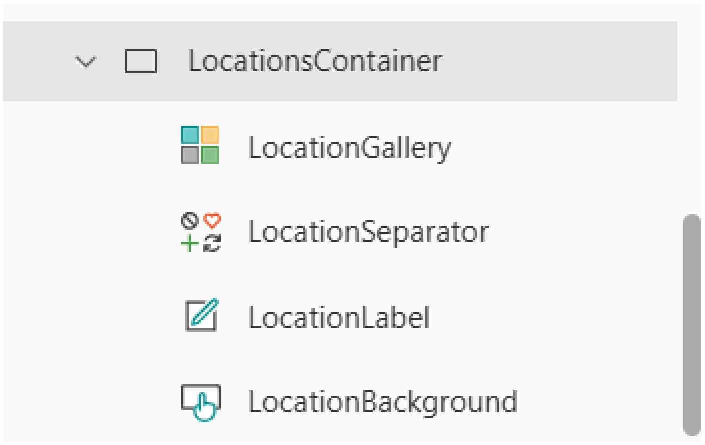
- Set Size to the following formula:

```
If(
    App.Width > 1200, 20,    // Large screens
    App.Width > 800, 16,     // Medium screens
    14                       // Small screens
)
```

43. Edit the following properties of the LocationContainer:
    - Set BorderStyle to BorderStyle.None.
    - Set DropShadow to DropShadow.None.
    - Set Height to *ContactInformContainer*.Height.
    - Set Width to 95% * Parent.Width.
    - Set X to Parent.Width/2 - Self.Width/2.
    - Set Y to 110% * (*UserTitleContainer*.Y + *UserTitleContainer*.Height) + *ImgAndContactsContainer*.Y

44. Nest the following items within the LocationContainer:
    - A button. Refer to Step 14 to add a button into LocationContainer. Rename this button to LocationBackground.
    - A text label. Refer to Step 21 to add a text label into LocationContainer. Rename this label to LocationLabel.
    - A horizontal line. Refer to Step 25 to add the horizontal line into LocationContainer. Rename this to LocationSeparator.
    - A Vertical Gallery. Refer to Step 25 to add this to the container. Rename this to LocationGallery.

    Your LocationContainer should have the following children items nested within it:



LocationsContainer
LocationGallery
LocationSeparator
LocationLabel
LocationBackground

45. Edit the following properties of LocationBackground so that it loses it’s functionalities as a button and is redesigned to become the background of the Location Information container:

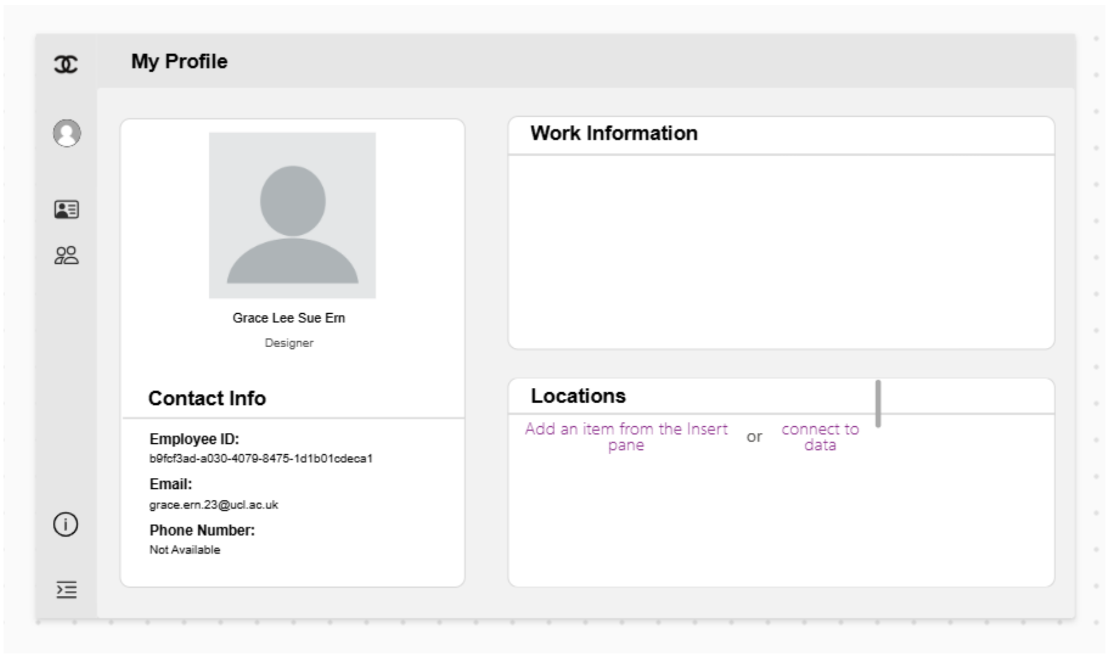
- Set AutoDisableOnSelect to false.
- Set Fill to RGBA(255, 255, 255, 1).
- Set Height to Parent.Height
- Set HoverColor, HoverFill and Text to nothing/null.
- Set RadiusBottomLeft, RadiusTopLeft, RadiusBottomRight, RadiusTopRight to 14.
- Set Width to 98% * Parent.Width.
- Set X to Parent.Width/2 - Self.Width/2.
- Set Y to Parent.Height/2 - Self.Height/2.

46. Set the properties of LocationSeparator to the following:
    - Set BorderColor to RGBA(202, 202, 202, 1).
    - Set BorderThickness to 1.
    - Set Color to RGBA(237, 237, 237, 1).
    - Set Fill to RGBA(237, 237, 237, 1).
    - Set Width to *LocationBackground*.Width.
    - Set Height to 0.
    - Set X to LocationBackground.X.
    - Set Y to 17% * Parent.Height.

47. Edit the following properties of LocationLabel:
    - Set Font to Font.Arial.
    - Set FontWeight to FontWeight.Semibold.
    - Set Height to *ContactInfo_MainTitle*.Height.
    - Set PaddingLeft, PaddingRight, PaddingTop and PaddingBottom to 0.
    - Set Text to ‘Locations’.
    - Set Width to 90% * Parent.Width.
    - Set X to *ContactInfo_MainTitle*.X
    - Set Y to (*Locations_Separator*.Y - *LocationsBackground*.Y)/2 - Self.Height/2
    - Set Size to the following:

```
If(
    App.Width > 1200, 20,    // Large screens
    App.Width > 800, 16,     // Medium screens
    14                       // Small screens
)
```

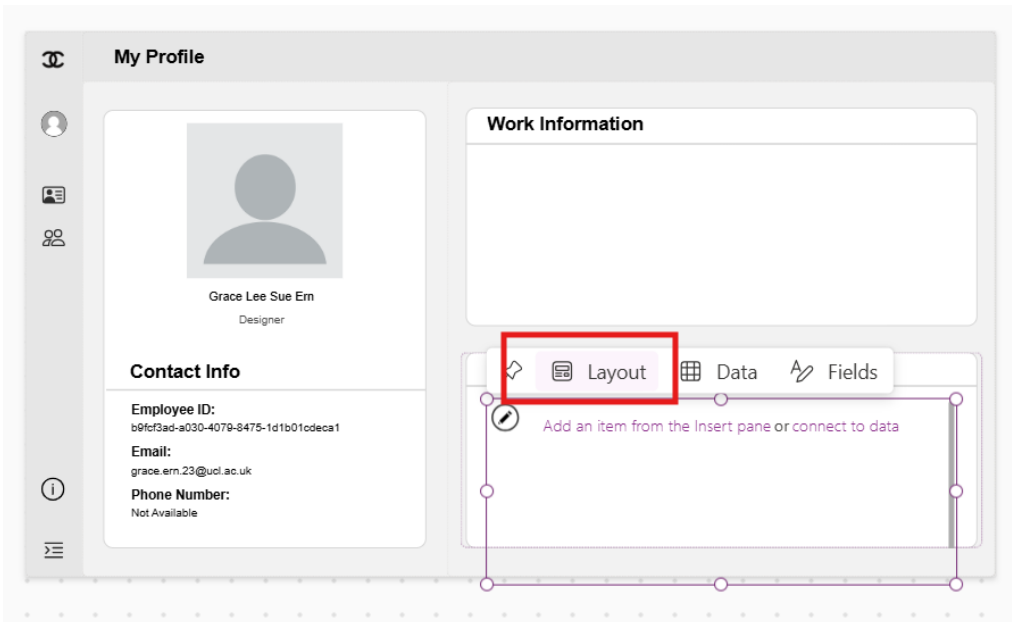
At this point, the ProfilePage should look like this:



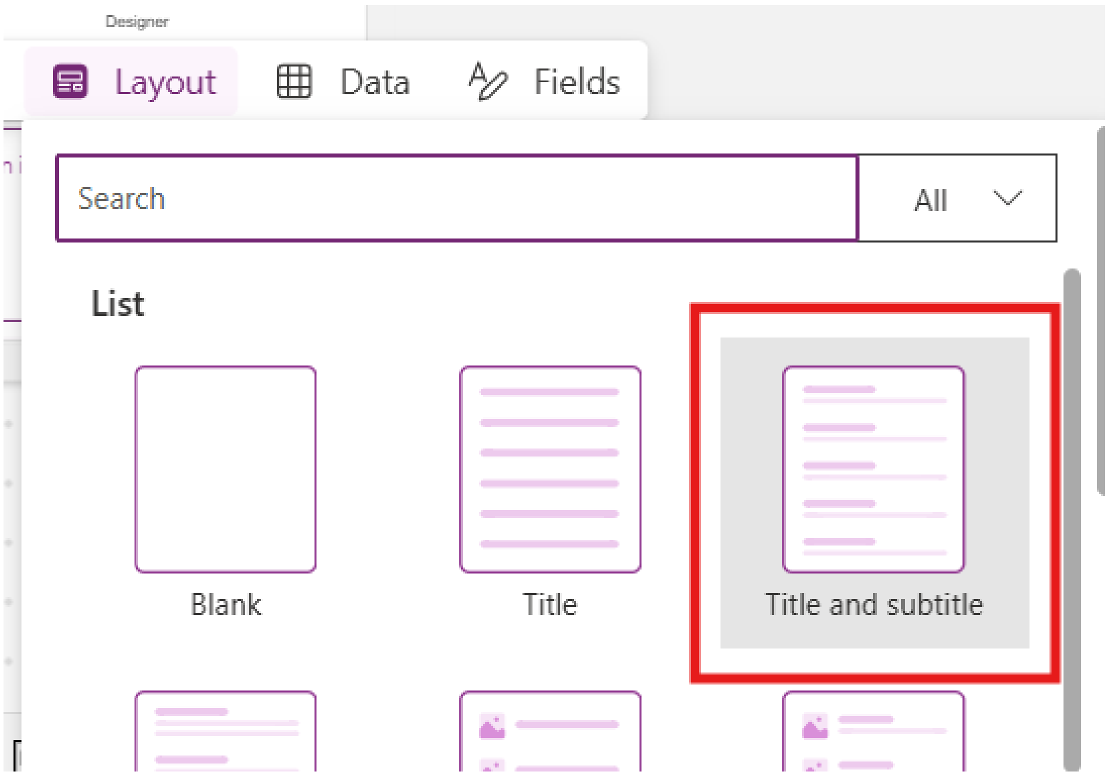
Grace Lee Sue Ern
Designer
Contact Info
Employee ID:
b9fcf3ad-a030-4079-8475-1d1b01cdeca1
Email:
grace.ern.23@ucl.ac.uk
Phone Number:
Not Available

48. Edit the following properties of the LocationGallery:
    - Set Height of the LocationGallery to Self.AllItemsCount * (Self.TemplateHeight + Self.TemplatePadding)
    - Set TemplateSize to 60.
    - Set Width to Parent.Width * 90%.
    - Set X to LocationLabel.X
    - Set Y to 105% * Contacts_Gallery.Y
    - Set Items to Address_Table. If an error occurs, jump to Step 53 before returning to Step 49.

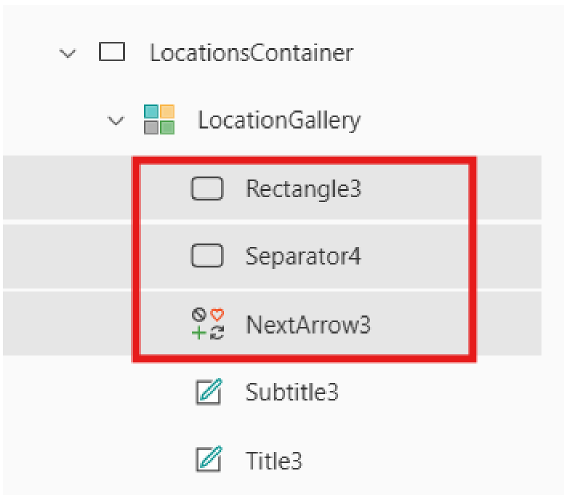
49. Hover over LocationGallery displayed on the screen, select the ‘Layout’ option displayed on the pop-up.



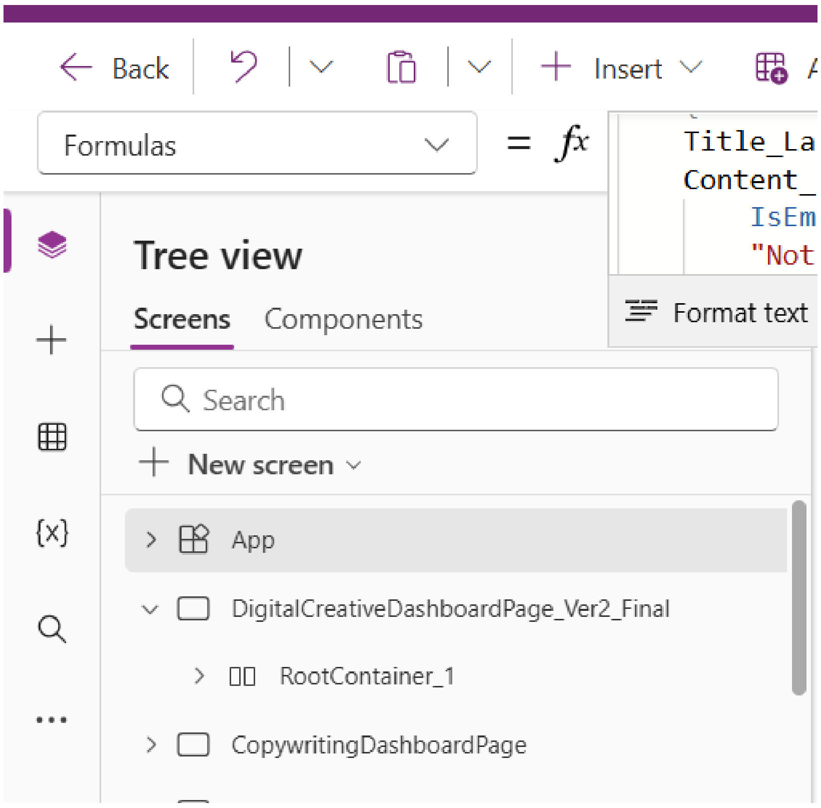
50. Select the ‘Title and Subtitle’ option displayed on the pop-up.



51. A number of items will be nested within the LocationGallery. Select, right-click, and delete the following items encircled in red. Only the Title and Subtitle should remain within the gallery.



52. All information regarding the location of the logged-in user shall now be added to LocationGallery. Return to the top of the Tree View, select 'App', and select 'Formulas' within the drop-down menu.



53. Add the following table, titled 'Address_Table', into the formula bar of the selected 'Formulas' property:

```
Address_Table = Table(
    {
        Title_Label: "Office Address:",
        Content_Label: If (
            IsBlankOrError(Office365Users.MyProfileV2().officeLocation)
,
            "5 Barlow Place, London, United Kingdom, W1J 6DG",
            Office365Users.MyProfileV2().officeLocation
        )
    },
    {
        Title_Label: "Primary Region:",
        Content_Label: If(
            IsBlankOrError(Office365Users.MyProfileV2().country),
            "United Kingdom",
            Office365Users.MyProfileV2().country
        )
    }
);
```

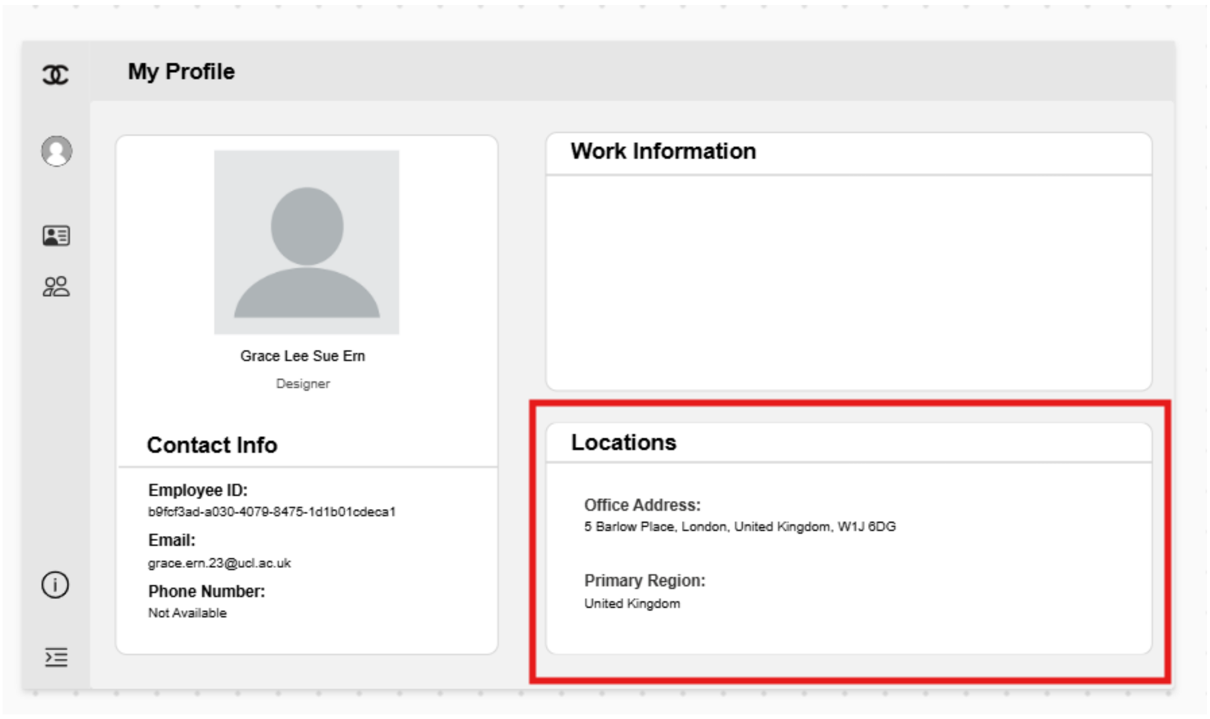
This table leverages data from the **Office365Users** database to retrieve the logged-in user's **address** and **country**. It defines a title label (e.g., Title_Label = "Office Address/Primary Region") and displays the corresponding data beneath it (e.g., Content_Label = Office365Users.MyProfileV2().OfficeLocation). Each content label includes a fallback option to ensure a default value is displayed if the data cannot be retrieved.

54. Select the 'Subtitle3' text label nested within the LocationGallery. Edit the properties of Subtitle3 to the following:
    - Set Font to Font.Arial.
    - Set FontWeight to FontWeight.Normal.
    - Set PaddingBottom to 7.
    - Set PaddingTop to 7.
    - Set Text to ThisItem.Content_Label.
    - Set VerticalAlign to VerticalAlign.Middle.

55. Select the 'Title3' text label nested within the LocationGallery. Edit the properties of Title3 to the following:
    - Set Font to Font.Arial.
    - Set FontWeight to FontWeight.Semibold.
    - Set PaddingBottom to 10.

- Set Text to ThisItem.Title_Label.
- Set Width to Parent.TemplateWidth.
- Set Y to (Parent.TemplateHeight - (Self.Size * 1.8 + *Locations_Subtitle_2*.Size * 1.8)) / 2.

The LocationsContainer of the application should now display as follows:



My Profile

Grace Lee Sue Ern

Designer

Contact Info

Employee ID:
b9fcf3ad-a030-4079-8475-1d1b01cdeca1

Email:
grace.ern.23@ucl.ac.uk

Phone Number:
Not Available

Work Information

Locations

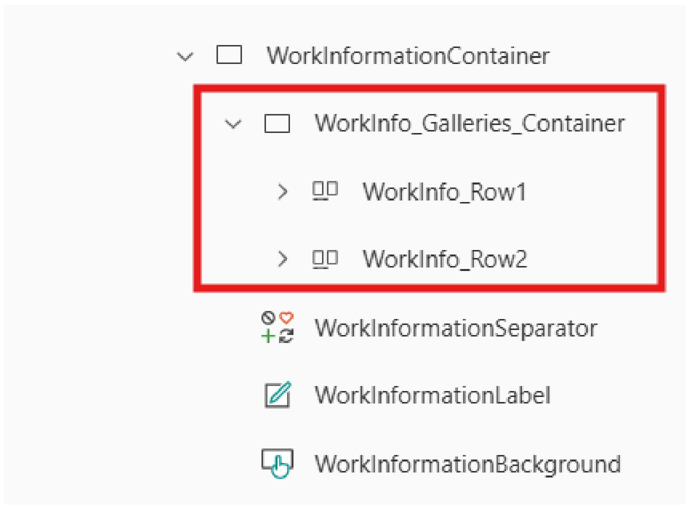
Office Address:
5 Barlow Place, London, United Kingdom, W1J 6DG

Primary Region:
United Kingdom

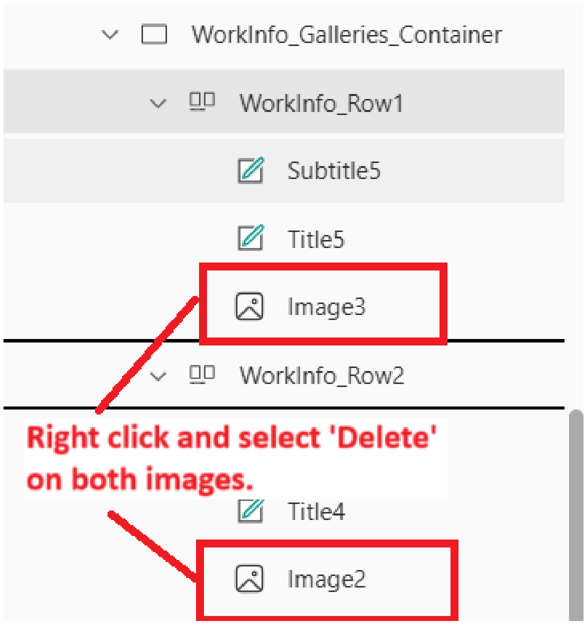
56. Select the WorkInfoContainer created in Step 37 and nest a Container within it. Rename this Container to WorkInfo_Galleries_Container.

57. Set the following properties to WorkInfo_Galleries_Container:
    - Set BorderStyle to BorderStyle.None.
    - Set DropShadow to DropShadow.None.
    - Set Height to 80% * Parent.Height.
    - Set Width to 90% * Parent.Width.
    - Set X to *WorkInformationLabel*.X
    - Set Y to *WorkInformationSeparator*.Y

58. Nest two Horizontal Galleries within WorkInfo_Galleries_Container. Name the galleries as WorkInfo_Row1 and WorkInfo_Row 2 respectively. By now, your WorkInfo_Galleries_Container should look like this:



59. Remove the images nested within WorkInfo_Row1 and WorkInfo_Row2.


WorkInfo_Galleries_Container
WorkInfo_Row1
Subtitle5
Title5
Image3
WorkInfo_Row2
Right click and select 'Delete' on both images.
Title4
Image2

60. Set the properties of WorkInfo_Row1 to the following:
    - Set BorderStyle to BorderStyle.None.
    - Set DisplayMode to DisplayMode.View.
    - Set Height to 30% * Parent.Height.
    - Set ShowScrollBar to false.
    - Set TemplateSize to 209.
    - Set Width to Parent.Width.
    - Set X to Parent.Width/2 - Self.Width/2.
    - Set Y to 95% * Parent.Height/2 - Self.Height.
    - Set Items to ProfilePage_Table. If an error occurs, jump to Step 62 before returning here.

61. Set the properties of WorkInfo_Row2 to the following:

- Set Height to WorkInfo_Row1.Height.
- Set ShowScrollbar to false.
- Set TemplateSize to 209.
- Set Width to 90% * Parent.Width.
- Set X to WorkInfo_Row1.X.
- Set Y to 150% * Parent.Height/2 - Self.Height/2.
- Set Items to ProfilePage1_Table. If an error occurs, jump to Step 62 before returning here.

62. This step will describe how the work information of the logged-in user will be displayed. Return to the Formulas bar as specified in Step 52, and add the following two tables into the formula bar of the selected ‘Formulas’ property:

```
ProfilePage_Table = Table(
    {
        Title_Label: "Team:",
        Content_Label: Office365Users.MyProfileV2().department
    },
    {
      Title_Label: "Time In Position:",
        //first formula calculates the number of years worked.
        Content_Label:
Round(DateDiff(DateValue(Office365Users.MyProfileV2().hireDate),
Today(), TimeUnit.Years), 0) & " years, " &

        //adds back full years to the start date, calculates months
remaining after those full years
        DateDiff(DateAdd(DateValue(Office365Users.MyProfileV2().hireDat
e),
            Round(DateDiff(DateValue(Office365Users.MyProfileV2().hireD
ate), Today(), TimeUnit.Years), 0),
             TimeUnit.Years),
            Today(),
            TimeUnit.Months) & " months"

    }
);

ProfilePage1_Table = Table(
    {
        Title_Label: "Manager:",
        Content_Label: If(
            IsBlankOrError(Office365Users.ManagerV2(User().Email).displ
ayName),
            "N/A",
            Office365Users.ManagerV2(User().Email).displayName
        )
    },
    {
        Title_Label: "Date of Joining:",
        Content_Label: If(
        !IsBlank(Office365Users.MyProfileV2().hireDate) &&
!IsError(Office365Users.MyProfileV2().hireDate),
        Text(DateValue(Office365Users.MyProfileV2().hireDate), "dd mmm
yyyy"),
        "Not Available"
    )
    },
    {
        Title_Label: "Time in Company:",
```

This PowerApps formula defines two tables, ProfilePage_Table and ProfilePage1_Table, which store user profile information retrieved from Office 365 Users. The tables contain key details such as department, manager, hire date, office location, and tenure in the company. The Time in Position and Time in Company fields calculate the user's tenure by determining the number of full years and remaining months since their hire date. Additionally, fallback conditions ensure that missing or erroneous data (e.g., manager name or hire date) is replaced with default values like "N/A" or "Not Available" for a seamless user experience.

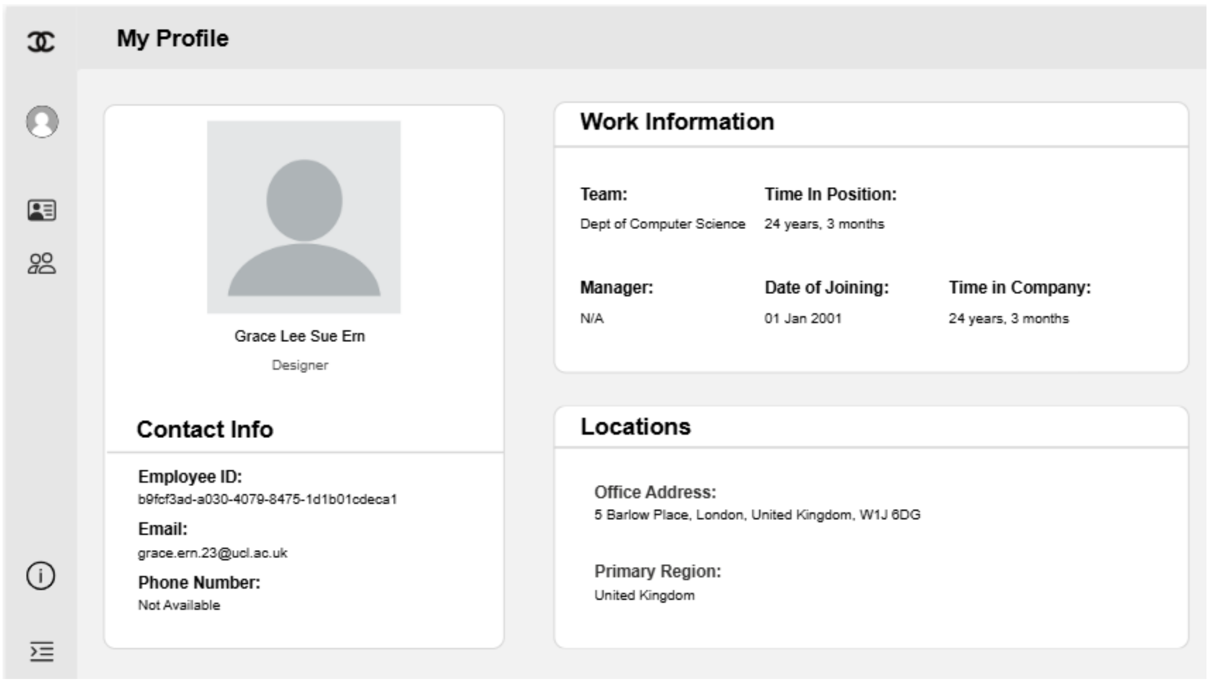
63. Select the ‘Subtitle5’ text label nested within the WorkInfo_Row1. Edit the properties of Subtitle5 to the following:
    - Set Font to Font.Arial.
    - Set FontWeight to FontWeight.Normal.
    - Set Height to Self.Size * 1.8.
    - Set PaddingBottom to 7.
    - Set PaddingTop to 7.
    - Set Size to 12.
    - Set Text to ThisItem.Content_Label.
    - Set VerticalAlign to VerticalAlign.Middle.
    - Set Width to Parent.TemplateWidth

64. Repeat Step 63 for ‘Subtitle4’ text label nested within WorkInfo_Row2.

65. Select the ‘Title5’ text label nested within the WorkInfo_Row1. Edit the properties of Title5 to the following:
    - Set Color to RGBA(0, 0, 0, 1).
    - Set Font to Font.Arial.
    - Set FontWeight to FontWeight.Semibold.
    - Set PaddingBottom to 10.
    - Set Height to Self.Size * 1.8.
    - Set Text to ThisItem.Title_Label.
    - Set Width to Parent.TemplateWidth.
    - Set X to (Parent.TemplateWidth - Self.Width)/2.
    - Set Y to 0.

66. Repeat Step 65 for ‘Title4’ text label nested within WorkInfo_Row2.

This concludes the documentation for both the navigation and the Profile page aspects of the Power App. The finalised ProfilePage screen should look like the following:



Grace Lee Sue Ern

Designer

Contact Info

Employee ID:
b9fcf3ad-a030-4079-8475-1d1b01cdeca1

Email:
grace.ern.23@ucl.ac.uk

Phone Number:
Not Available

Work Information

Team:	**Time In Position:**	
Dept of Computer Science	24 years, 3 months	
Manager:	**Date of Joining:**	**Time in Company:**
N/A	01 Jan 2001	24 years, 3 months

Locations

Office Address:
5 Barlow Place, London, United Kingdom, W1J 6DG

Primary Region:
United Kingdom