

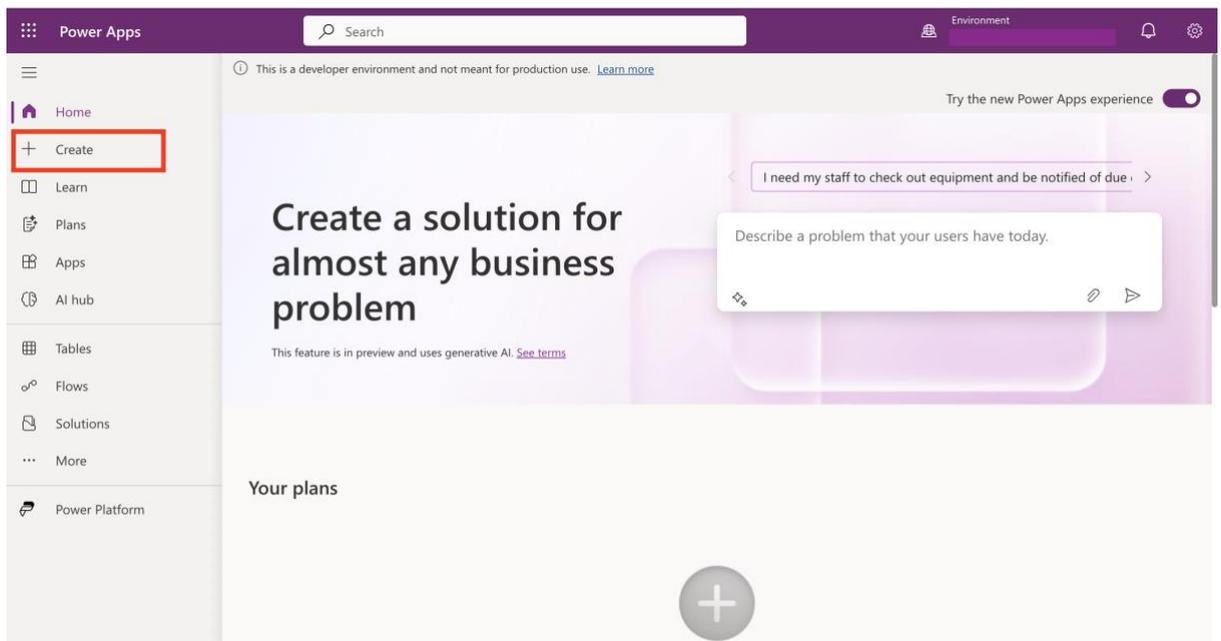
Creating a Navigation Menu in Power Apps

Steps:

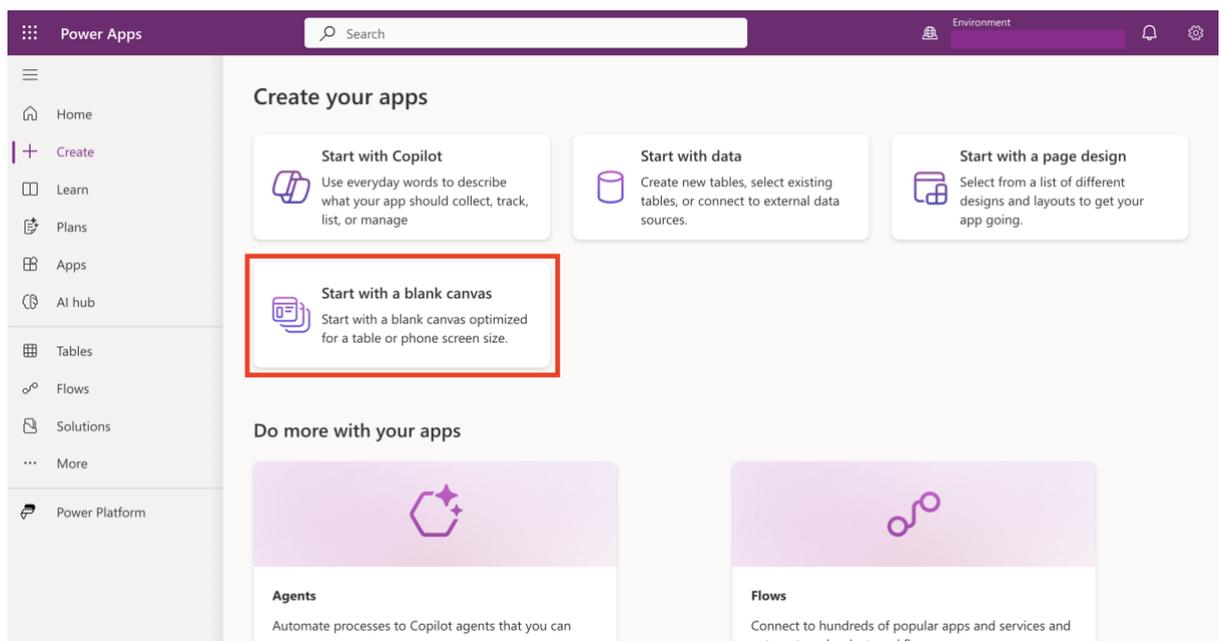
Create a new Canvas App

Steps – Creating a new Canvas App:

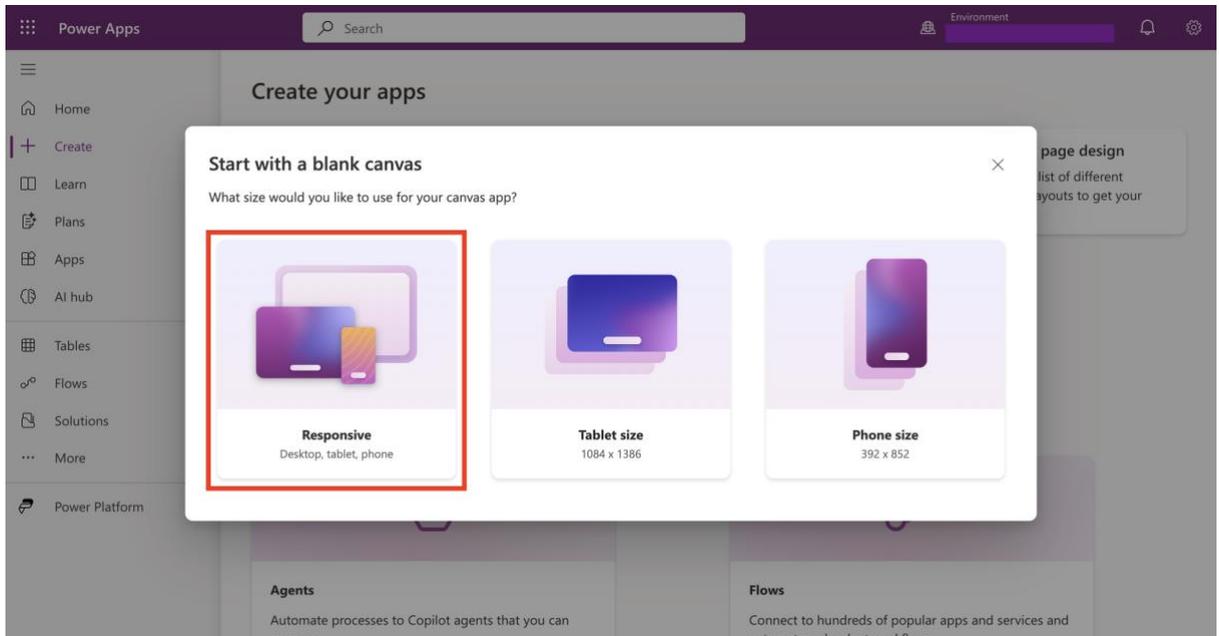
- Open the Power Apps from your Microsoft 365 portal and choose “Create”.



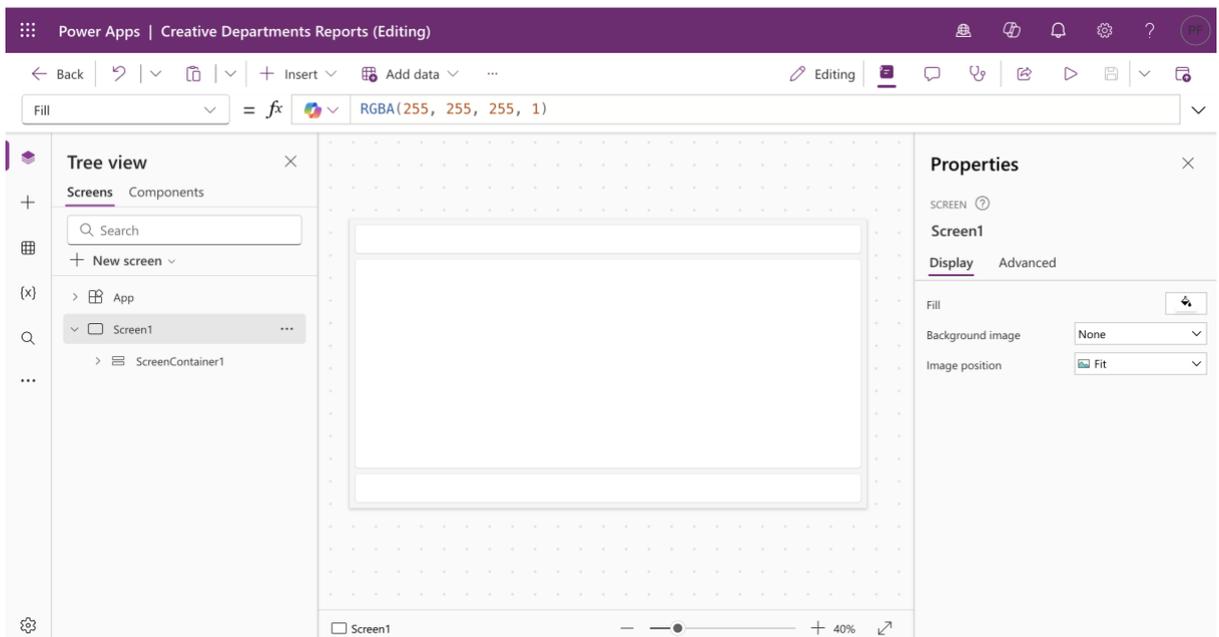
- Choose “Start with a blank canvas”.



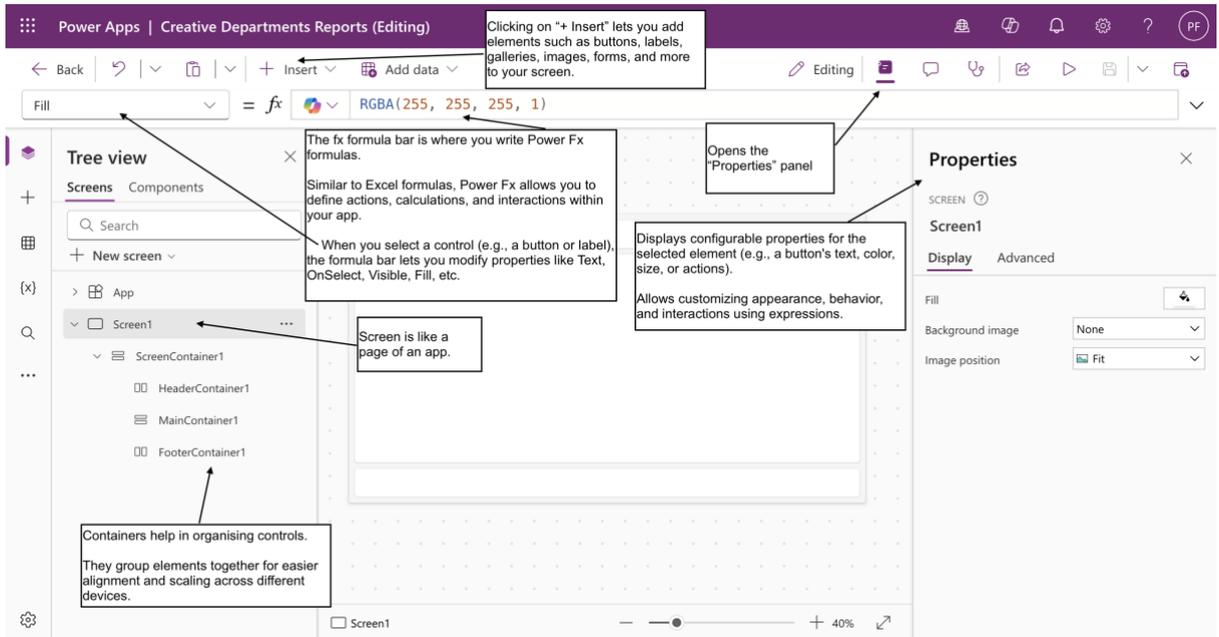
- Select “Responsive” size for the canvas app.



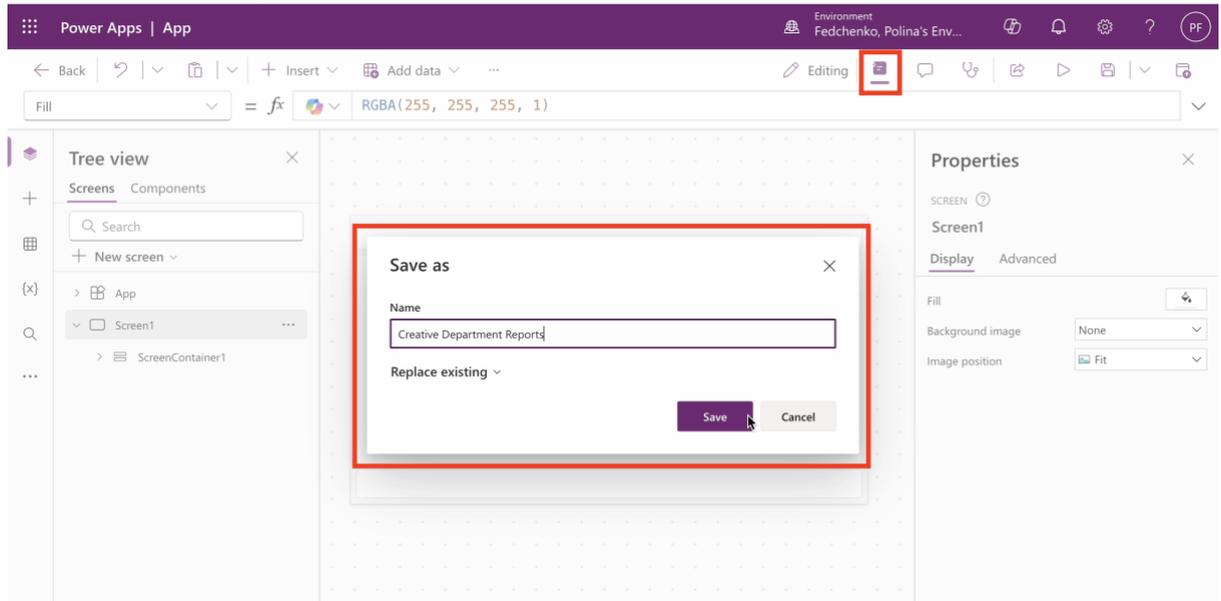
- You will be redirected straight to a blank canvas app as shown below:



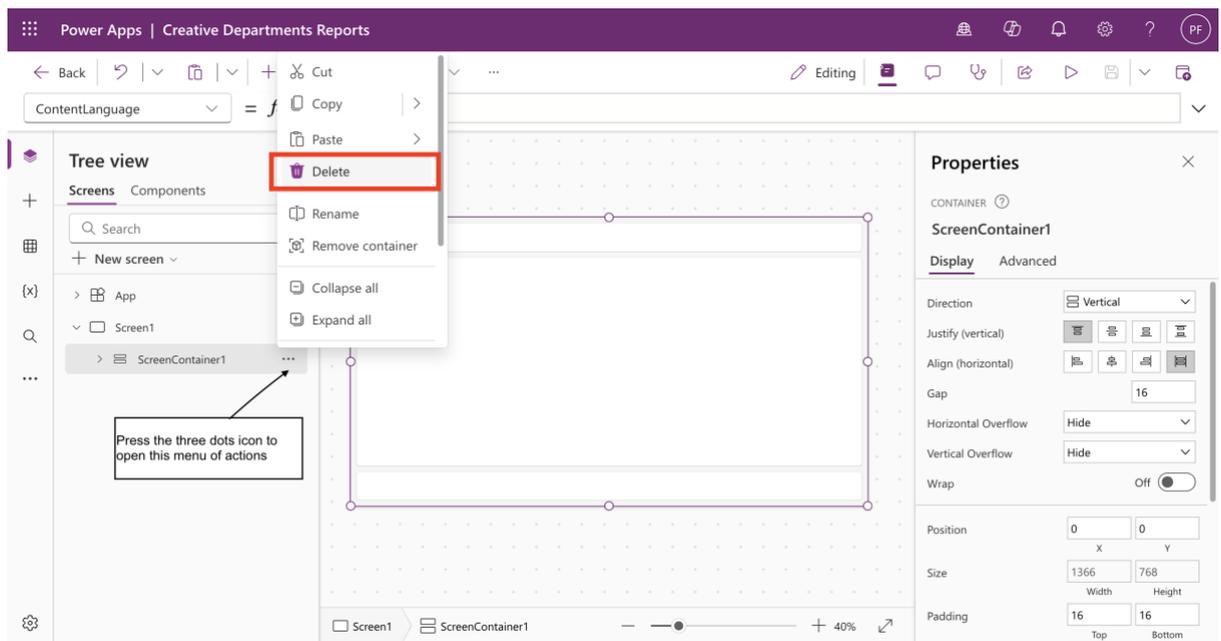
- Using the screenshot below you can familiarise yourself with key development features of Power Apps:



- Save the app and give it a name:



- The contents of Screen 1 can be cleared for now:



Define Named Formulas

What are Named Formulas in Power Apps?

Named Formulas in Power Apps is a way to define reusable calculations, values, or logic that can be referenced throughout the app.

Named formulas are the equivalent of named cells and named formulas in Excel, managed with the Name Manager.[1]

They are defined in **App.Formulas** and the definition is the single source of truth, and the value cannot be changed somewhere else in the app.[1]

If the named formula performs calculation that is dependent on control properties or database records, it gets automatically recalculated when these dependencies change. [1]

Alternative to named formulas for initializing data and setting global variables - OnStart property, which only executes once when the app starts, and has slower load up time compared to named formulas.

Advantages of using named formulas over the OnStart property:

- Consistency: ensure all components use the same values throughout the app
- Easier maintenance: if you need to update a value, you only change it in one place
- Better readability: avoid repetitive long formulas across various controls in the app
- Improved performance: computed once and reused, reducing the number of repeated calculations in different controls
- Faster app load up time

Named Formula for the Navigation Menu Icons

To make the navigation menu more intuitive and visually appealing we will **add icons** to represent different sections of the app.

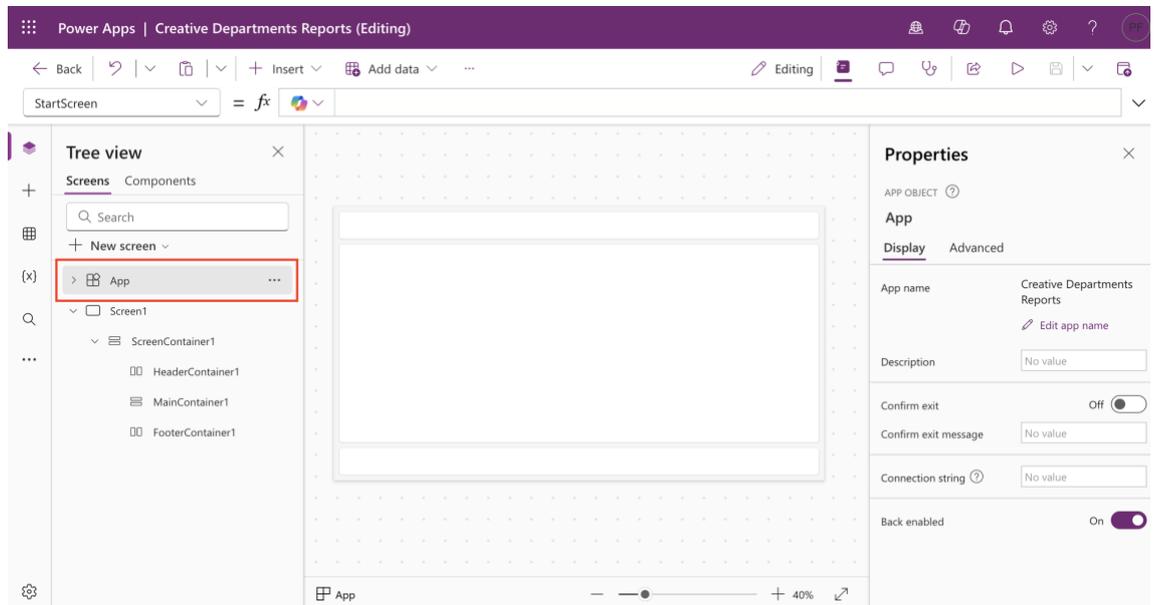
We will define a **named formula** that assigns icons to buttons used for switching between the "Individual Reports" and "Team Reports" pages.

In the later steps, we will create the **navigation menu component**, integrate it into all screens within the app and **use this named formula for the icons** as shown on the screenshots below.

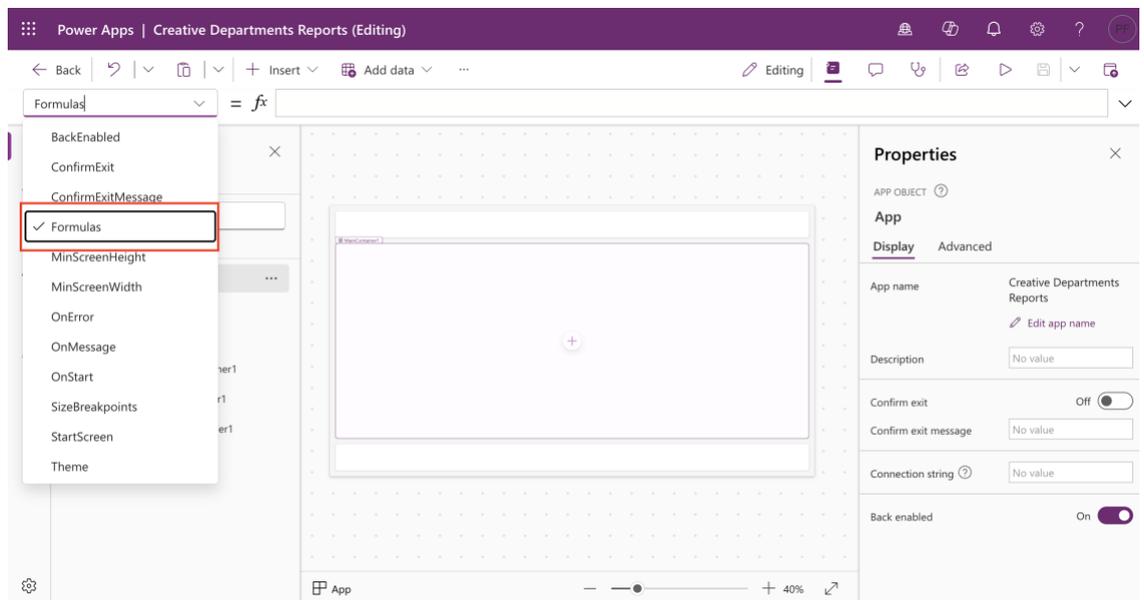


Steps - How to create Named Formula for Navigation Menu's icons:

- Select “App” in the “Tree View” on the left:



- Then select “Formulas” property from the drop-down list:



- Set up a **custom variable (= the named formula) “customIcons”** which is a Table. This table allows to define and use **custom icons** instead of using built-in Power App icons.

What is in the Table?

Each row in the table contains two pieces of information:

- **IconName:** A name to identify the icon (e.g., "Individual_Reports_Filled")
- **IconData:** The actual icon design written in SVG format.
 - For the purpose of creating the app described in this documentation, the SVG code was taken from the [free open source Bootstrap Icons library](#) [2]

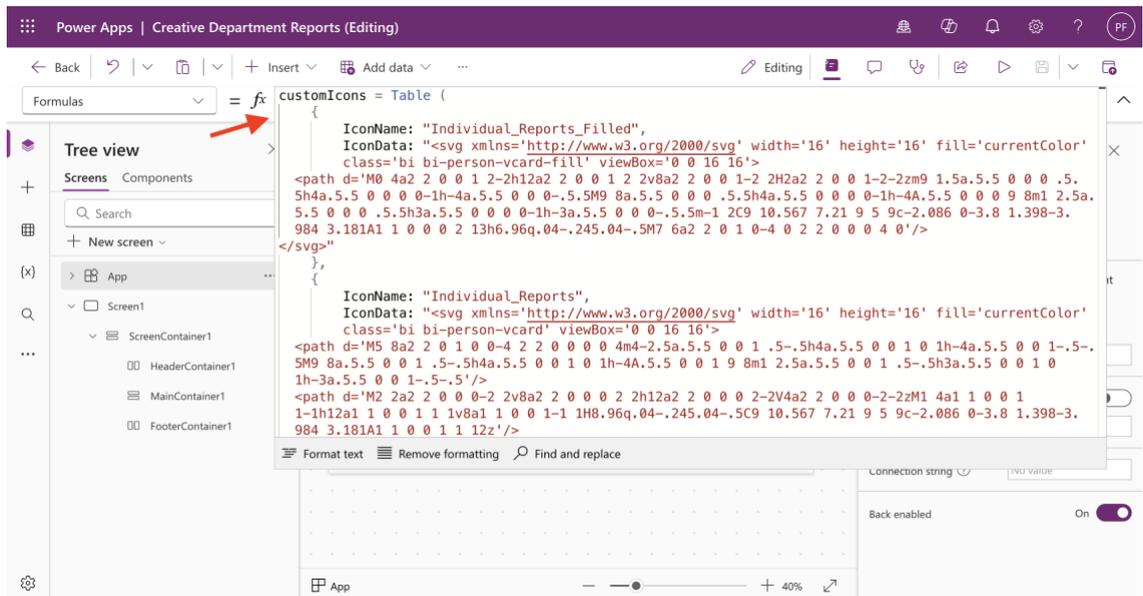
Why use this named formula Table?

- Allows you to store and organize multiple icons in one place (you can easily update them just in a single place if needed).
- Makes it easy to switch icons dynamically (e.g., on button click).
- Ensures icons look consistent throughout the app.

Bootstrap icons used:

“IconName” (from the customIcon Table)	Name & Link of the icon in Bootstrap	The image of the icon
Individual_Reports_Filled	Person vcard fill	
Individual_Reports	Person vcard	
Team_Reports_Filled	People fill	
Team_Reports	People	

You can copy and paste the formula for “customIcons” given below into the fx formula bar like so:



Notes:

- Always remember to put the semicolon (“;”) at the end of your line of formula in the fx formula bar.
- “IconData” contains the SVG code enclosed in double quotes. Therefore, before pasting the SVG code from Bootstrap, you have to replace all internal double quotes with single quotes, to avoid errors in Power App.

Formula:

```

customIcons = Table (
    {
        IconName: "Individual_Reports_Filled",
        IconData: "<svg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' fill='currentColor' class='bi bi-person-
vcard-fill' viewBox='0 0 16 16'>
    <path d='M0 4a2 2 0 0 1 2-2h12a2 2 0 0 1 2 2v8a2 2 0 0 1-2
2H2a2 2 0 0 1-2-2zm9 1.5a.5.5 0 0 0 .5.5h4a.5.5 0 0 0 0-1h-4a.5.5
0 0 0 0-.5.5M9 8a.5.5 0 0 0 .5.5h4a.5.5 0 0 0 0-1h-4A.5.5 0 0 0 9
8m1 2.5a.5.5 0 0 0 .5.5h3a.5.5 0 0 0 0-1h-3a.5.5 0 0 0-.5.5m-1
2C9 10.567 7.21 9 5 9c-2.086 0-3.8 1.398-3.984 3.181A1 1 0 0 0 2
13h6.96q.04-.245.04-.5M7 6a2 2 0 1 0-4 0 2 2 0 0 0 4 0'/'>
</svg>"
    },
    {
        IconName: "Individual_Reports",
        IconData: "<svg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' fill='currentColor' class='bi bi-person-
vcard' viewBox='0 0 16 16'>
    <path d='M5 8a2 2 0 1 0 0-4 2 2 0 0 0 4m4-2.5a.5.5 0 0 1 .5-.5h4a.5.5 0 0 1 0-1h-4a.5.5 0 0 1-.5-
.5M9 8a.5.5 0 0 1 .5-.5h4a.5.5 0 0 1 0-1h-4A.5.5 0 0 1 9 8m1 2.5a.5.5 0 0 1 0-1
1h12a1 1 0 0 1 1v8a1 1 0 0 1-1 1H8.96q.04-.245.04-.5C9 10.567 7.21 9 5 9c-2.086 0-3.8 1.398-3.
984 3.181A1 1 0 0 1 1 12z'/'>
    </svg>"
    }
)
    
```

```

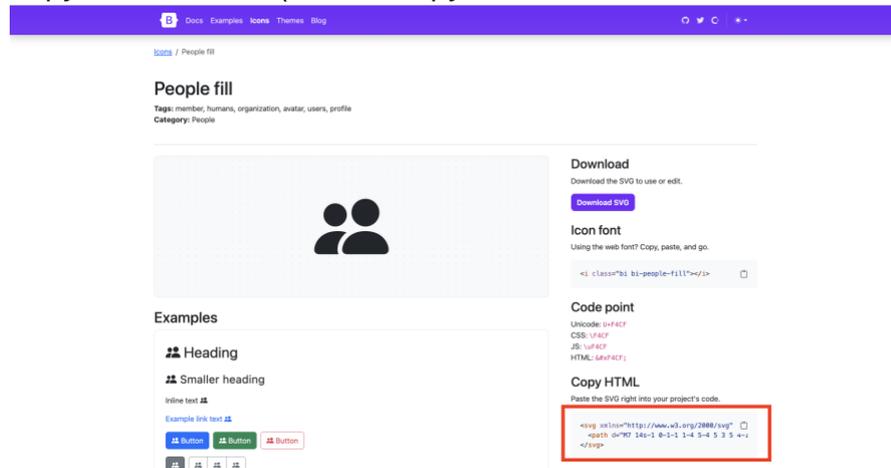
.5h4a.5.5 0 0 1 0 1h-4A.5.5 0 0 1 9 8m1 2.5a.5.5 0 0 1 .5-
.5h3a.5.5 0 0 1 0 1h-3a.5.5 0 0 1-.5-.5'/>
  <path d='M2 2a2 2 0 0 0-2 2v8a2 2 0 0 0 2 2h12a2 2 0 0 0 2-
2V4a2 2 0 0 0-2-2zM1 4a1 1 0 0 1 1-1h12a1 1 0 0 1 1 1v8a1 1 0 0
1-1 1H8.96q.04-.245.04-.5C9 10.567 7.21 9 5 9c-2.086 0-3.8 1.398-
3.984 3.181A1 1 0 0 1 1 12z'/'>
</svg>"
  },
  {
    IconName: "Team_Reports_Filled",
    IconData: "<svg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' fill='currentColor' class='bi bi-people-
fill' viewBox='0 0 16 16'>
  <path d='M7 14s-1 0-1-1 1-4 5-4 5 3 5 4-1 1-1 1zm4-6a3 3 0 1 0
0-6 3 3 0 0 0 0 6m-5.784 6A2.24 2.24 0 0 1 5 13c0-1.355.68-2.75
1.936-3.72A6.3 6.3 0 0 0 5 9c-4 0-5 3-5 4s1 1 1 1zM4.5 8a2.5 2.5
0 1 0 0-5 2.5 2.5 0 0 0 0 5'/'>
</svg>"
  },
  {
    IconName: "Team_Reports",
    IconData: "<svg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' fill='currentColor' class='bi bi-people'
viewBox='0 0 16 16'>
  <path d='M15 14s1 0 1-1-1-4-5-4-5 3-5 4 1 1 1 1zm-7.978-1L7
12.996c.001-.264.167-1.03.76-1.72C8.312 10.629 9.282 10 11
10c1.717 0 2.687.63 3.24 1.276.593.69.758 1.457.76 1.721-
.008.002-.014.002zM11 7a2 2 0 1 0 0-4 2 2 0 0 0 0 4m3-2a3 3 0 1
1-6 0 3 3 0 0 1 6 0M6.936 9.28a6 6 0 0 0-1.23-.247A7 7 0 0 0 5
9c-4 0-5 3-5 4q0 1 1 1h4.216A2.24 2.24 0 0 1 5 13c0-1.01.377-
2.042 1.09-2.904.243-.294.526-.569.846-.816M4.92 10A5.5 5.5 0 0 0
4 13H1c0-.26.164-1.03.76-1.724.545-.636 1.492-1.256 3.16-
1.275ZM1.5 5.5a3 3 0 1 1 6 0 3 3 0 0 1-6 0m3-2a2 2 0 1 0 0 4 2 2
0 0 0 0-4'/'>
</svg>"
  }
});

```

Additional: How to work with SVG in Power Apps in general?

In simple terms, SVG is a text-based code that defines an image. It gets rendered on the app within the Image control into an actual image.

On the [Bootstrap website](#), you can choose the icon that you want to use and copy the SVG code (under “Copy HTML” as shown on screenshot below).



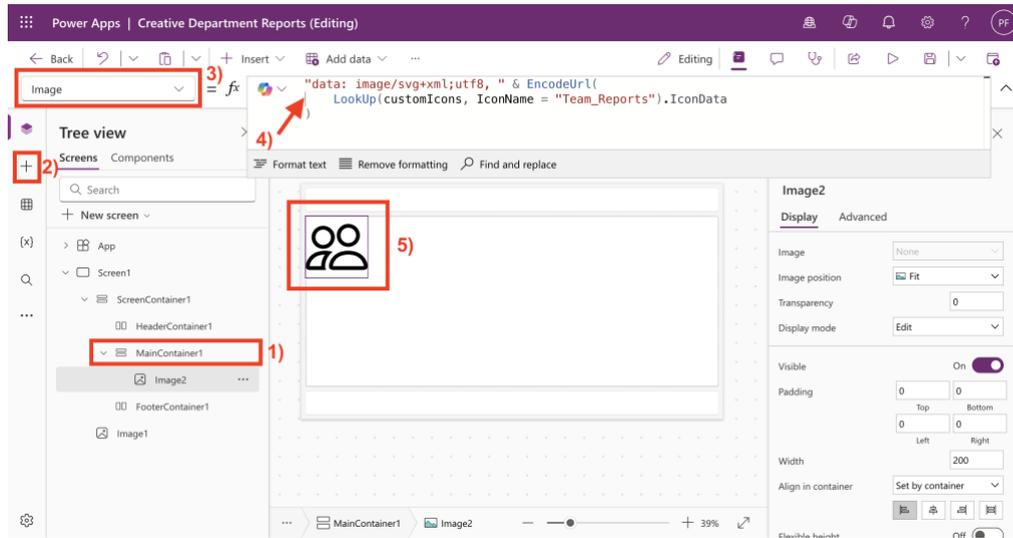
Note: Since SVG code is inserted into Power App formulas as a text in double quotes, before pasting the SVG code into Power Apps, **replace the internal double quotes with single quotes** within the SVG code, to avoid any errors.

If you want to get the image from the SVG code, you can generally follow these steps:

- 1) Select the container where you would like to insert the Image control
- 2) Press on “+” (or on “+ Insert” in the top bar) and type “Image” to insert the Image control. It will appear on the screen.
- 3) Select Image property from the drop-down list.
- 4) Paste the formula below into the formula bar. This formula is used to display an SVG icon in Power Apps by converting it into a format that an Image control can understand.

```
"data: image/svg+xml;utf8, " & EncodeUrl(  
  
    "<svg xmlns='http://www.w3.org/2000/svg' width='16' height='16'  
fill='currentColor' class='bi bi-people' viewBox='0 0 16 16'>  
  <path d='M15 14s1 0 1-1-1-4-5-4-5 3-5 4 1 1 1 1zm-7.978-1L7 12.996c.001-  
.264.167-1.03.76-1.72C8.312 10.629 9.282 10 11 10c1.717 0 2.687.63 3.24  
1.276.593.69.758 1.457.76 1.721-.008.002-.014.002zM11 7a2 2 0 1 0 0-4 2 2 0 0 0  
4m3-2a3 3 0 1 1-6 0 3 3 0 0 1 6 0M6.936 9.28a6 6 0 0 0-1.23-.247A7 7 0 0 0 5 9c-4  
0-5 3-5 4q0 1 1h4.216A2.24 2.24 0 0 1 5 13c0-1.01.377-2.042 1.09-2.904.243-  
.294.526-.569.846-.816M4.92 10A5.5 5.5 0 0 4 13H1c0-.26.164-1.03.76-1.724.545-  
.636 1.492-1.256 3.16-1.275ZM1.5 5.5a3 3 0 1 1 6 0 3 3 0 0 1-6 0m3-2a2 2 0 1 0 0  
4 2 2 0 0 0 0-4' />  
</svg>"  
)
```

5) The icon is displayed within the Image control (in this example, image control is called “Image2” under the “Tree View”).



Building a Navigation Menu Component

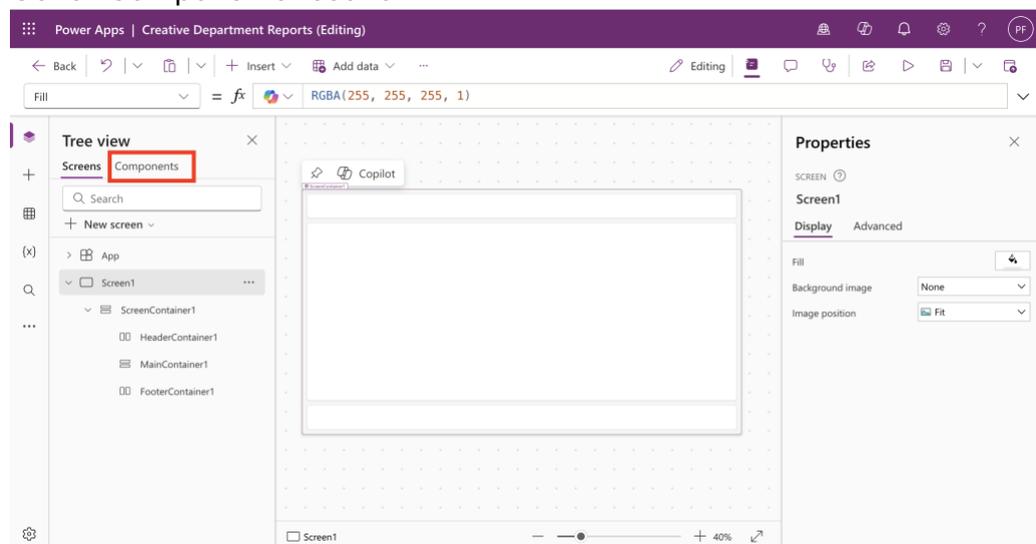
A **component** in Power Apps is a **reusable UI element** that can be added on multiple screens within an app.

It helps maintain consistency by ensuring the uniform look and feel, reduces duplication of work and is easily maintained as it can be updated in a single place and changes will apply everywhere it is used.

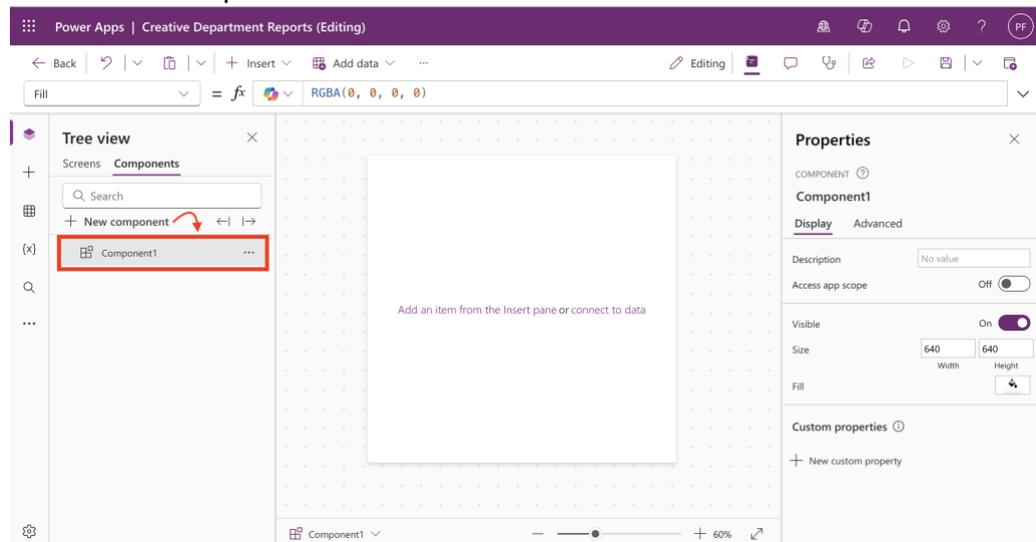
Steps – #1 Creating Compact Navigation Menu component :

Creating a new component

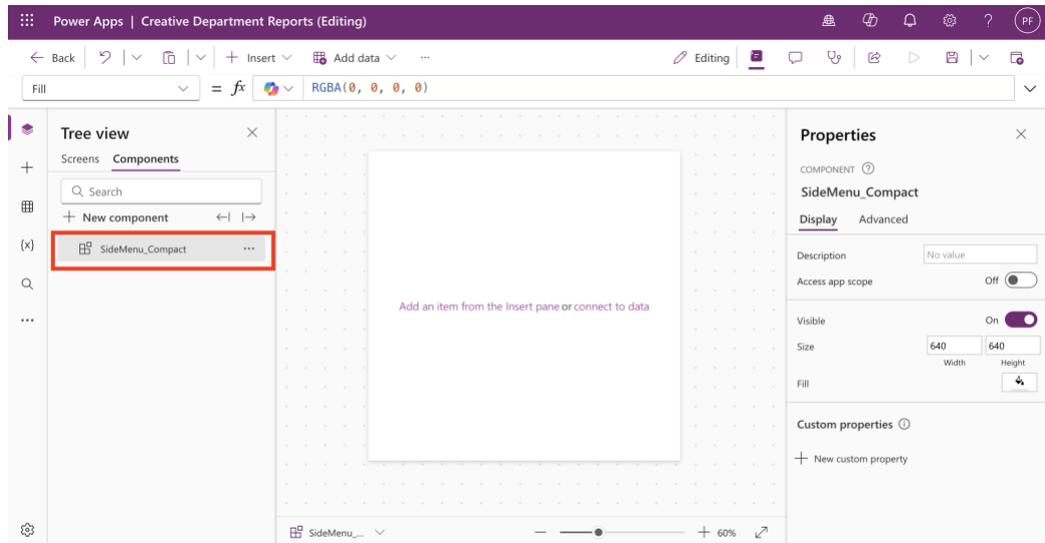
- Go to “Components” section:



- Add a new component:



- Rename accordingly (generally, use names that reflect the meaning of component/variable well, yet they should be concise, as a good practice). Let's call it "SideMenu_Compact":



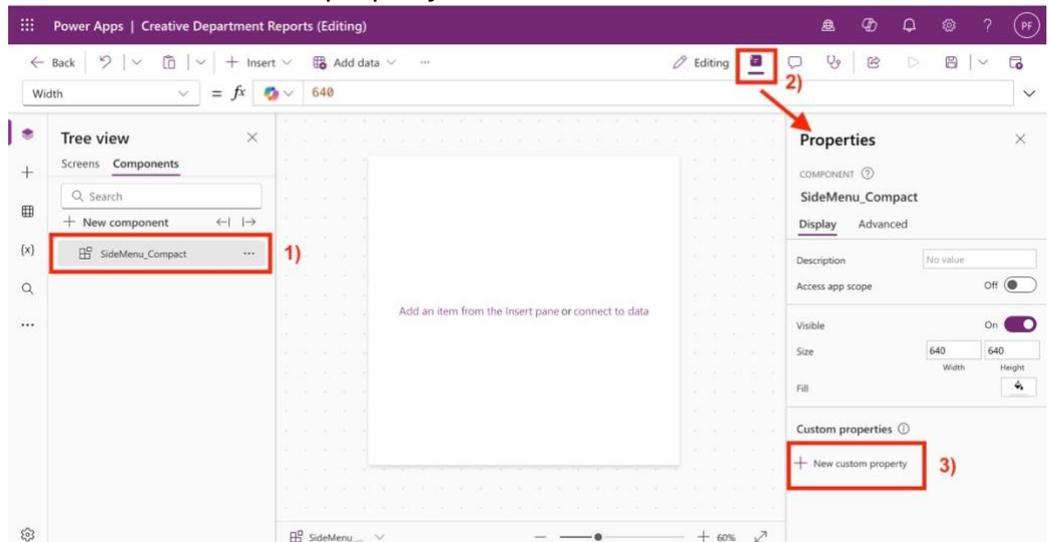
Custom properties

Custom properties in Power Apps canvas components allow to relay different values or logic between the component and the app that is hosting the component. [3][4]

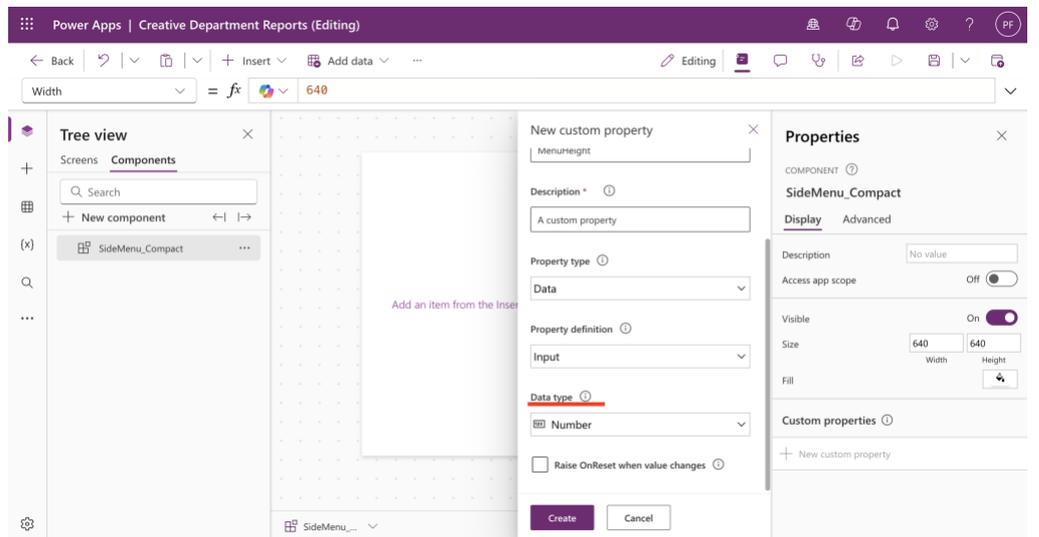
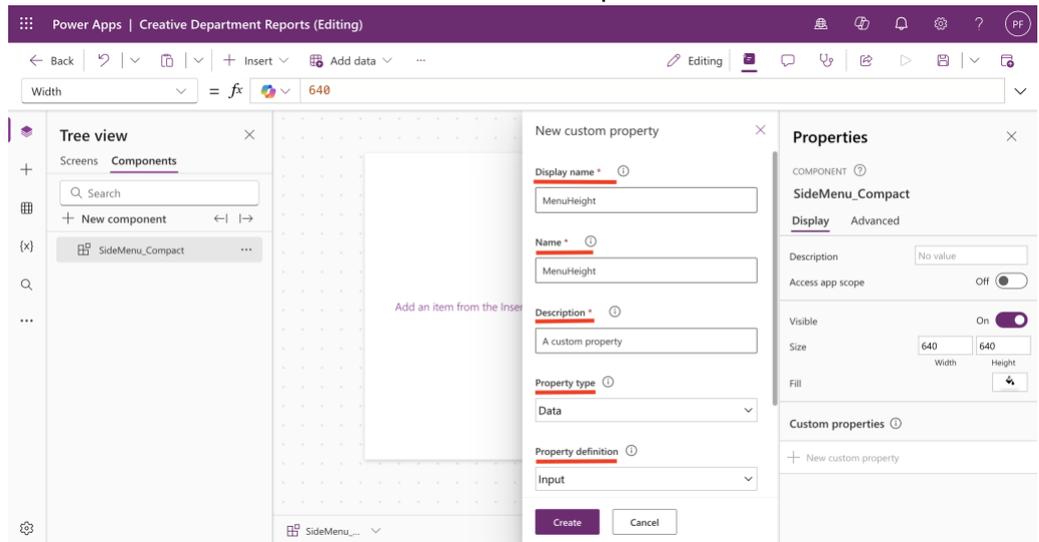
Using custom properties is considered a best practice since they provide easier maintainability and flexibility by being a centralized logic within the component.

For example, for the background colour custom property – the colour value can be changed once in a single place if needed, i.e., within this property, and this change will be applied to all the fields (e.g. of a field is an element such as button can have field called “width”, “height”, “colour” etc.) for which the property is used automatically, instead of going and changing the colour in all fields manually.

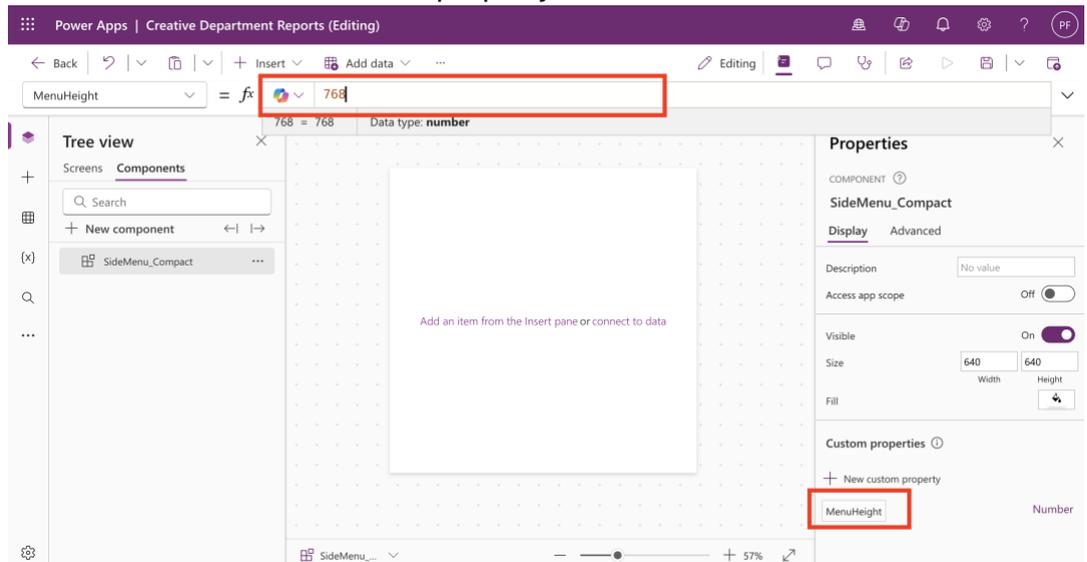
- Adding a new custom property for the “SideMenu_Compact” component:
 - 1) Ensure you have selected the SideMenu_Compact under the “Tree View”
 - 2) Open the “Properties” panel on the right - hand side
 - 3) Click on “New custom property”



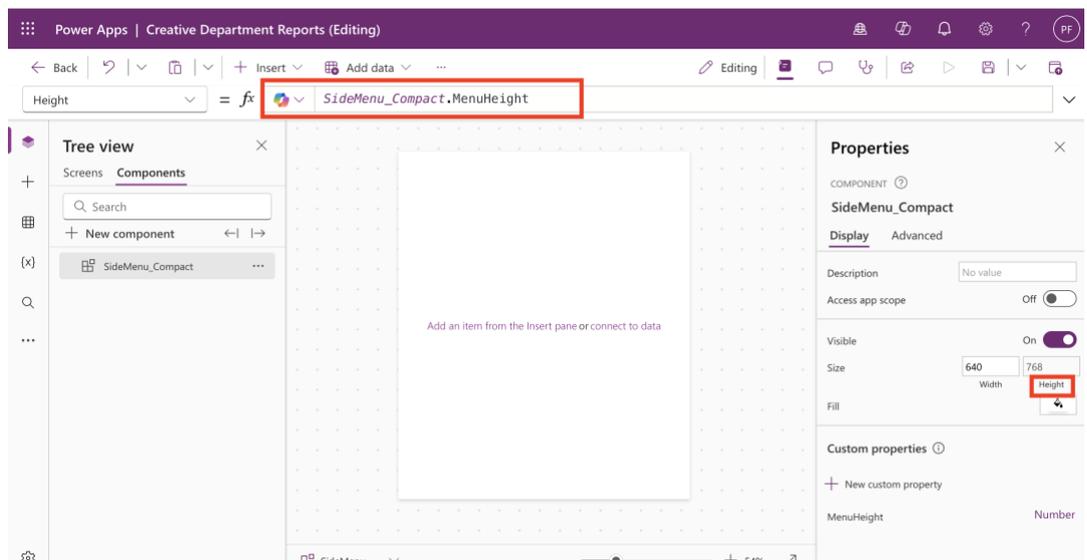
- 4) Set the fields of the new custom property called MenuHeight as shown on the screenshots below and then press “Create”:



- Press on the newly created property “MenuHeight” and in the formula bar enter the default value for this property of 768:

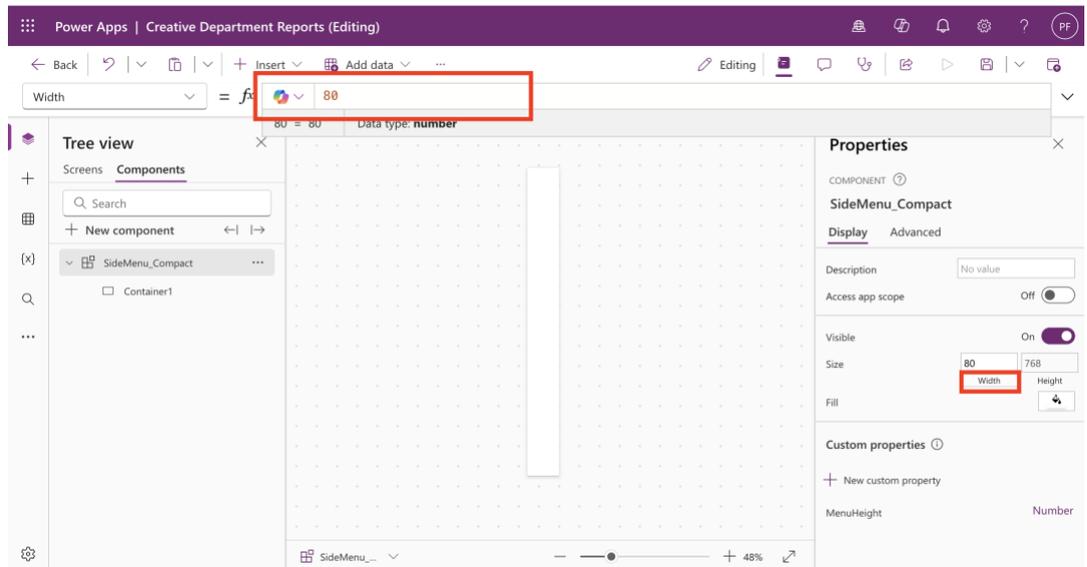


- Press on “Height” field in the Properties panel and then enter the following formula to the formula bar: *SideMenu_Compact*.MenuHeight (Where *SideMenu_Compact* is the name of the component and MenuHeight is the name of the property we have just created.)



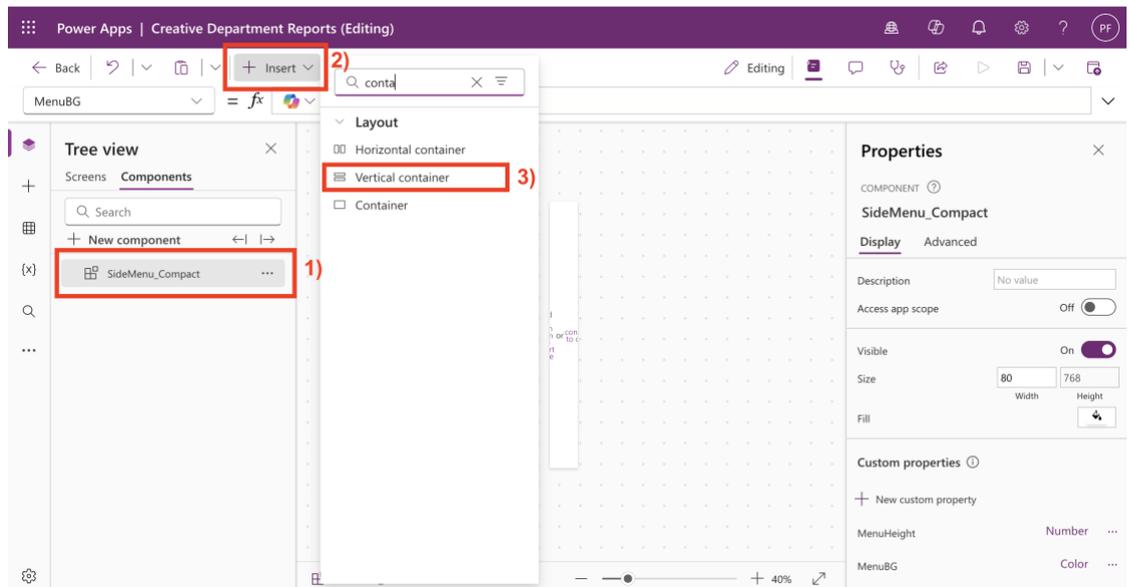
Continuing to create the Navigation Menu Component

- Press on “Width” field in the Properties panel and then set the width to 80:

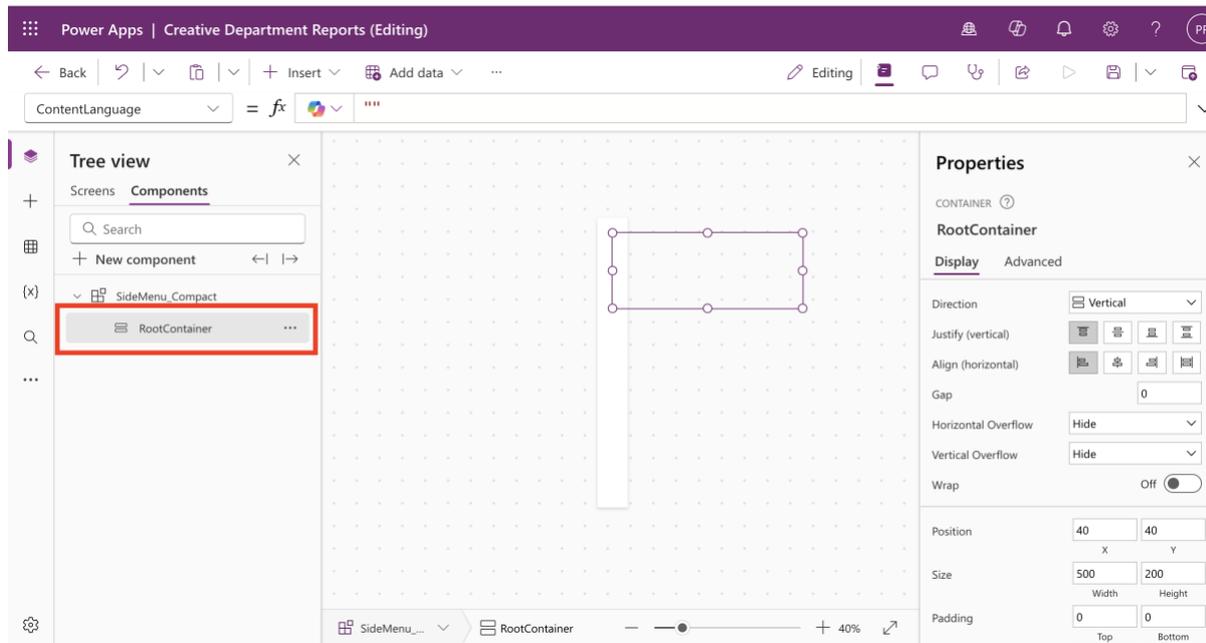


- Let's add a vertical container as the root container for the navigation menu component. The root container will group all elements, such as buttons, ensuring they stay organized and responsive. [5]

- 1) Select the component
- 2) Press “+ Insert”
- 3) Search for “Container” and select “Vertical container”



- Rename the container to “RootContainer”

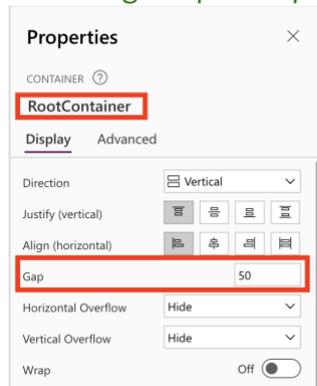


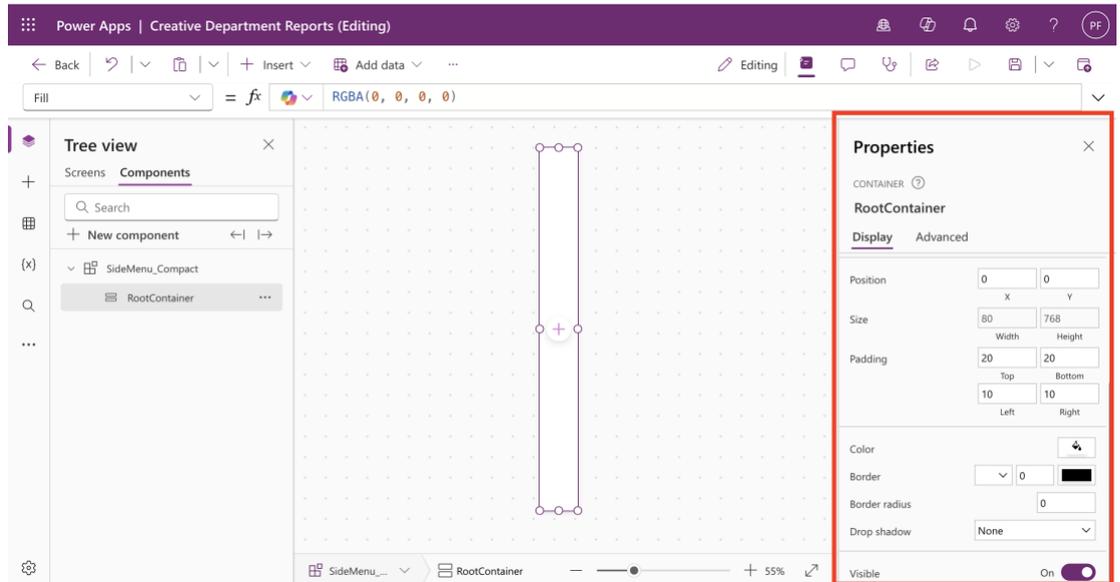
- Now set the properties of the RootContainer:

- X = 0
- Y = 0
- Width = Parent.Width
- Height = Parent.Height
- Top padding = 20
- Bottom padding = 20
- Left padding = 10
- Right padding = 10
- Border = select “None” from the drop-down list
- Border radius = 0
- Drop shadow = select “None” from the drop-down list

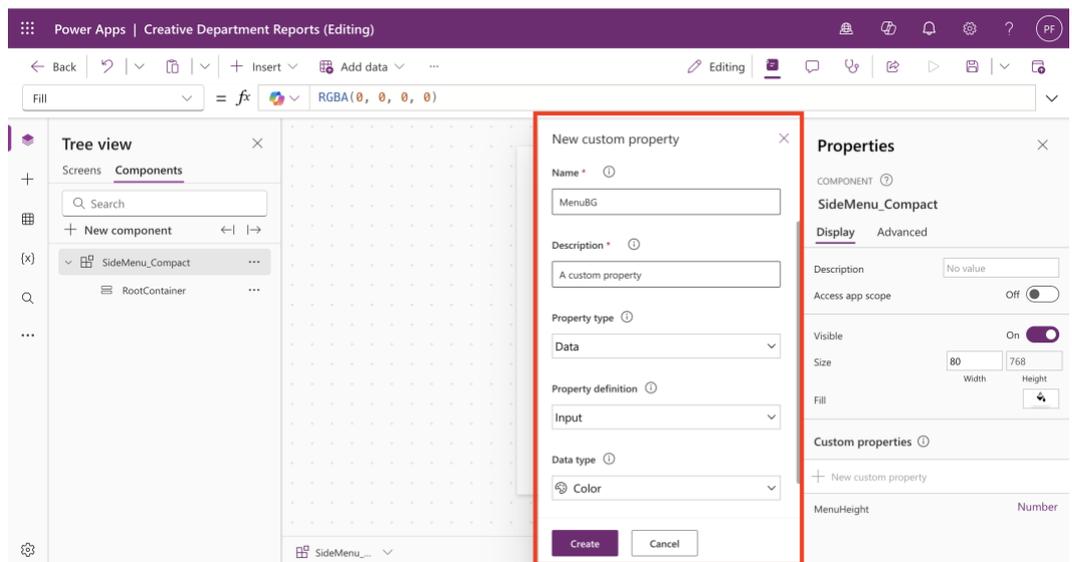
(Leave the properties not mentioned here as they are by default)

Note: set the “Gap” field in the Properties panel to 50 to space elements out within the “RootContainer”. This is not shown on the screenshots up to adding the profile picture under the “UserProfileButton” below.



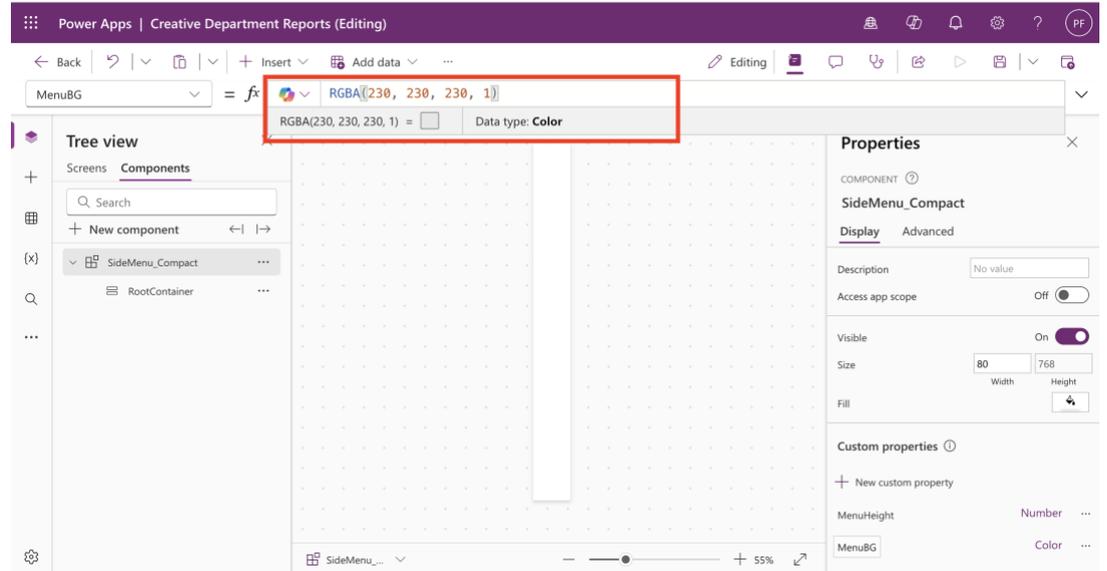


- Create **new custom property for the background colour** of the navigation menu. The Display name and the Name are “MenuBG”.



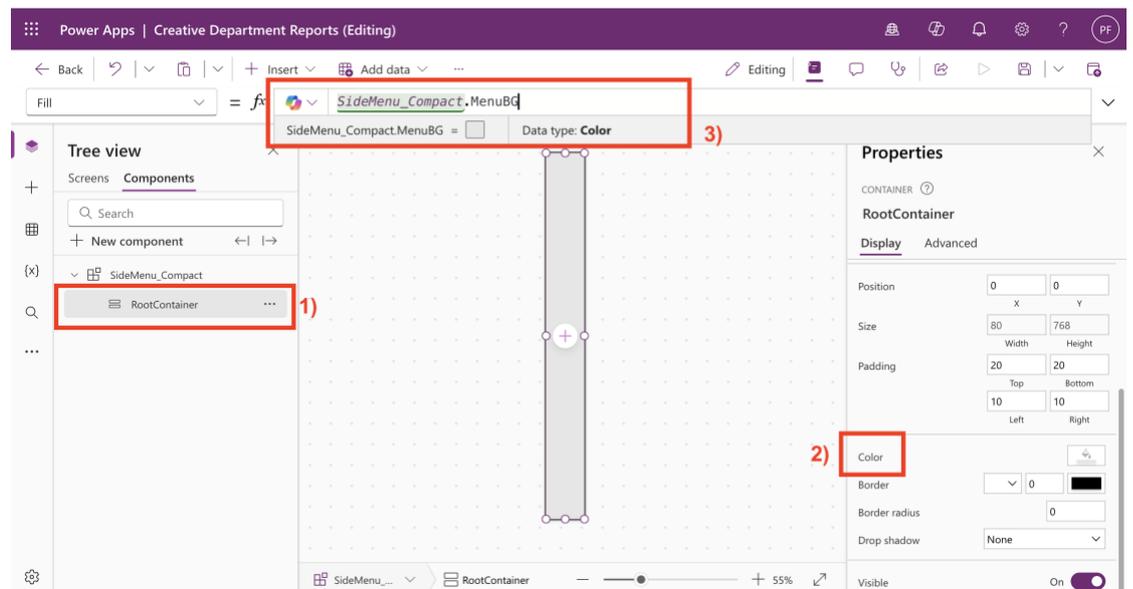
Set the default colour to: `RGBA(230, 230, 230, 1)`

It is a function in Power Apps that defines a colour using the RGBA (Red, Green, Blue, Alpha) format. 1 (Alpha) sets the opacity to 100% (fully opaque).

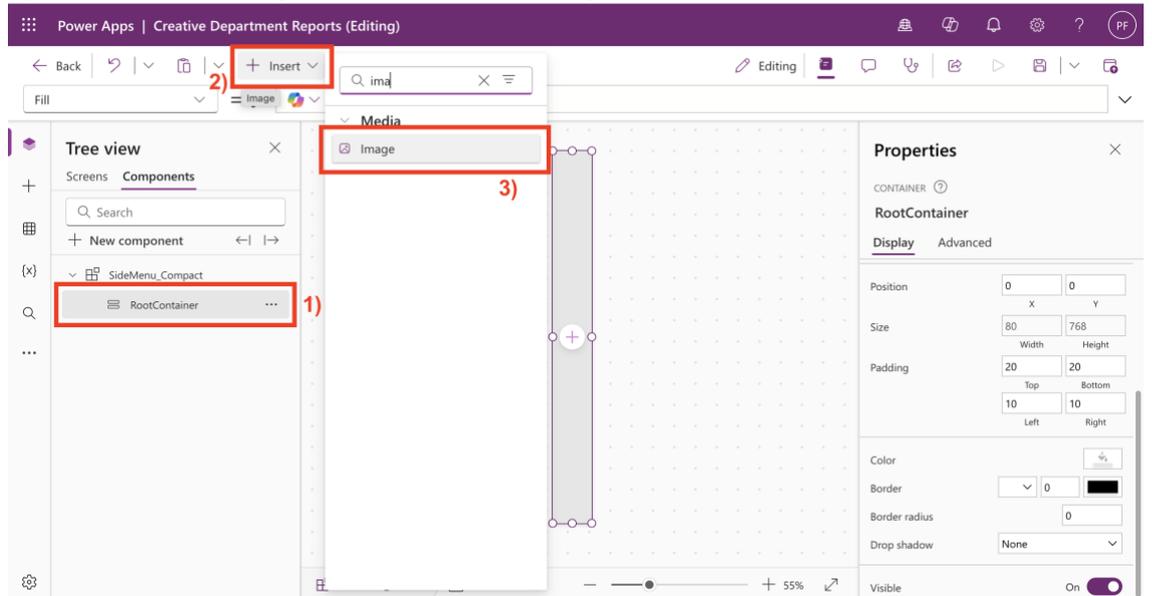


- Let's apply this custom property to the "RootContainer":
 - Select the "RootContainer"
 - Press on "Color" field in the Properties panel
 - Enter the following formula to the formula bar (as you see to the left of the formula bar the property selected is called "Fill"):

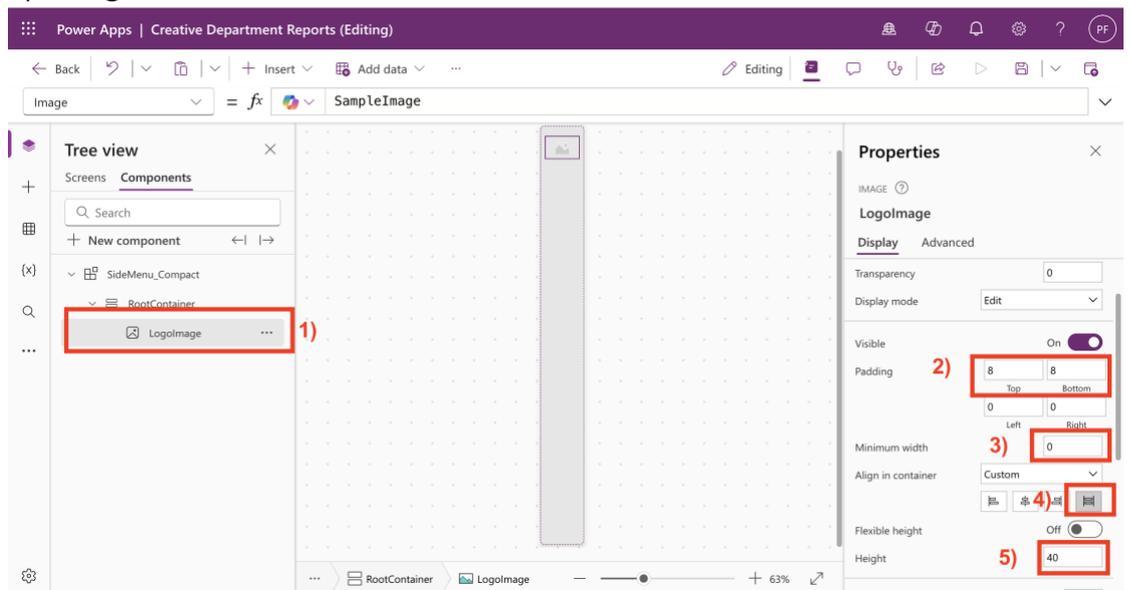
`SideMenu_Compact.MenuBG`



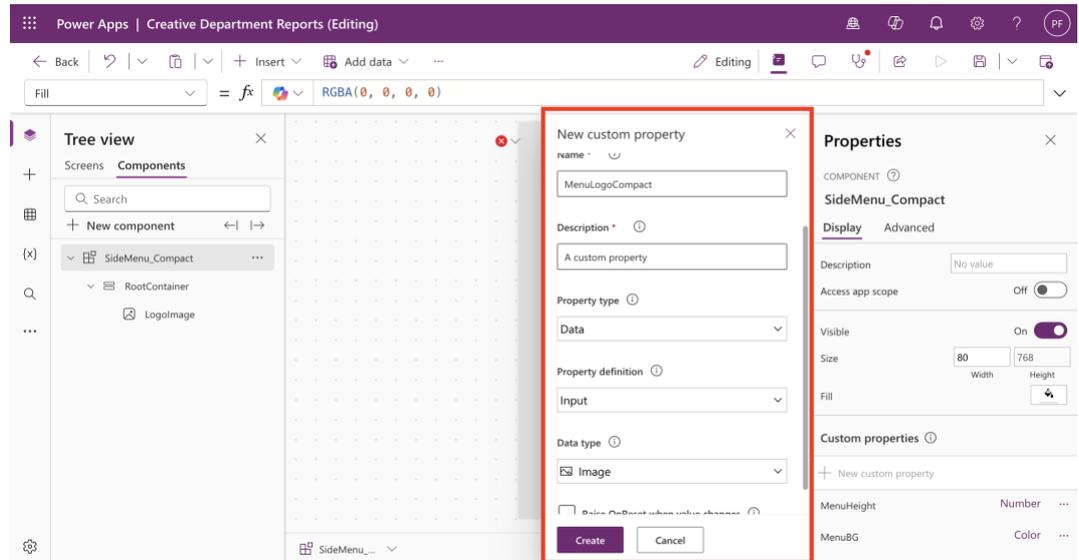
- Add Image control for the menu logo:
 - 1) Select the “RootContainer”
 - 2) Press “+ Insert”
 - 3) Search for “Image” and select the Image control



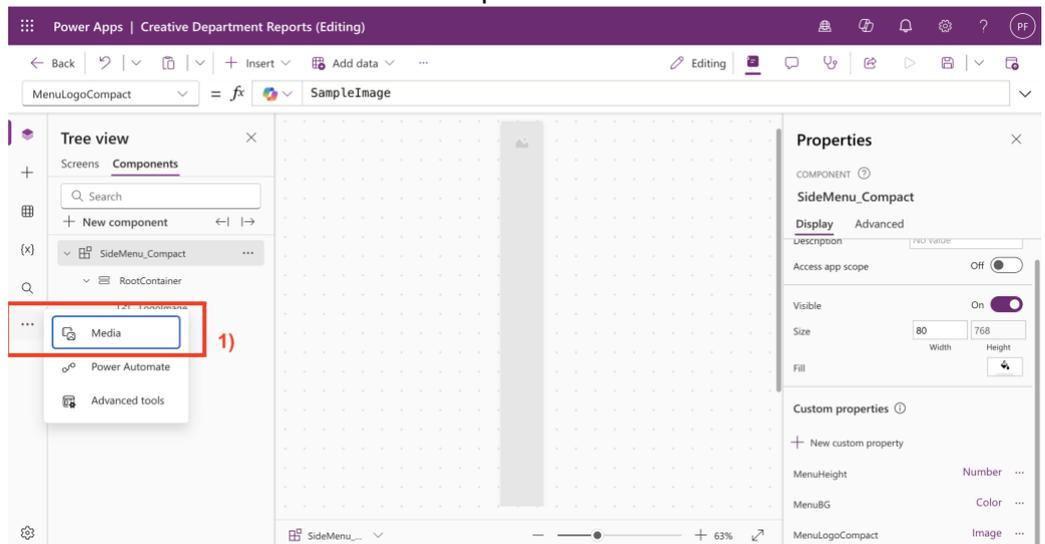
- Name the Image control as “LogoImage” and edit its properties in the Properties panel:
 - 1) Select the “LogoImage” control
 - 2) Top padding = 8
Bottom padding = 8
 - 3) Minimum width = 0
 - 4) Width alignment = stretch
 - 5) Height = 40



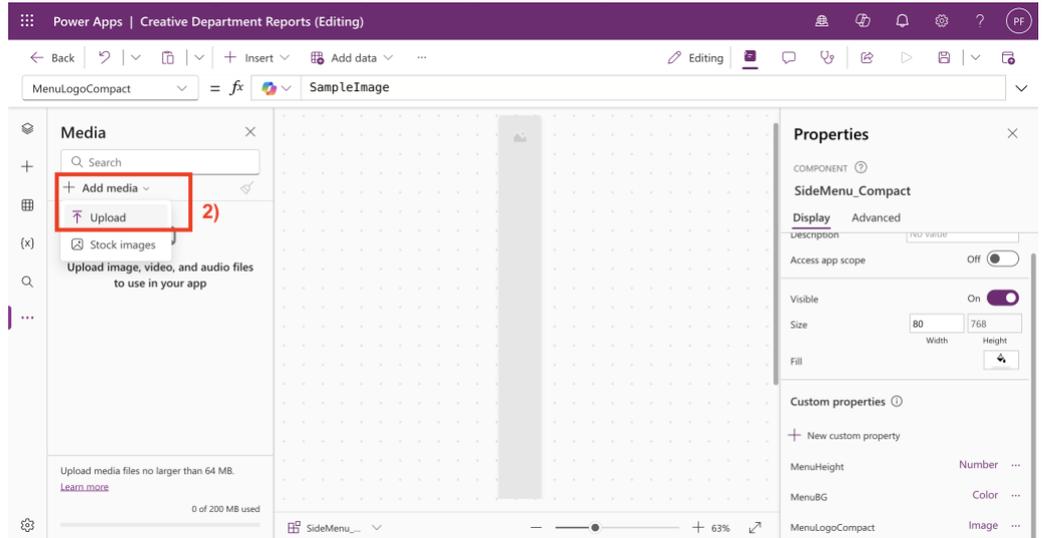
- Create a new custom property called “MenuLogoCompact”, set its Data type to Image:



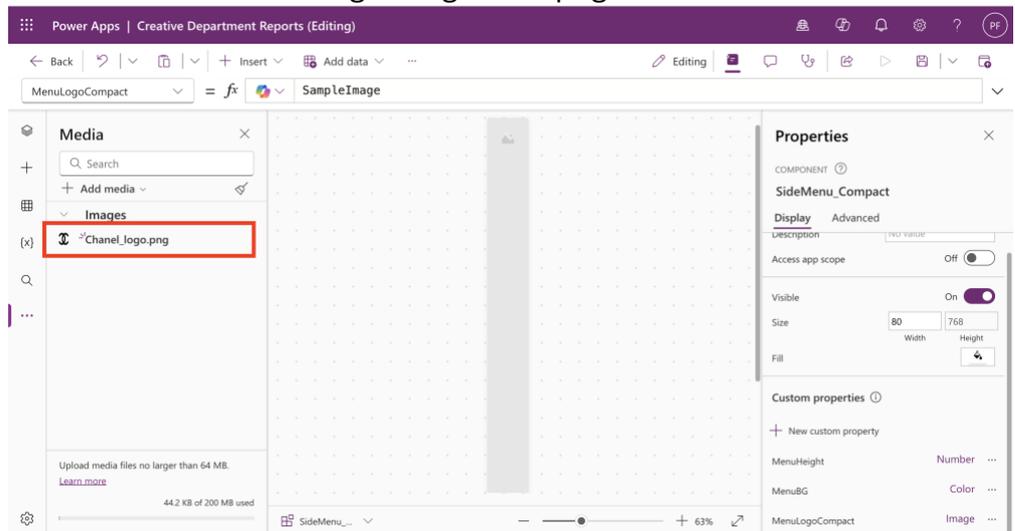
- To set a default image for the newly created custom property “MenuLogoCompact”, we need to add an image to the Power App.
 - 1) Press on the three dots on the left panel and select “Media”



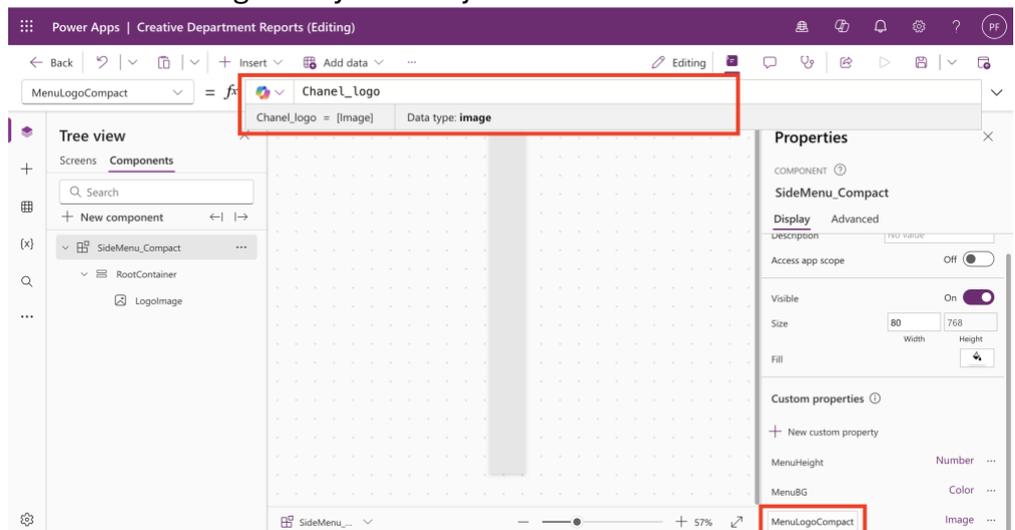
2) Select the image of the logo from your computer files:



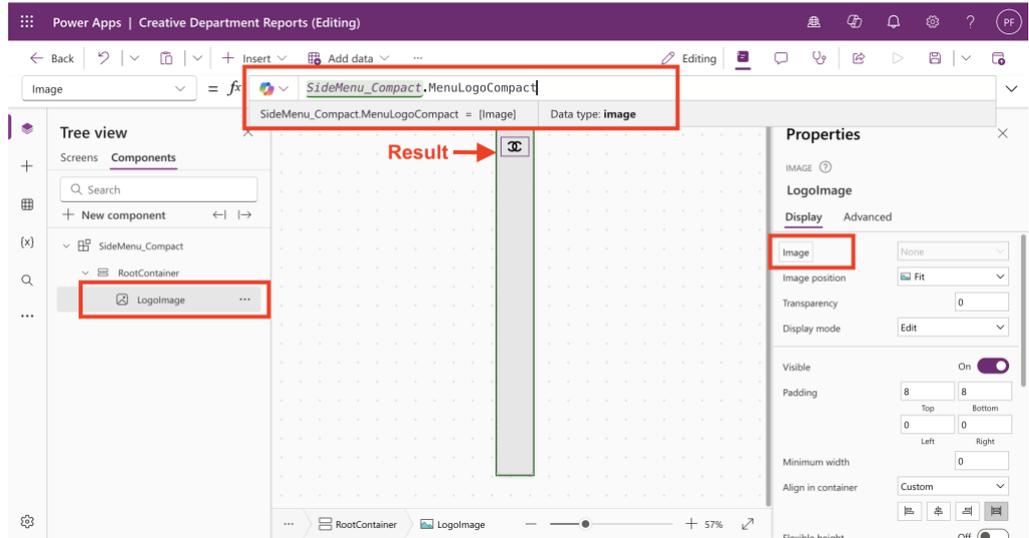
I've added this Chanel logo image with .png extension:



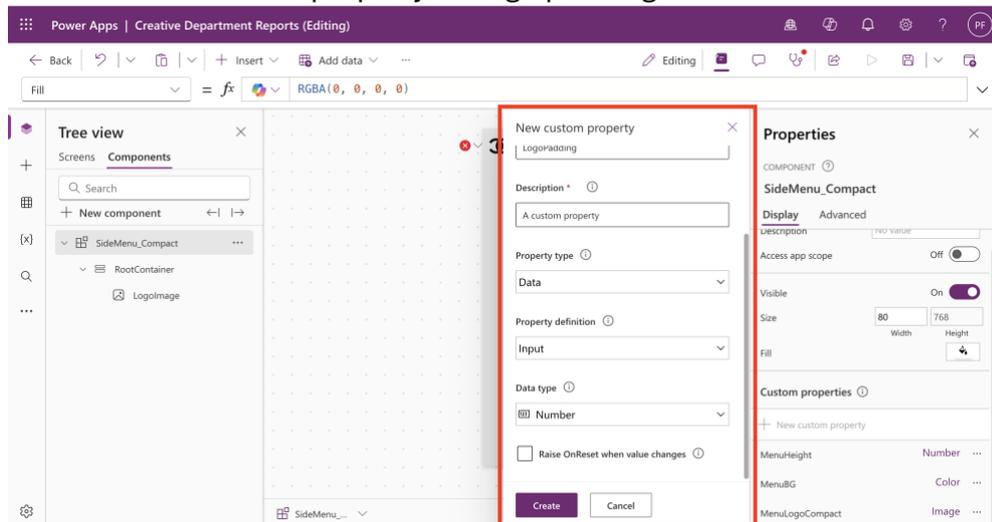
3) Set the default image of the “MenuLogoCompact” property to the name of the image that you have just added to Media:



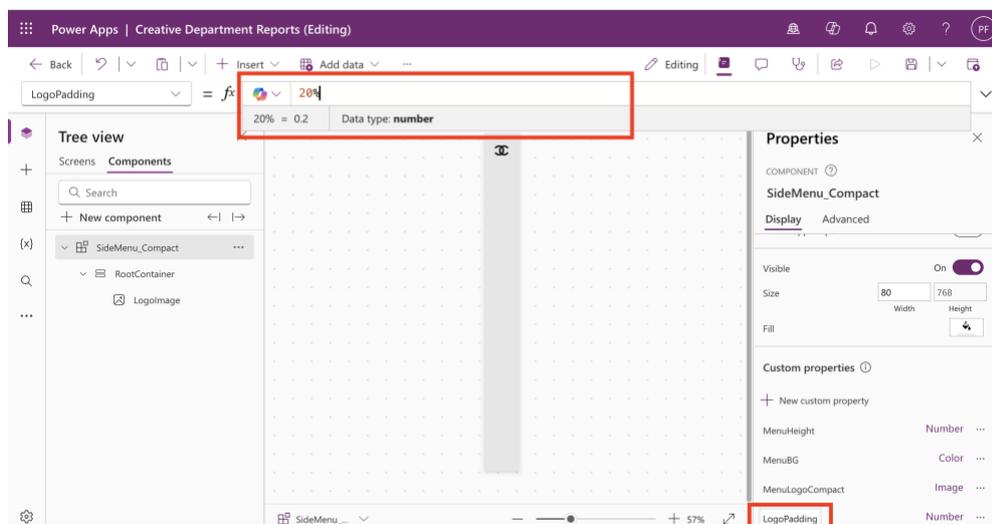
4) Apply the property to the “LogoImage” control:



- Create a new custom property for logo padding:

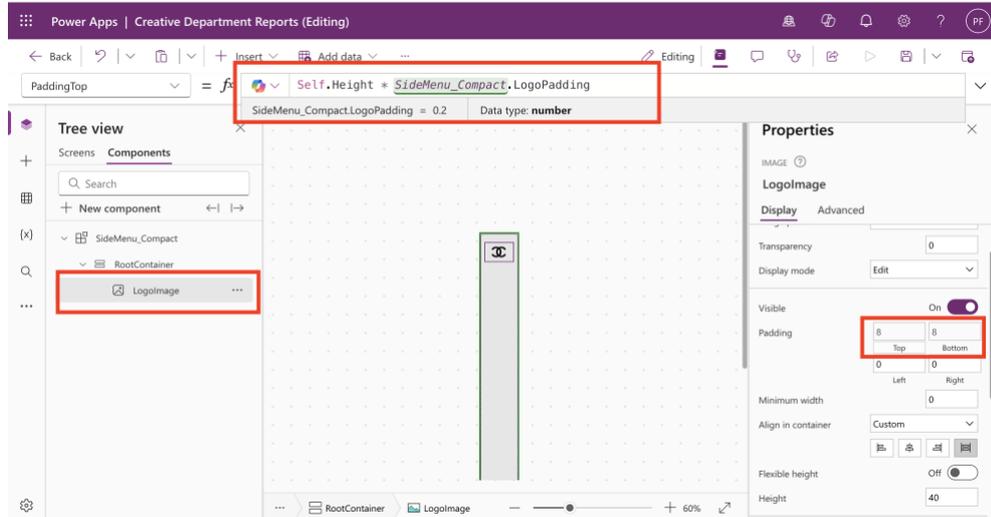


Set the default value to 20%:



Apply the new property to the “LogoImage” control. Set the Top and Bottom padding to the following formula: **Self.Height** *

SideMenu_Compact.LogoPadding

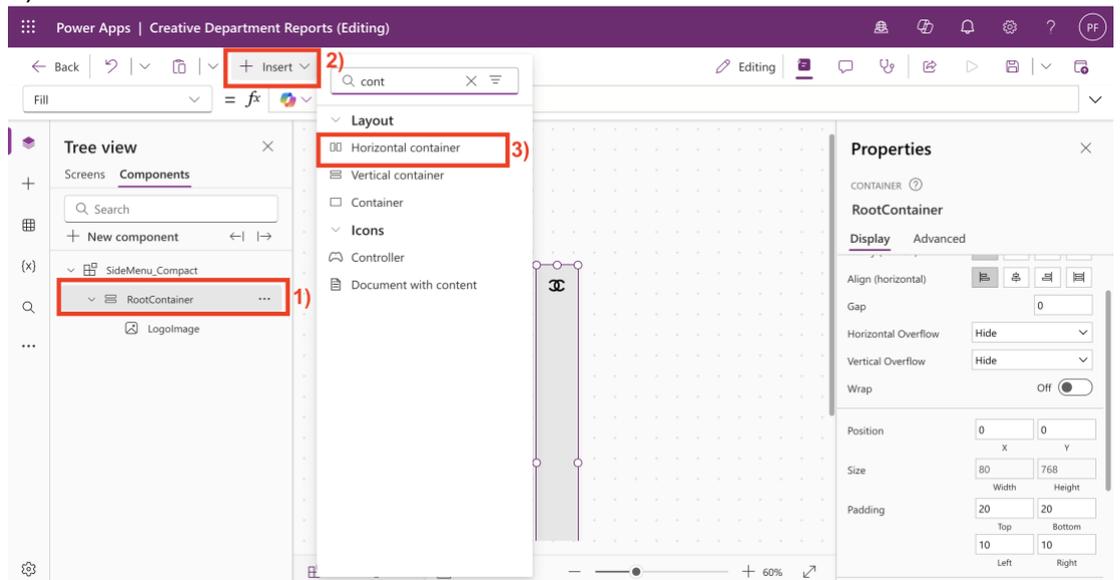


Button to Profile Page + User Profile Image

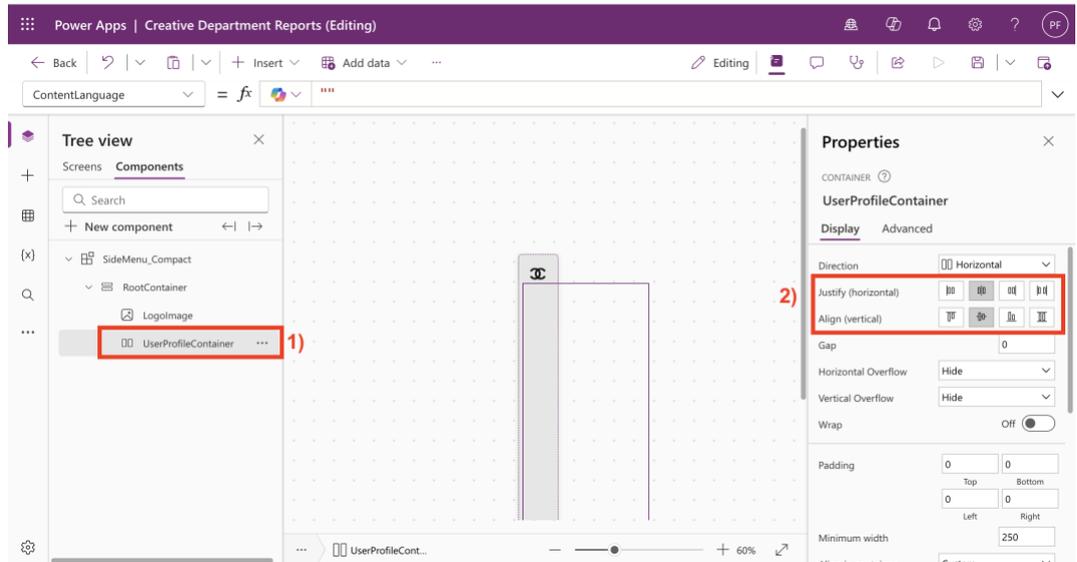
- Let's add the user profile image and button to go to the profile page to the menu.

To store this profile page button component, let's create a horizontal container:

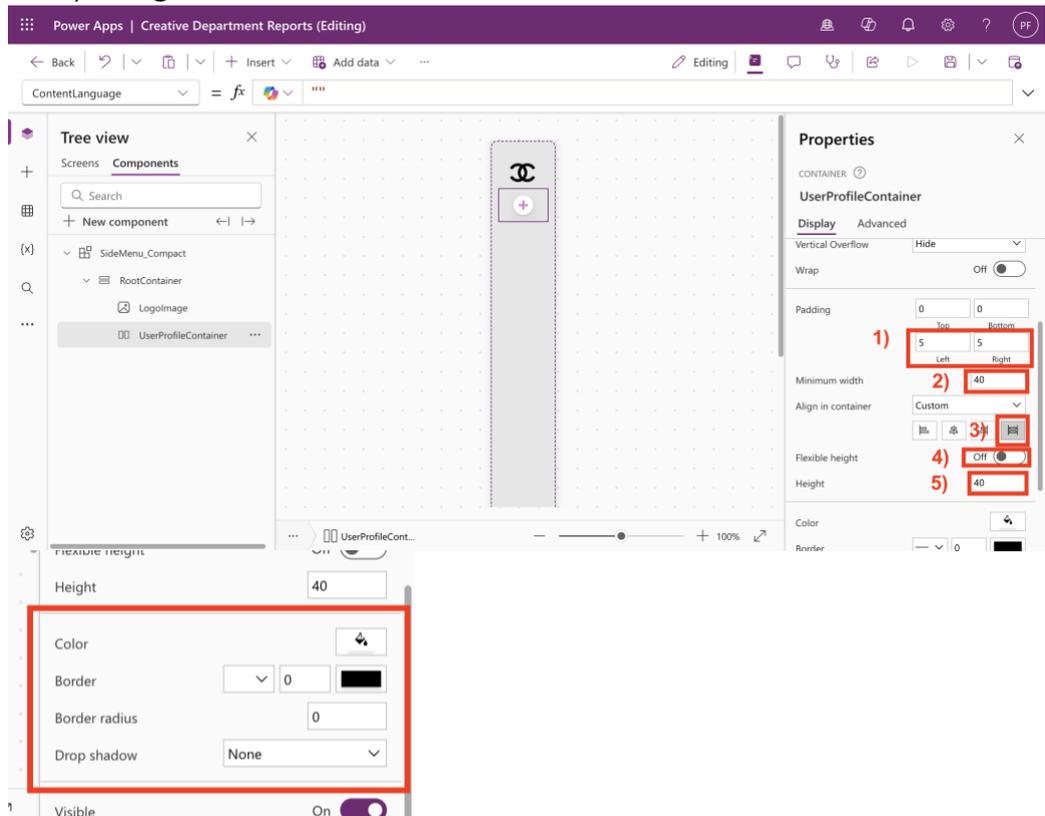
- 1) Select the “RootContainer”
- 2) Press the “+ Insert”
- 3) Choose the “Horizontal container”



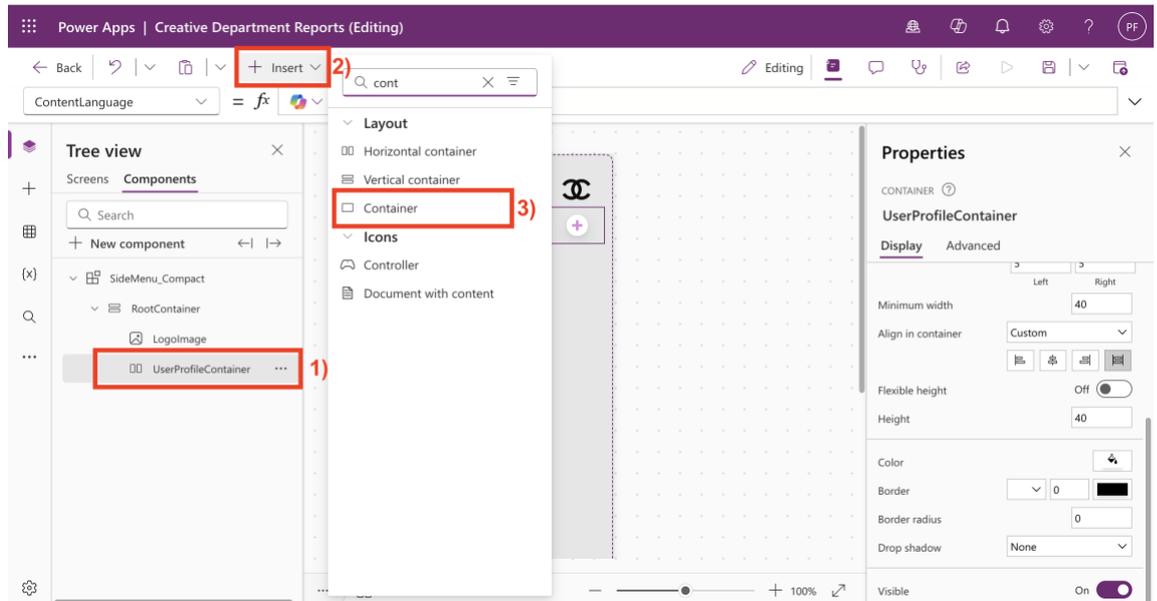
- Now let's adjust the new horizontal container:
 - 1) Rename the container to "UserProfileContainer"
 - 2) Set the Justify(Horizontal) and Align(vertical) fields as shown on the screenshot below:



- 3) Set the following properties:
 - 1) Left padding = 5
Right padding = 5
 - 2) Minimum width = 40
 - 3) Align in container = stretch
 - 4) Turn off flexible height
 - 5) Height = 40



- Add a blank container into the horizontal “UserProfileContainer” and name it “UserContainer”:



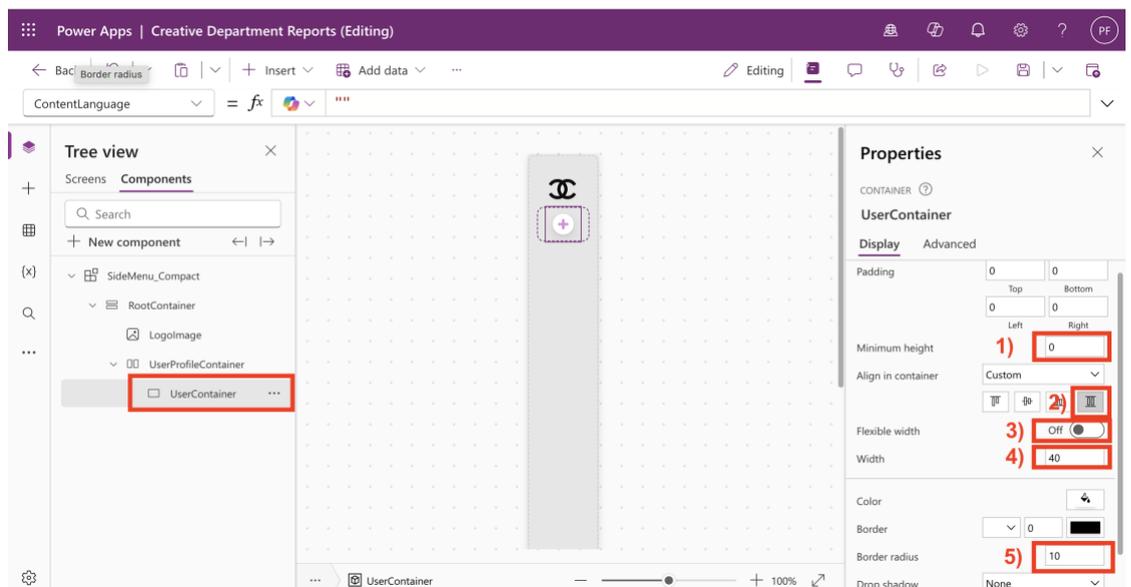
Set its properties:

- 1) Minimum height = 0
- 2) Align in container = stretch
- 3) Turn off flexible width
- 4) Width = 40
- 5) Border radius = 10

And set the:

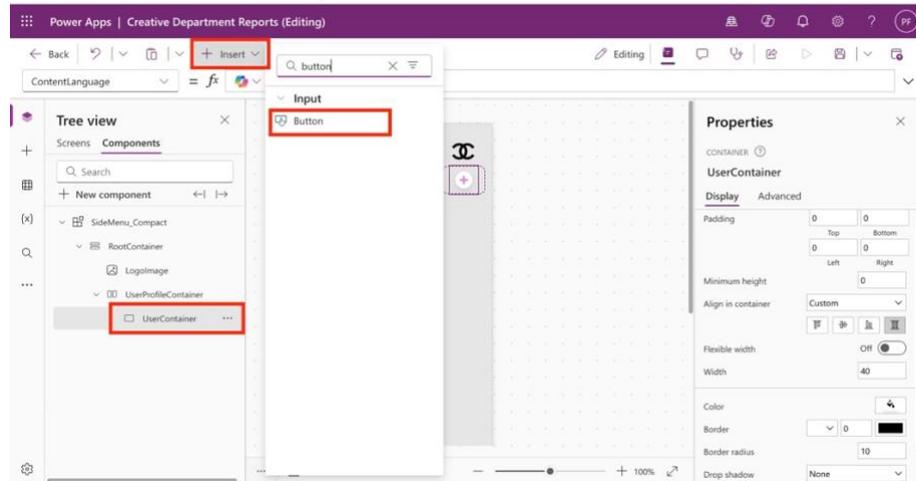
Border = None

Drop shadow = None



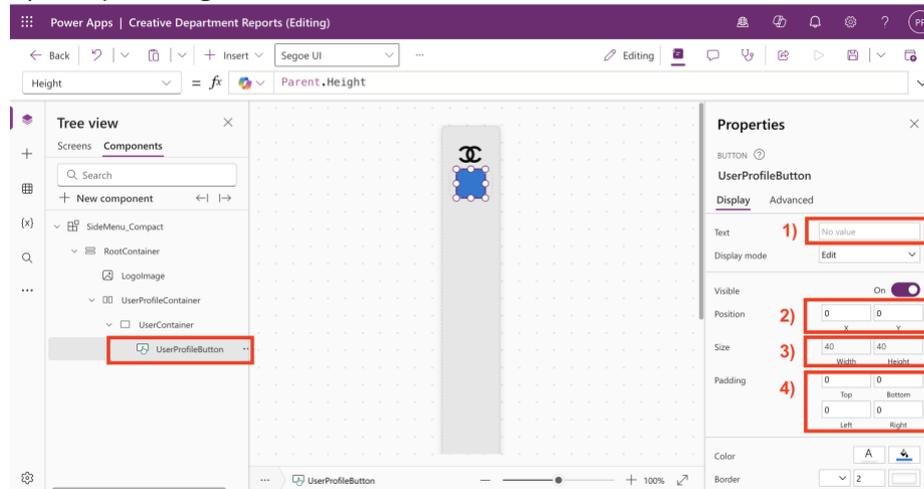
Button for navigation to the Profile Page

- Insert a button to the blank “UserContainer” which will be redirecting users to the Profile Page and name the button “UserProfileButton”:

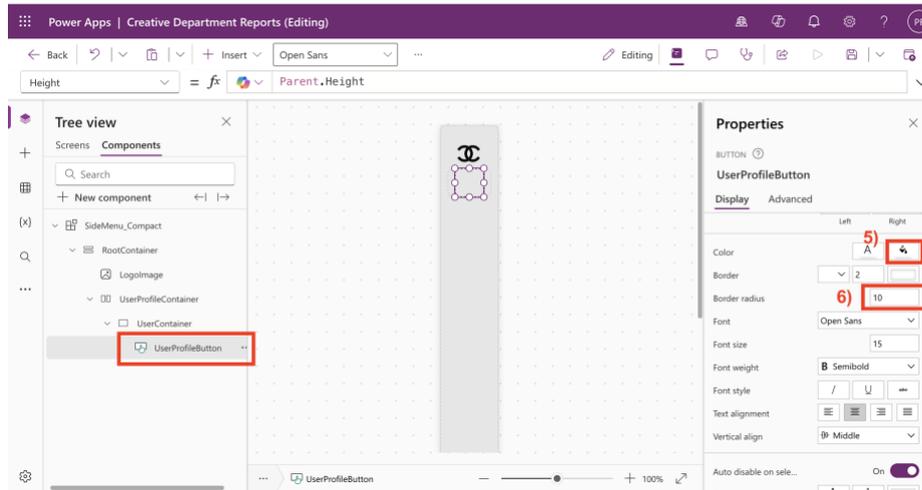


Edit the button's properties:

- 1) Remove text
- 2) X = 0, Y = 0
- 3) Width = Parent.Width, Height = Parent.Height
- 4) All paddings are 0

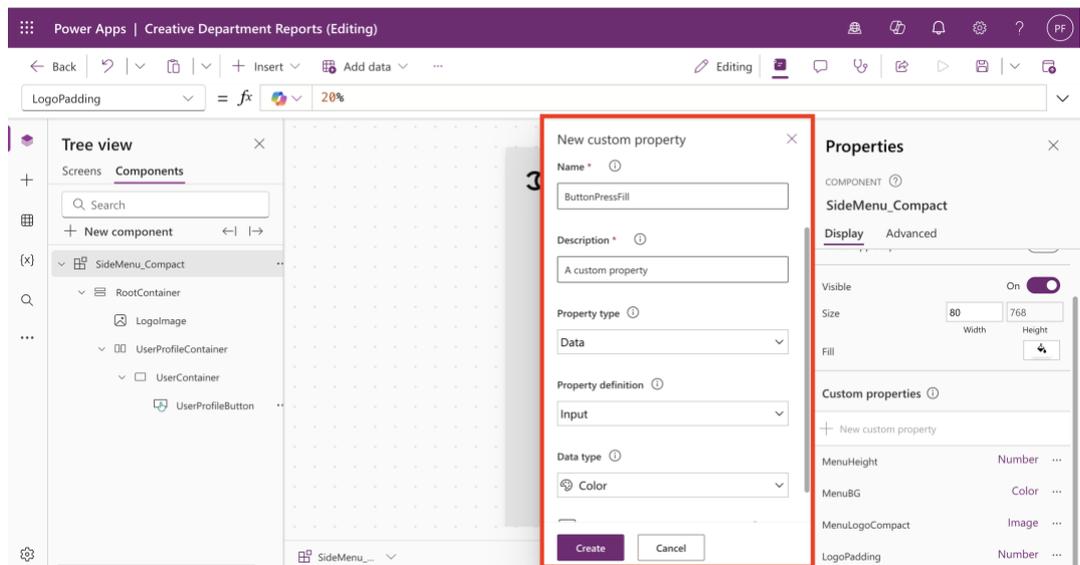


- 5) Set the fill to transparent colour
- 6) Border radius = 10

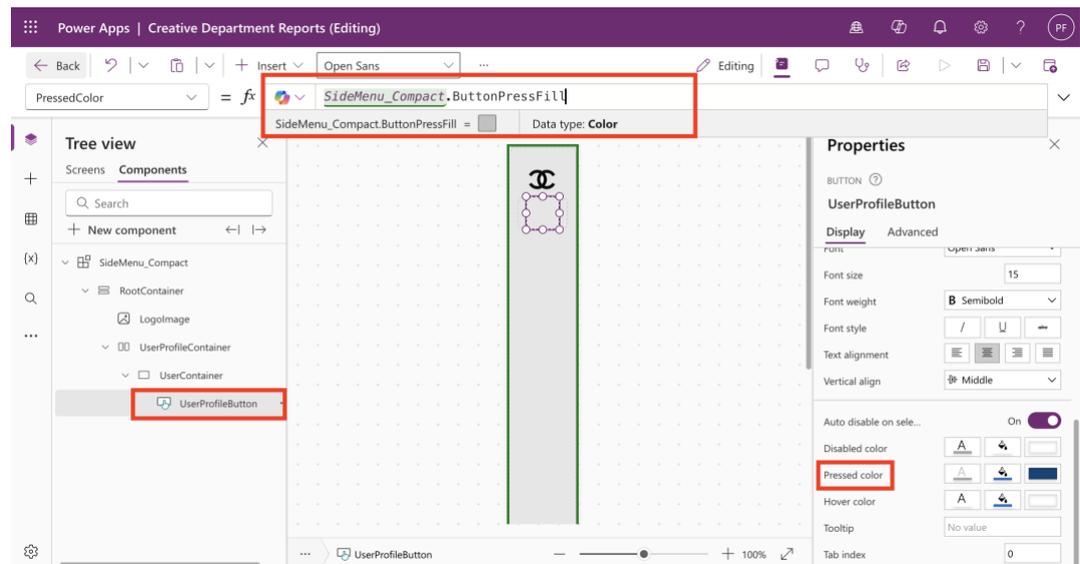


- Let's create custom properties for the "Pressed color" and "Hover color" fields of the "UserProfileButton" (these properties will be used later for other buttons as well).
- Creating "Pressed color" custom property called "ButtonPressFill" of data type "Color".

Set its default colour to: `RGBA(0, 0, 0, 0.2)`

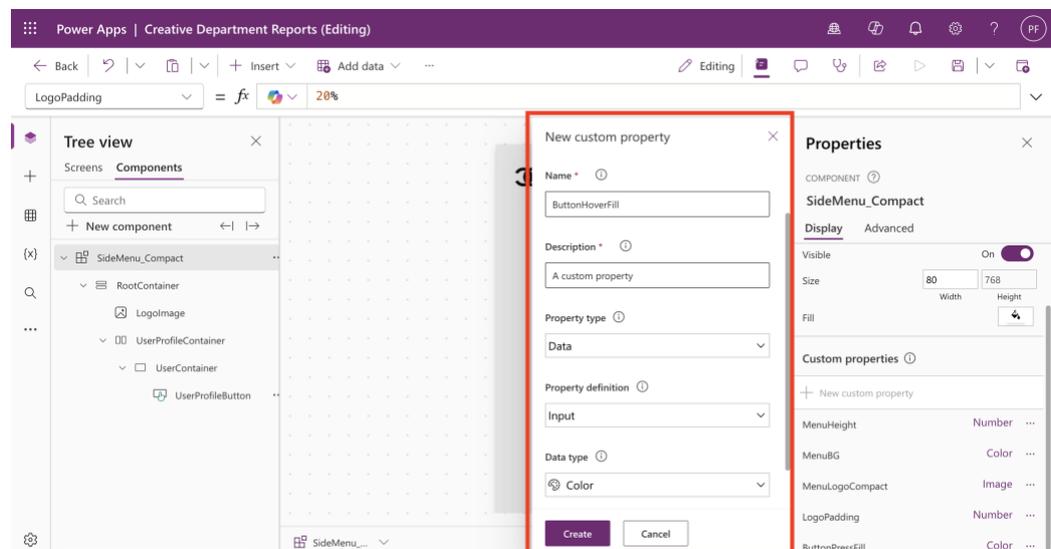


Apply the custom property to the button “Pressed color” field:

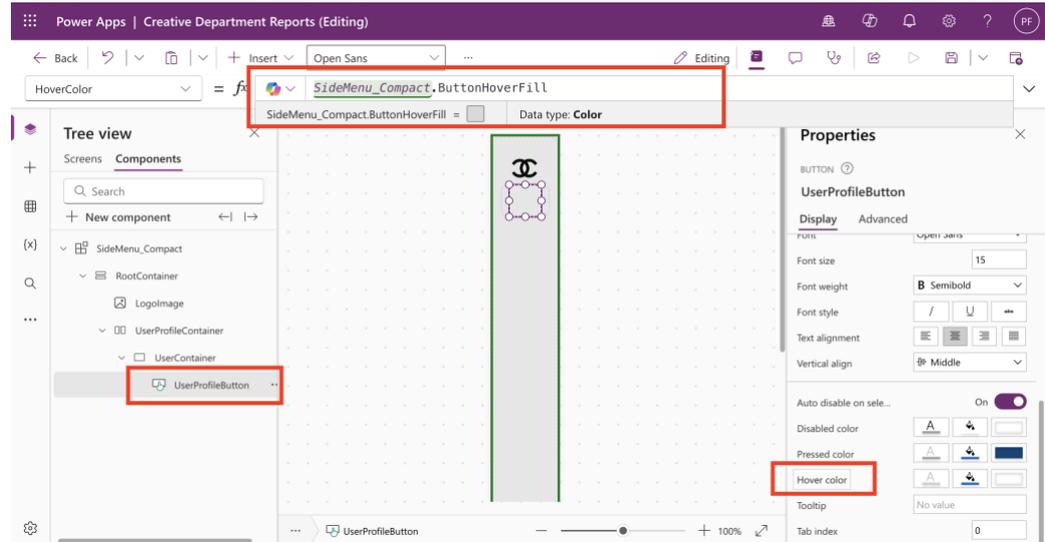


- Creating “Hover color” custom property called “ButtonHoverFill” of data type “Color”.

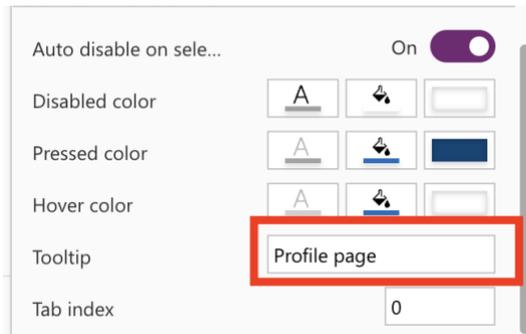
Set its default colour to: `RGBA(0, 0, 0, 0.1)`



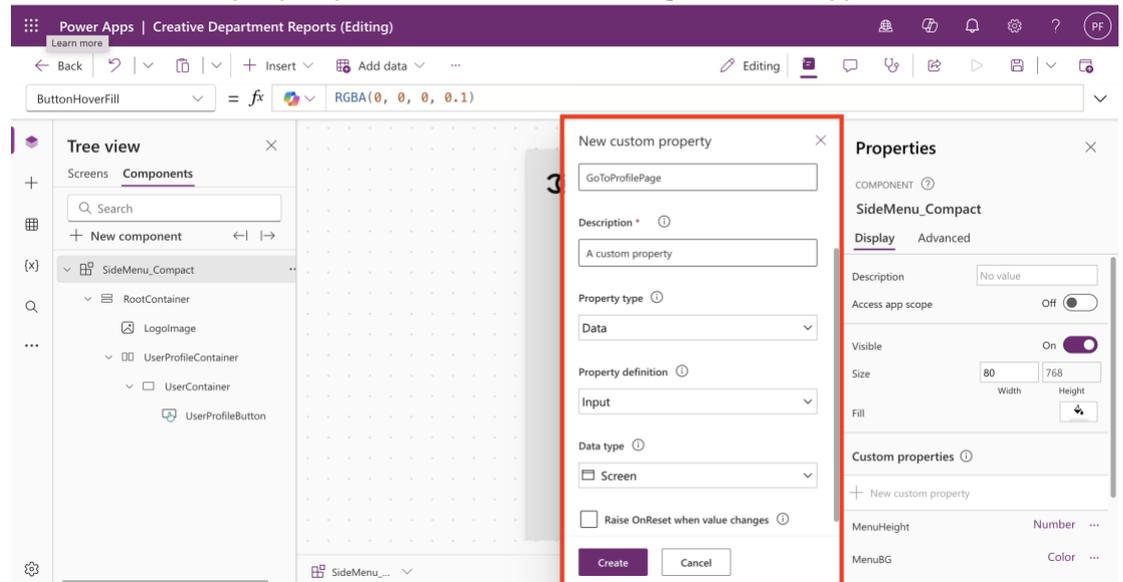
Apply the custom property to the button “Hover color” field:



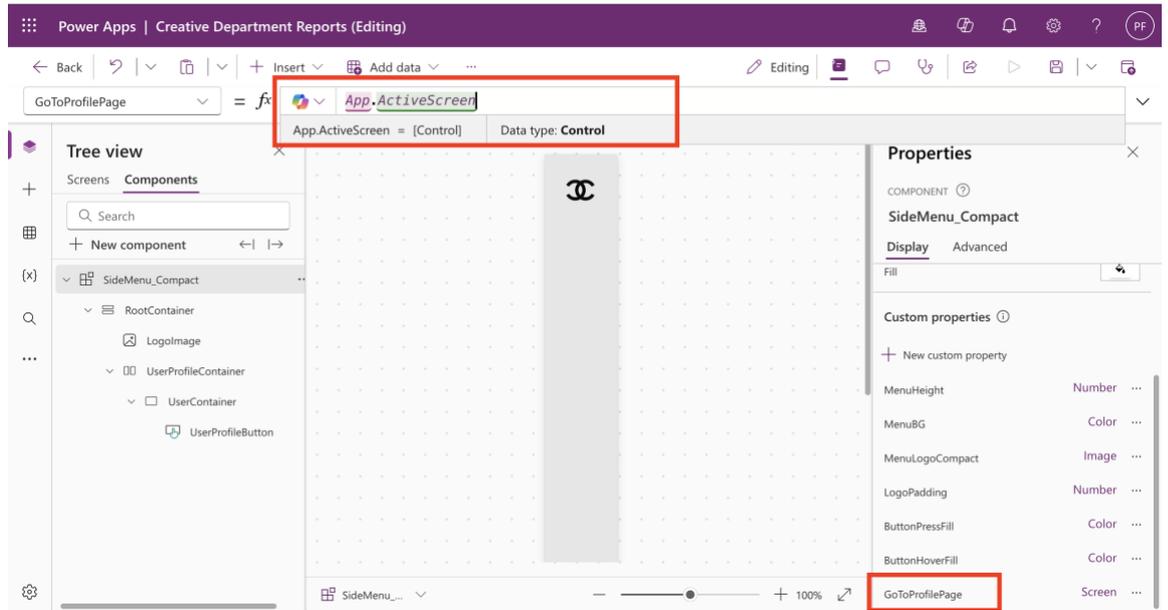
Also, set a “Tool tip” field to say “Profile page” when mouse is hovered above the button:



- Create a custom property called “GoToProfilePage” of data type “Screen”



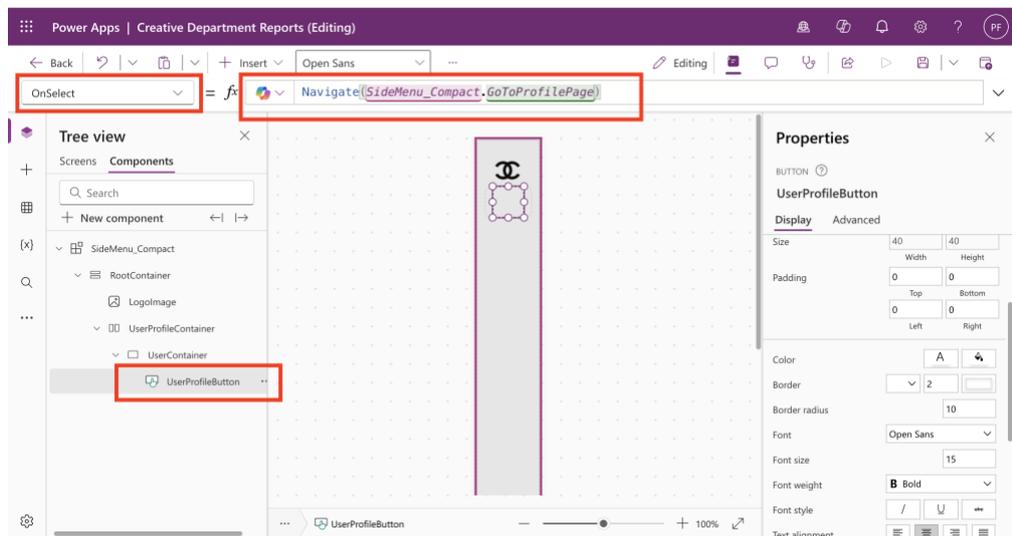
Set the default value to: *App.ActiveScreen*



Apply the property to the button:

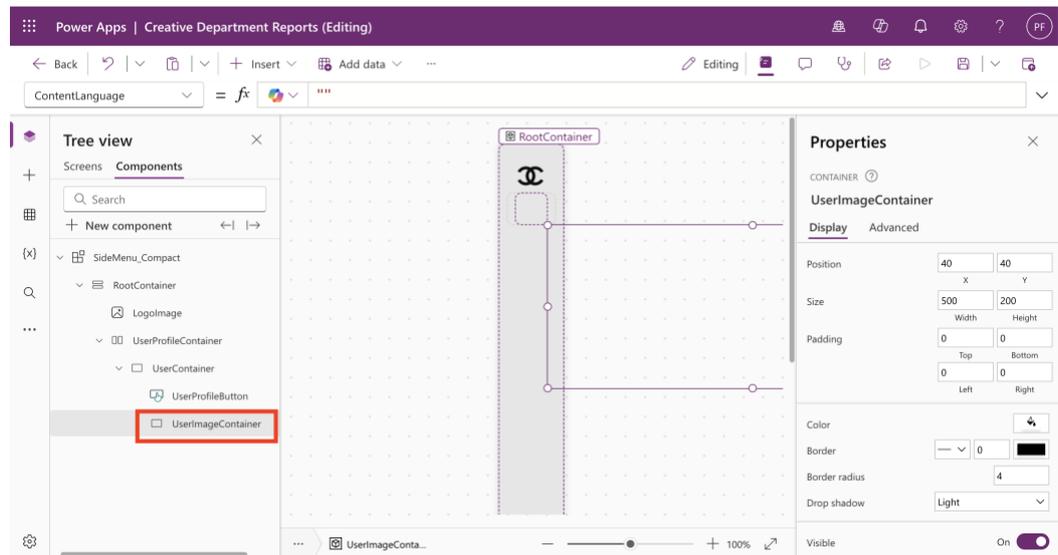
- Select the "UserProfileButton"
- Find the "OnSelect" property in the drop-down list at the top
- Enter the following formula to the formula bar:

Navigate(SideMenu_Compact.GoToProfilePage)

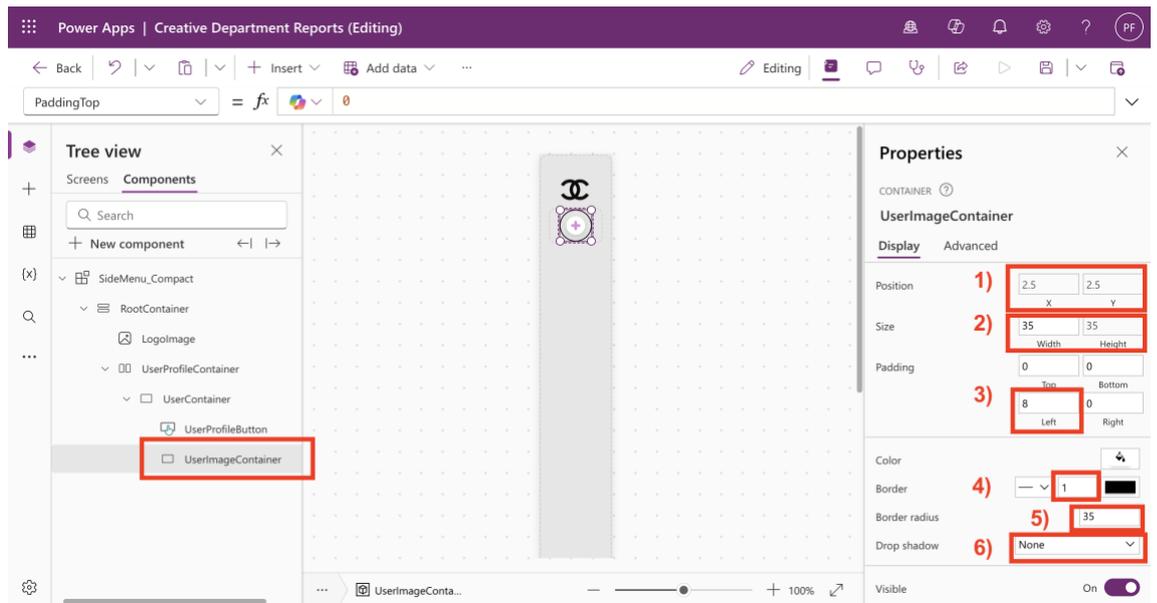


Adding User Image under the button

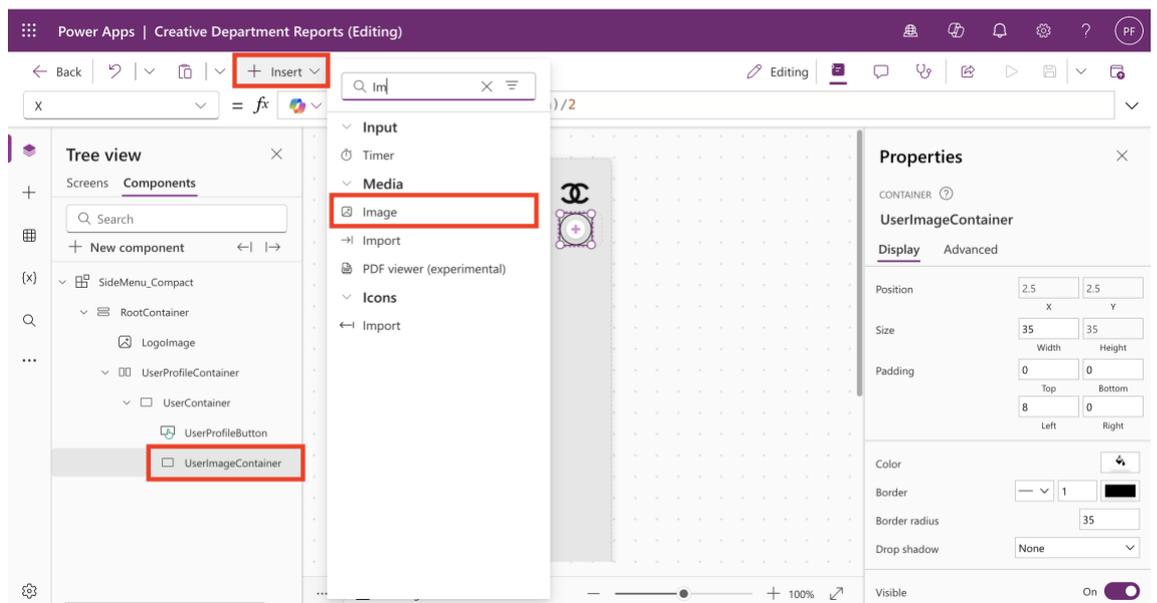
- Under the “UserProfileContainer” insert a new blank container and name it “UserImageContainer”. Make sure that this container is located under the button as shown here (right-click on the container and click “Reorder” in necessary)



- Set its properties as follows:
 - 1) $X = (\text{Parent.Width} - \text{Self.Width})/2$
 - 2) $Y = (\text{Parent.Height} - \text{Self.Height})/2$
 - 2) Width = 35
Height = Self.Width
 - 3) Left padding = 8
 - 4) Border width = 1
 - 5) Border radius = 35 (to make it a circle)
 - 6) Drop shadow = None



- Insert Image control under the blank “UserImageContainer”:



Rename the Image control to “UserImage” and edit its properties:

- X = 0
- Y = 0
- Width = Parent.Width
- Height = Parent.Height
- Border width = 0

Power Apps | Creative Department Reports (Editing)

Height = fx Parent.Height

Tree view

- Screens
- Components
 - Search
 - New component
 - SideMenu_Compact
 - RootContainer
 - Logolmage
 - UserProfileContainer
 - UserContainer
 - UserProfileButton
 - UserImageContainer
 - UserImage

Properties

IMAGE

UserImage

Display Advanced

Visible On

Position

Size

Padding

Color

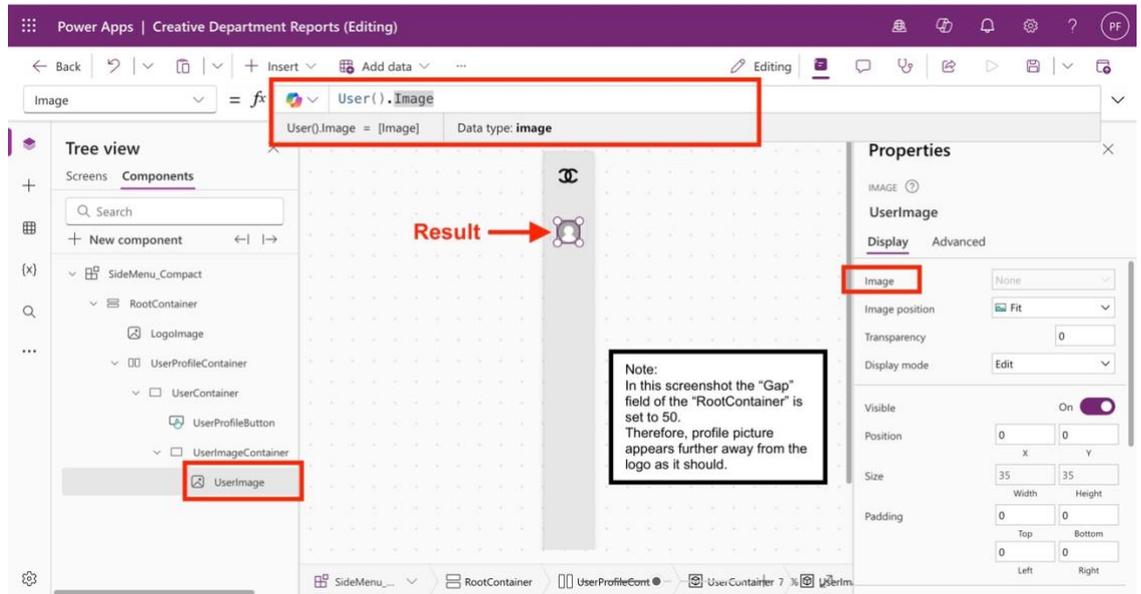
Border

Border radius

Auto disable on sele... On

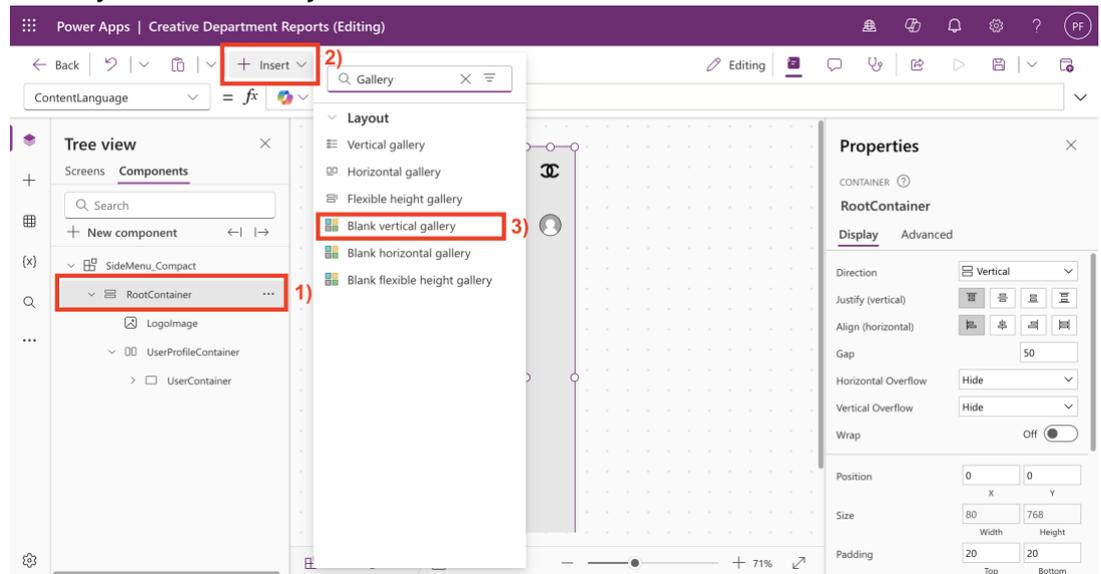
- For the “UserImage” control select the “Image” field in the properties panel and paste the following formula to the formula bar: `User().Image`

This formula pulls the image from Microsoft 365 (Office 365) profile. If the user has no profile picture, it may return an empty value or a default placeholder.

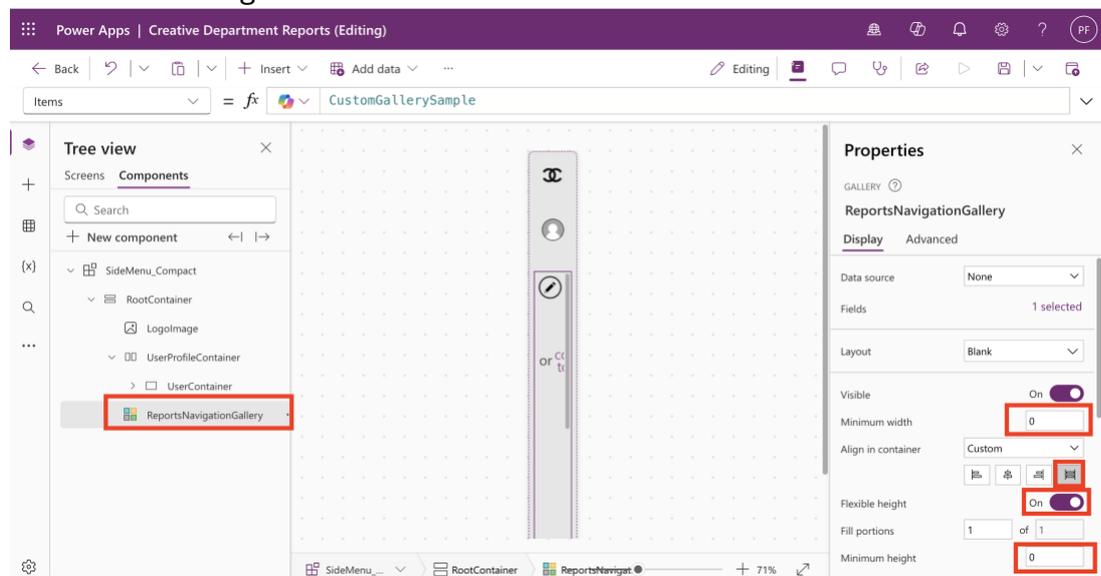


Buttons for Navigation between “Individual Reports” and “Team Reports” pages

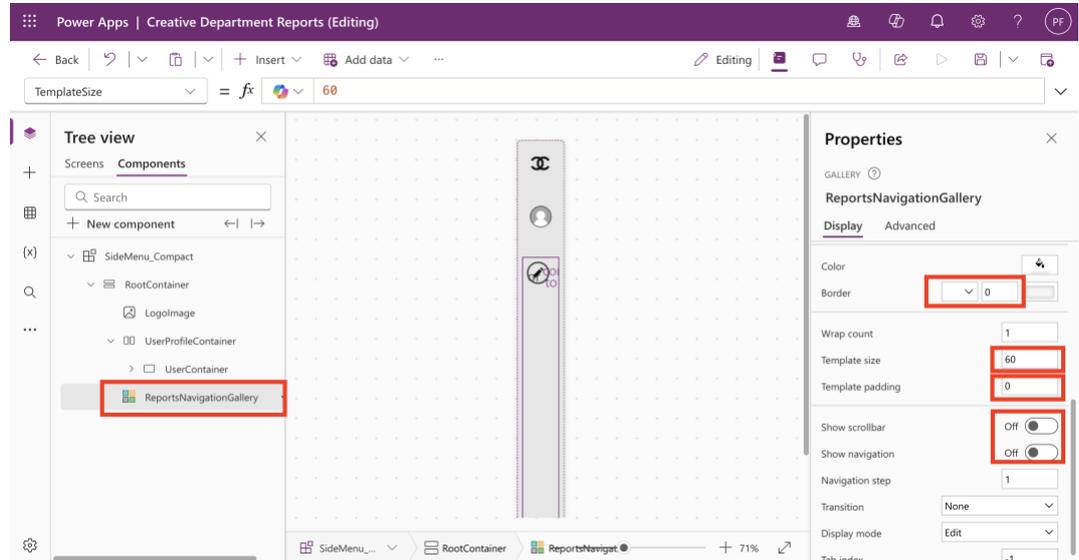
- To store the buttons for navigation between “Individual Reports” and “Team Reports” pages, we will use a **“Blank vertical gallery”**. The “Blank vertical gallery” allows to hold multiple buttons (or other elements) in a structured vertical list. It is easy to add or remove buttons from it so it provides a scalable, and easily customisable way to manage and style the menu layout.



- Rename the gallery (give any meaningful name) to “ReportsNavigationGallery” and start modifying its properties:
 - Minimum width = 0
 - Align in container = stretch
 - Flexible height = turn it On
 - Minimum height = 0

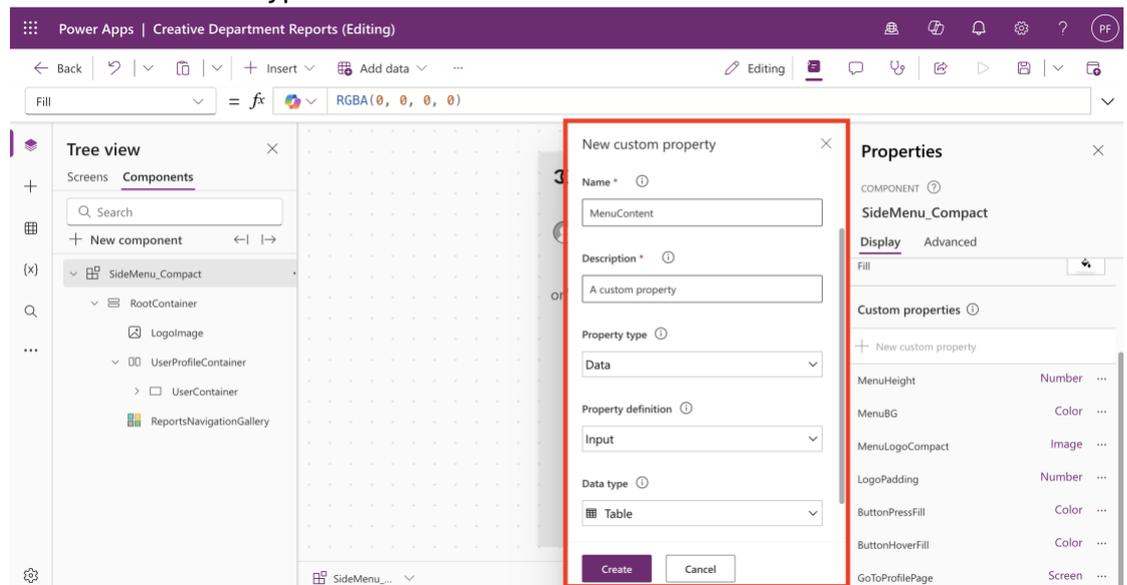


- Remove border
- Template size = 60
- Template padding = 0
- Show scrollbar = turn it Off
- Show navigation = turn it Off



- The gallery needs a **data source** set in its properties to define the list of items it will display. This allows the gallery to dynamically generate content, such as the navigation buttons for "Individual Reports" and "Team Reports" based on the provided data.

Let's create a custom property called "MenuContent" for that. Note it should be of data type "Table":



Before setting the default value for the "MenuContent" property, we need to create couple more things.

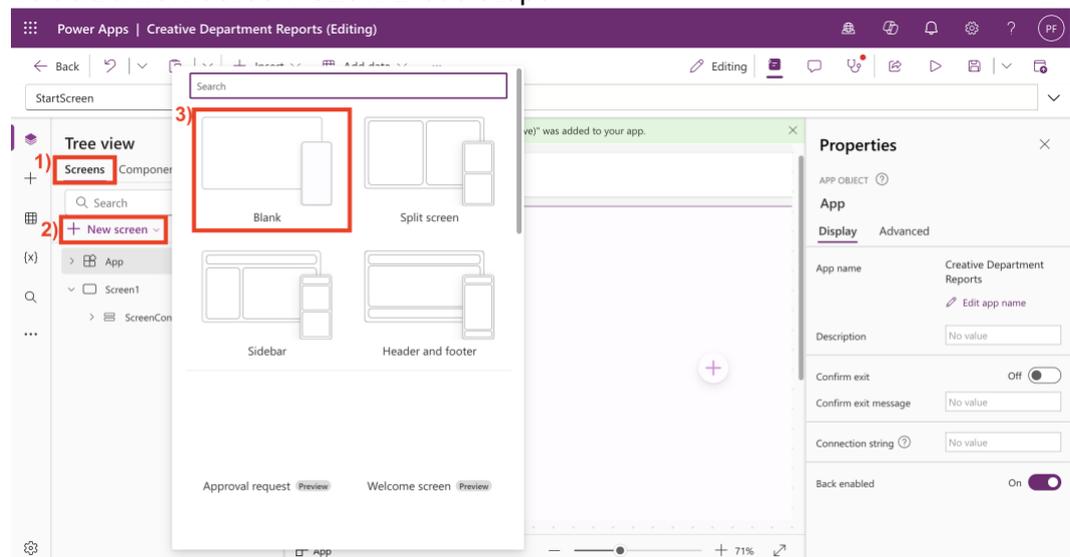
- This version of the app is dedicated to the users from the Digital Creative and Copywriting teams at Chanel.
In the relevant section of the documentation, you can find how users are navigated to the right pages withing the app based on their account and which team they belong to.

Therefore, in Power Apps we need to create:

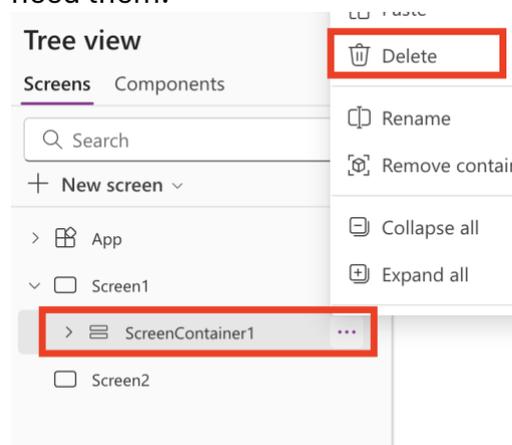
- team reports screen for each of the teams
- individual report screens for each member of the two teams
- profile page screen
- Department (Default/Home Page) – a default screen if the user cannot be matched with any team

In this example, our team used our UCL Microsoft accounts, therefore, assigning Polina and Grace to Copywriting team, and Felicia and Santtosh to Digital Creative team.

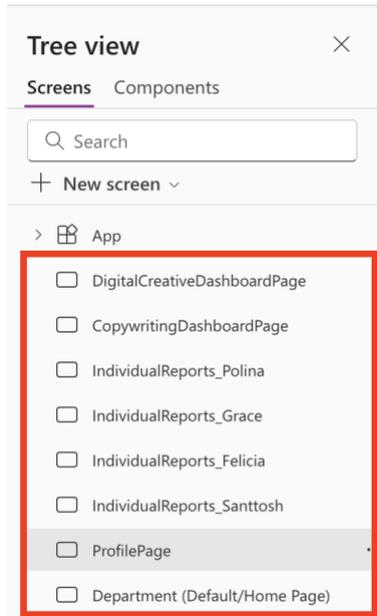
To add a new screen follow these steps:



Let's clear the contents of Screen1 that was there all this time, we don't need them:

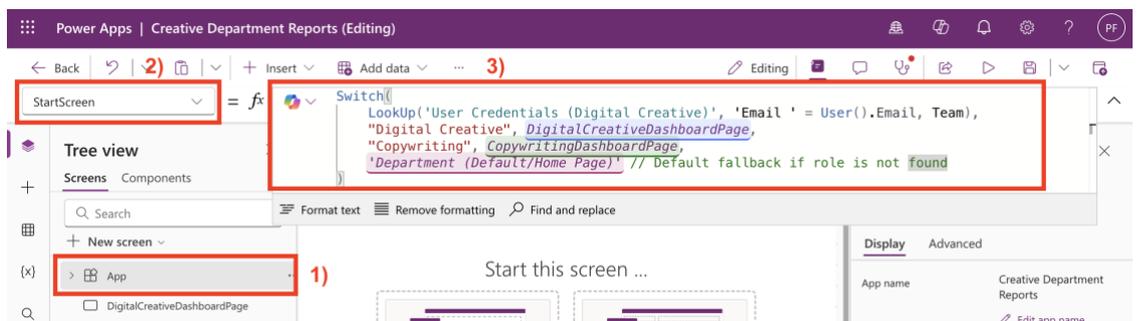


And in total, we need to have 8 screens , name them accordingly:

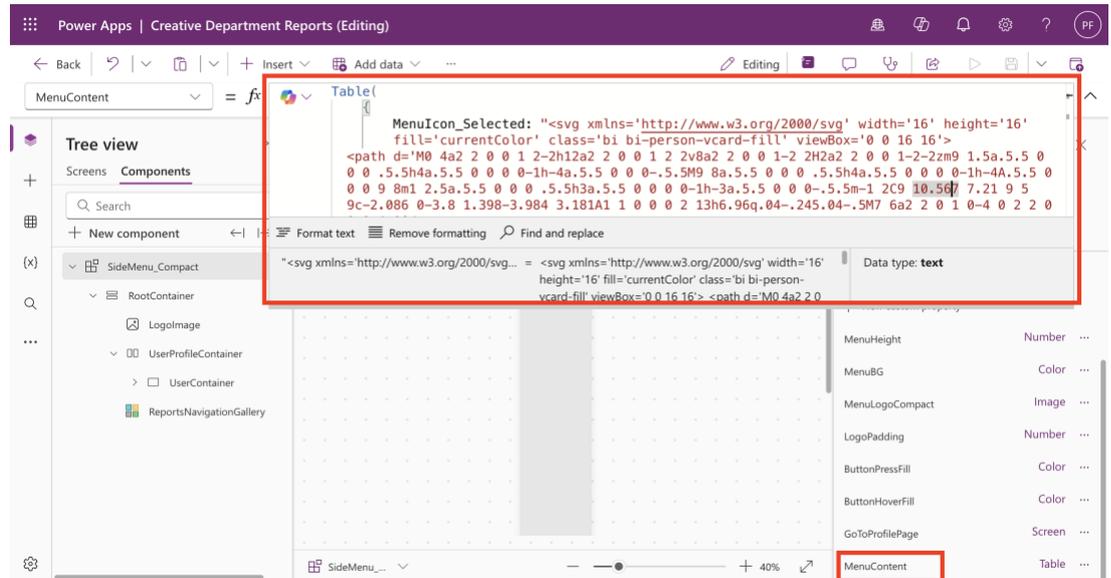


- Ensure that the following formula is present in the “StartScreen” property.
 - 1) Go to “App”
 - 2) Select “StartScreen” property from the drop-down list
 - 3) Enter the following formula

```
Switch(  
    Lookup('User Credentials (Digital Creative)', 'Email ' =  
    User().Email, Team),  
    "Digital Creative", DigitalCreativeDashboardPage,  
    "Copywriting", CopywritingDashboardPage,  
    'Department (Default/Home Page)' // Default fallback if  
    role is not found  
)
```



- Now, going back to “MenuContent” property. Let’s add the following formula as a default formula for this property:



Note that the data here, such as MenuIcon svg image, is just a placeholder, and the actual data will be set once we add this component to the screens:

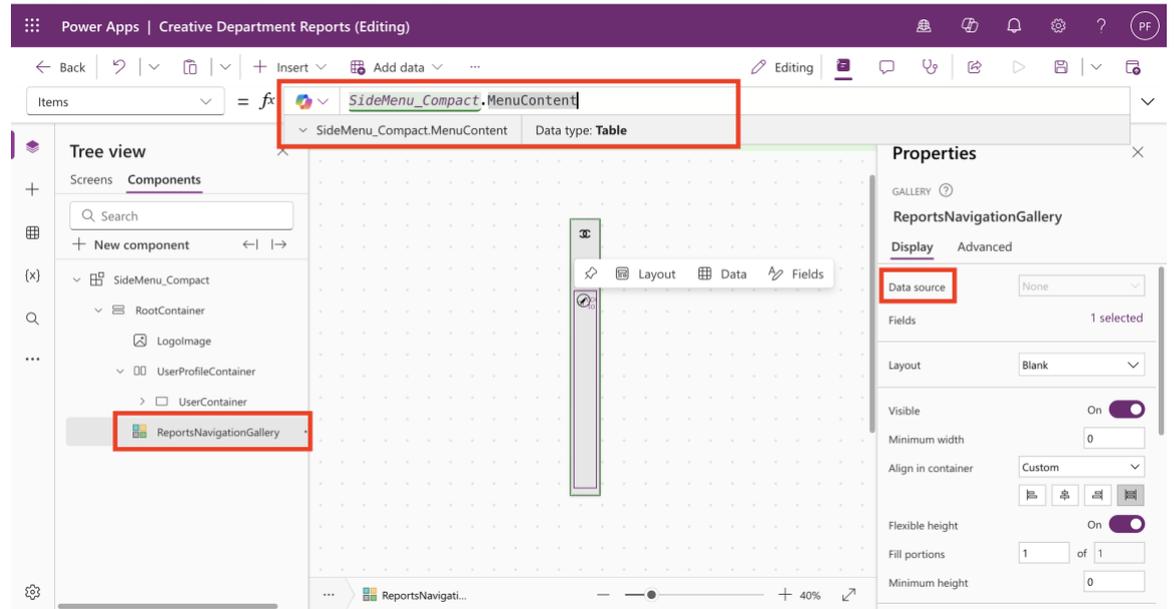
```
Table(
{
  MenuIcon_Selected: "<svg
xmlns='http://www.w3.org/2000/svg' width='16' height='16'
fill='currentColor' class='bi bi-person-vcard-fill' viewBox='0
0 16 16'>
  <path d='M0 4a2 2 0 0 1 2-2h12a2 2 0 0 1 2 2v8a2 2 0 0 1-2
2h2a2 2 0 0 1-2-2zm9 1.5a.5.5 0 0 0 .5.5h4a.5.5 0 0 0 0-1h-
4a.5.5 0 0 0-.5.5m9 8a.5.5 0 0 0 .5.5h4a.5.5 0 0 0 0-1h-4A.5.5
0 0 0 9 8m1 2.5a.5.5 0 0 0 .5.5h3a.5.5 0 0 0 0-1h-3a.5.5 0 0
0-.5.5m-1 2c9 10.567 7.21 9 5 9c-2.086 0-3.8 1.398-3.984
3.181A1 1 0 0 0 2 13h6.96q.04-.245.04-.5m7 6a2 2 0 1 0-4 0 2 2
0 0 0 4 0' />
</svg>",
  MenuIcon: "<svg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' fill='currentColor' class='bi bi-
person-vcard' viewBox='0 0 16 16'>
  <path d='M5 8a2 2 0 1 0 0-4 2 2 0 0 0 4m4-2.5a.5.5 0 0 1
.5-.5h4a.5.5 0 0 1 0 1h-4a.5.5 0 0 1-.5-.5m9 8a.5.5 0 0 1 .5-
.5h4a.5.5 0 0 1 0 1h-4A.5.5 0 0 1 9 8m1 2.5a.5.5 0 0 1 .5-
.5h3a.5.5 0 0 1 0 1h-3a.5.5 0 0 1-.5-.5' />
  <path d='M2 2a2 2 0 0 0-2 2v8a2 2 0 0 0 2 2h12a2 2 0 0 0 2-
2V4a2 2 0 0 0-2-2zM1 4a1 1 0 0 1 1-1h12a1 1 0 0 1 1 1v8a1 1 0
0 1-1 1H8.96q.04-.245.04-.5c9 10.567 7.21 9 5 9c-2.086 0-3.8
1.398-3.984 3.181A1 1 0 0 1 1 12z' />
</svg>",
  MenuID: 1,
}
```

```

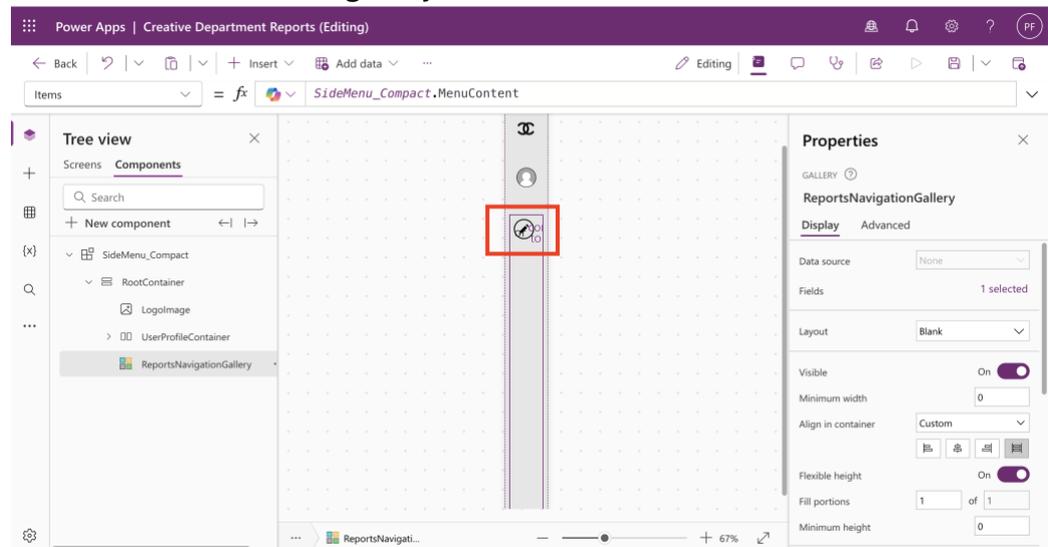
        MenuName: "Individual Reports",
        ScreenToGoTo: Switch(
            LookUp('User Credentials (Digital Creative)',
Email = User().Email).IndividualReportScreenName,
            "IndividualReport_Polina",
IndividualReports_Polina,
            "IndividualReport_Grace", IndividualReports_Grace,
            "IndividualReport_Santtosh",
IndividualReports_Santtosh,
            "IndividualReport_Felicia",
IndividualReports_Felicia,
            'Department (Default/Home Page)' ), // Default
screen if no match
            ToolTips: "Individual Reports"
        },
        {
            MenuIcon_Selected: "<svg
xmlns='http://www.w3.org/2000/svg' width='16' height='16'
fill='currentColor' class='bi bi-people-fill' viewBox='0 0 16
16'>
            <path d='M7 14s-1 0-1-1 1-4 5-4 5 3 5 4-1 1-1 1zm4-6a3 3 0 1
0 0-6 3 3 0 0 0 6m-5.784 6A2.24 2.24 0 0 1 5 13c0-1.355.68-
2.75 1.936-3.72A6.3 6.3 0 0 0 5 9c-4 0-5 3-5 4s1 1 1 1zM4.5
8a2.5 2.5 0 1 0 0-5 2.5 2.5 0 0 0 0 5'/>
            </svg>",
            MenuIcon: "<svg xmlns='http://www.w3.org/2000/svg'
width='16' height='16' fill='currentColor' class='bi bi-
people' viewBox='0 0 16 16'>
            <path d='M15 14s1 0 1-1-1-4-5-4-5 3-5 4 1 1 1 1zm-7.978-1L7
12.996c.001-.264.167-1.03.76-1.72C8.312 10.629 9.282 10 11
10c1.717 0 2.687.63 3.24 1.276.593.69.758 1.457.76 1.721-
.008.002-.014.002zM11 7a2 2 0 1 0 0-4 2 2 0 0 0 4m3-2a3 3 0
1 1-6 0 3 3 0 0 1 6 0M6.936 9.28a6 6 0 0 0-1.23-.247A7 7 0 0 0
5 9c-4 0-5 3-5 4q0 1 1 1h4.216A2.24 2.24 0 0 1 5 13c0-
1.01.377-2.042 1.09-2.904.243-.294.526-.569.846-.816M4.92
10A5.5 5.5 0 0 0 4 13H1c0-.26.164-1.03.76-1.724.545-.636
1.492-1.256 3.16-1.275ZM1.5 5.5a3 3 0 1 1 6 0 3 3 0 0 1-6 0m3-
2a2 2 0 1 0 0 4 2 2 0 0 0 0-4'/>
            </svg>",
            MenuID: 2,
            MenuName: "Team Reports",
            ScreenToGoTo: Switch(
                LookUp('User Credentials (Digital Creative)',
Email = User().Email, Team),
                "Digital Creative", DigitalCreativeDashboardPage,
                "Copywriting", CopywritingDashboardPage,
                'Department (Default/Home Page)'),
            ToolTips: "Team Reports"
        }
    )

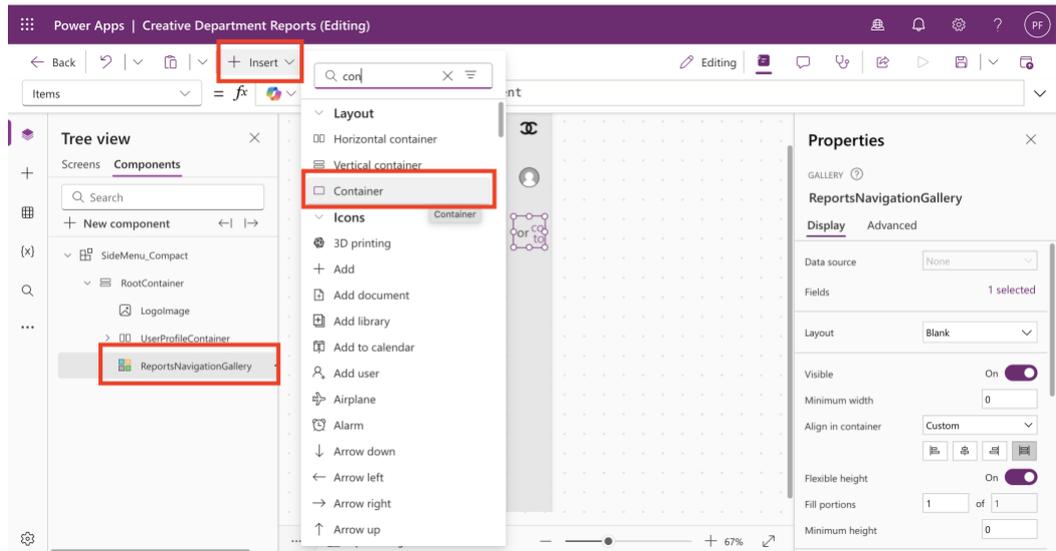
```

Apply the property to the gallery's "Data source" field:



- Now, working with the contents of the gallery.
Add a blank container:
Ensure to click on this pencil icon on the gallery first, to be able to insert the container inside the gallery:





Set its properties:

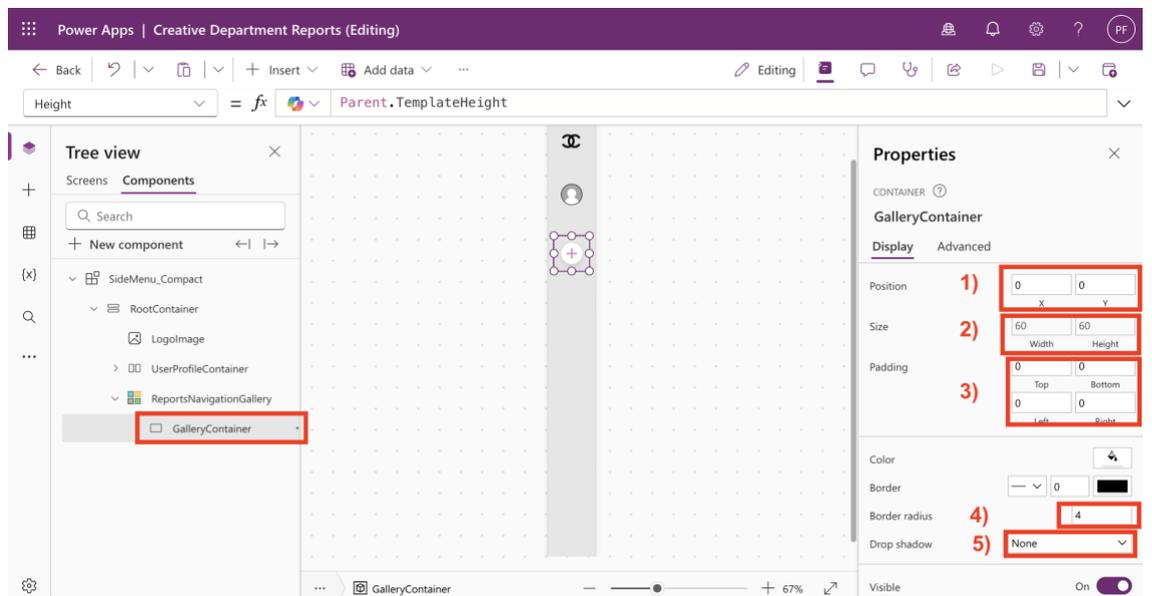
- 1) X = 0, Y = 0
- 2) Width = Parent.TemplateWidth
Height = Parent.Templateheight

3) All paddings = 0

* Border can be set to None or border width to 0 as shown here.

4) Border radius = 4

5) Drop shadow = None



- Add another blank container “MenuButtonContainer”.

Set its properties:

1) X = (Parent.Width - Self.Width) / 2

Y = (Parent.Height - Self.Height) / 2

2) Width = **Self.Height**

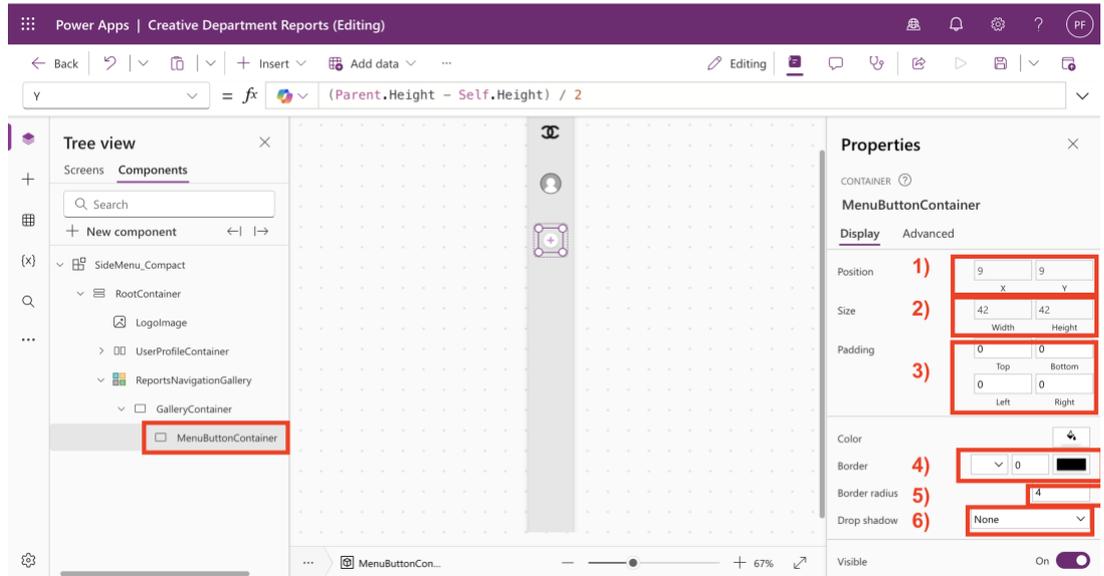
Height = **70% * Parent.Height**

3) All paddings = 0

4) Border None and width = 0

5) Border radius = 4 * **Correction** – the border radius must be set to 10 so that the shape of the rectangle is more rounded. Will be fixed on some future screenshots.

6) Drop shadow = None



- Under the “MenuButtonContainer” insert an Image control to store the menu icon in. Name it “Menulcon”.

Set its properties:

- X = 0, Y = 0
- Width = **Self.Height**, Height = **Parent.Height**
- Top padding = Bottom padding = 5
- Border to None and width to 0
- Border radius = 0

Power Apps | Creative Department Reports (Editing)

Height = Parent.Height

Tree view

- Screens
- Components
 - Search
 - + New component
 - SideMenu_Compact
 - RootContainer
 - LogoImage
 - UserProfileContainer
 - ReportsNavigationGallery
 - GalleryContainer
 - MenuButtonContainer
 - MenuIcon**

Properties

IMAGE

MenuIcon

Display Advanced

Visible On

Position X: 0 Y: 0

Size Width: 42 Height: 42

Padding Top: 5 Bottom: 5 Left: 0 Right: 0

Color

Border

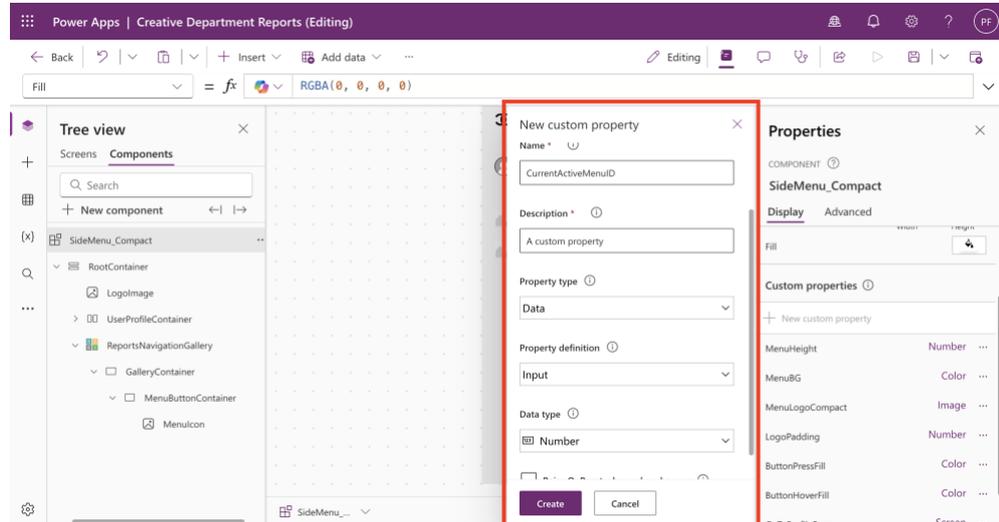
Border radius: 0

- We need a bunch of custom properties for the menu icon image and the button that we will create very soon.

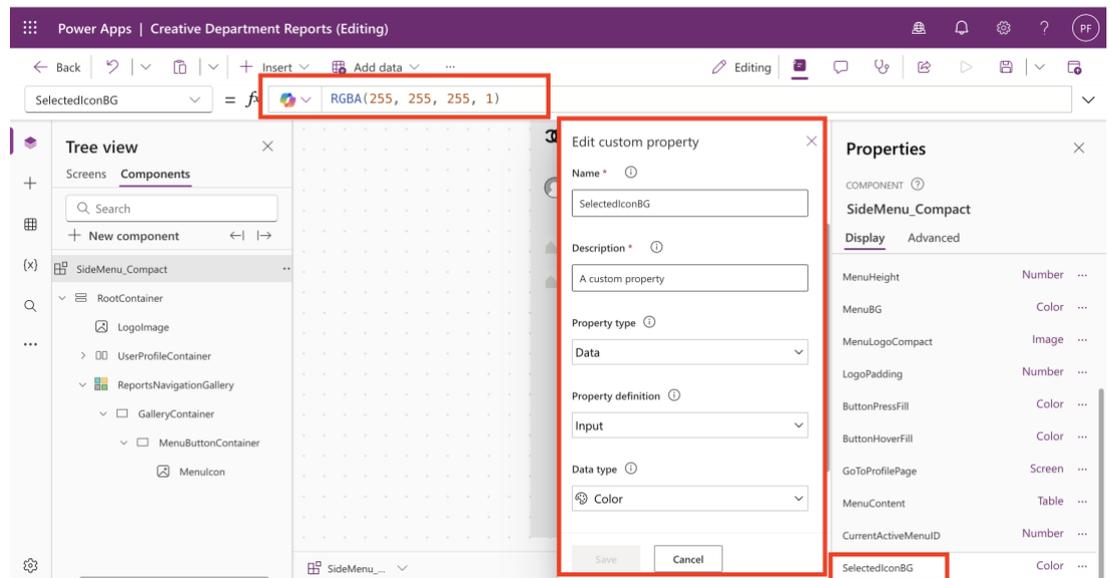
Let's create custom property "CurrentActiveMenuID" to identify the menu icon which has been pressed and therefore currently active while we are on a certain screen.

Date type is Number.

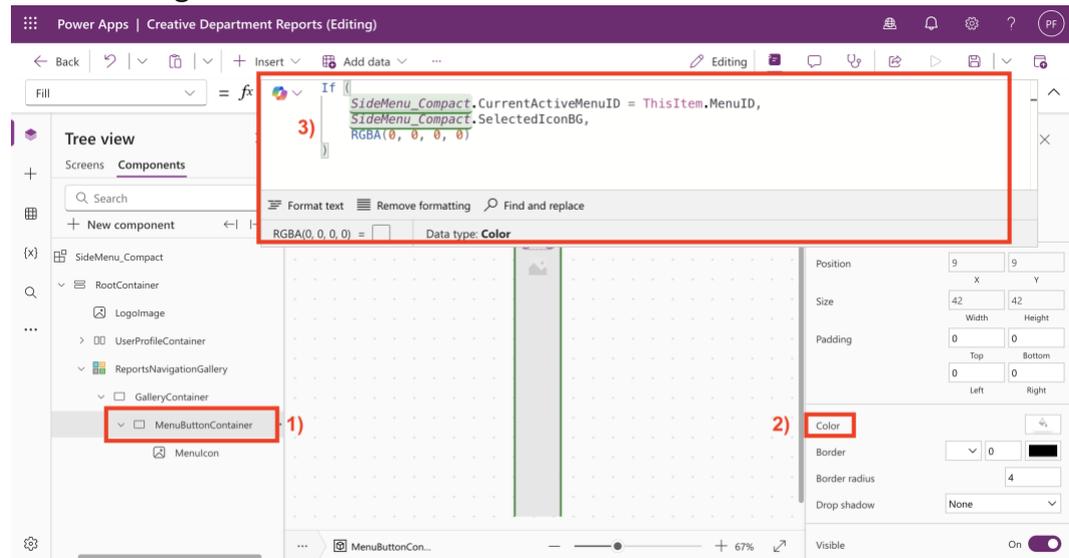
And set its default value to 1 for now.



Now, when menu icon is active, let's turn its background colour fill to white. For that create a new custom property "SelectedIconBG" of data type color and set its default colour to white:



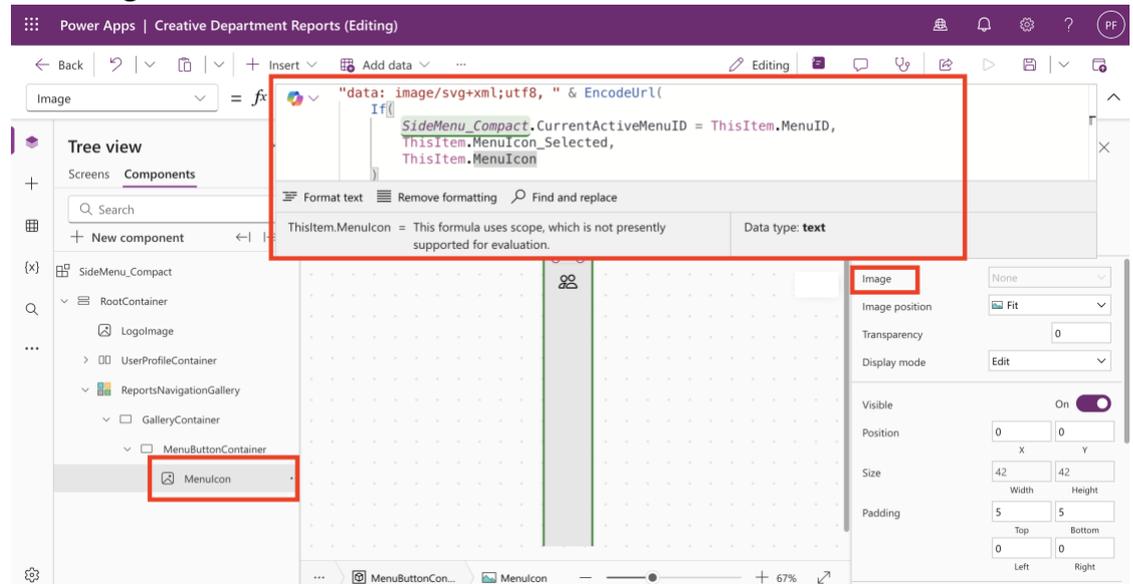
Apply this custom property to the Fill of the MenuButtonContainer, using the following formula:



```

If (
    SideMenu_Compact.CurrentActiveMenuID = ThisItem.MenuID,
    SideMenu_Compact.SelectedIconBG,
    RGBA(0, 0, 0, 0)
)
    
```

- Now, for the Menulcon Image control property called “Image” add the following formula:

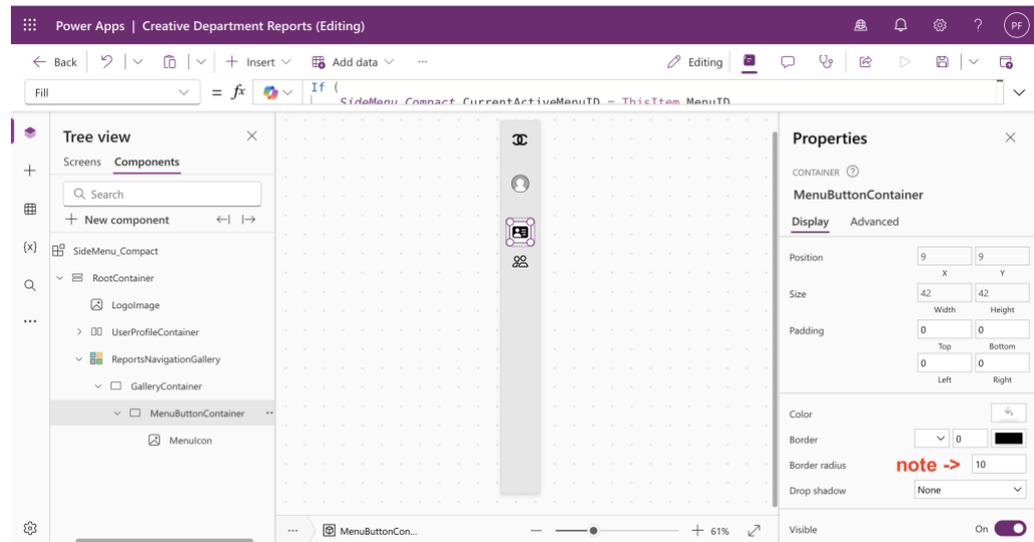


```

"data: image/svg+xml;utf8, " & EncodeUrl(
  If(
    SideMenu_Compact.CurrentActiveMenuID =
ThisItem.MenuID,
    ThisItem.MenuIcon_Selected,
    ThisItem.MenuIcon
  )
)

```

This is what we have so far:



- Let's add buttons to navigate between pages.
Insert a button under MenuButtonContainer and make sure it is the top-most item there.
Set its properties:
 - Text = blank
 - X = 0, Y = 0
 - Width = Parent.Width, Height = Parent.Height
 - All paddings = 5
 - Color = transparent
 - Border none and width 0
 - Border radius = 10
 - PressedFill = *SideMenu_Compact.ButtonPressFill*
 - HoverFill = *SideMenu_Compact.ButtonHoverFill*
 - Tool tip = ThisItem.ToolTips

Power Apps | Creative Department Reports (Editing)

Back | Segoe UI | Editing | ThisItem.ToolTips

Tree view

Screens Components

Search

+ New component

- SideMenu_Compact
 - RootContainer
 - Logolmage
 - UserProfileContainer
 - ReportsNavigationGallery
 - Gallery.Container
 - MenuButton.Container
 - MenuButton
 - MenuIcon

Properties

BUTTON ?

MenuButton

Display Advanced

Text: No value

Display mode: Edit

Visible: On

Position: X: 0, Y: 0

Size: Width: 42, Height: 42

Padding: Top: 5, Bottom: 5, Left: 5, Right: 5

Color: A

Border: 0

Properties

BUTTON ?

MenuButton

Display Advanced

Color: A

Border: 0

Border radius: 10

Font: Segoe UI

Font size: 15

Font weight: B Semibold

Font style: / U abc

Text alignment: [Left] [Center] [Right] [Justify]

Vertical align: Middle

Auto disable on sele...: On

Disabled color: A

Properties

BUTTON ?

MenuButton

Display Advanced

Font: Segoe UI

Font size: 15

Font weight: B Semibold

Font style: / U abc

Text alignment: [Left] [Center] [Right] [Justify]

Vertical align: Middle

Auto disable on sele...: On

Disabled color: A

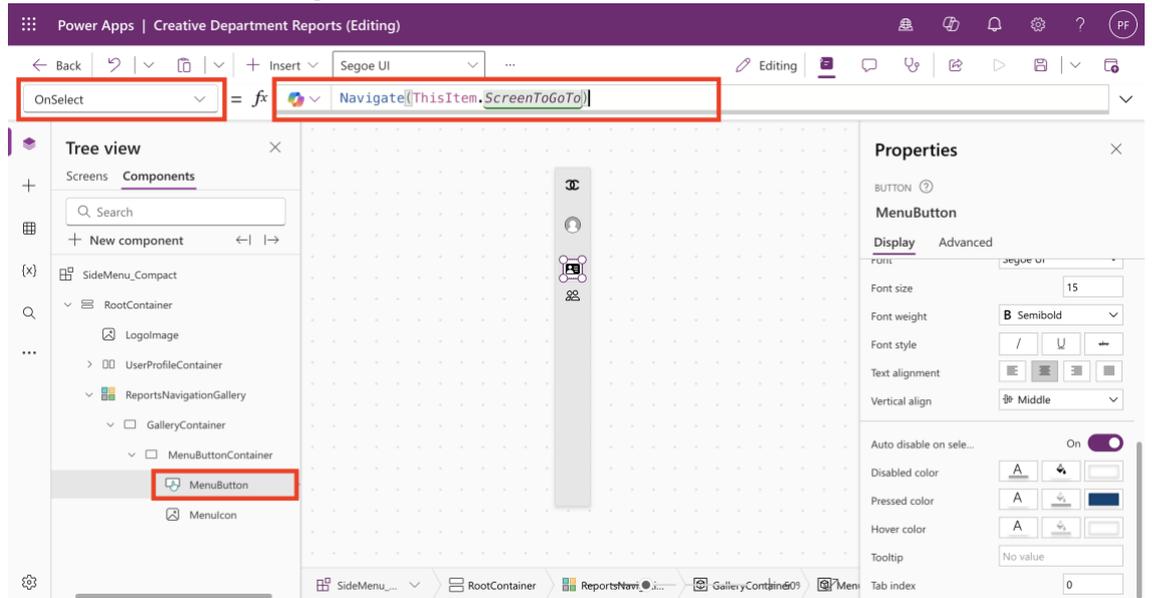
Pressed color: [Color]

Hover color: A

Tooltip: Individual Reports

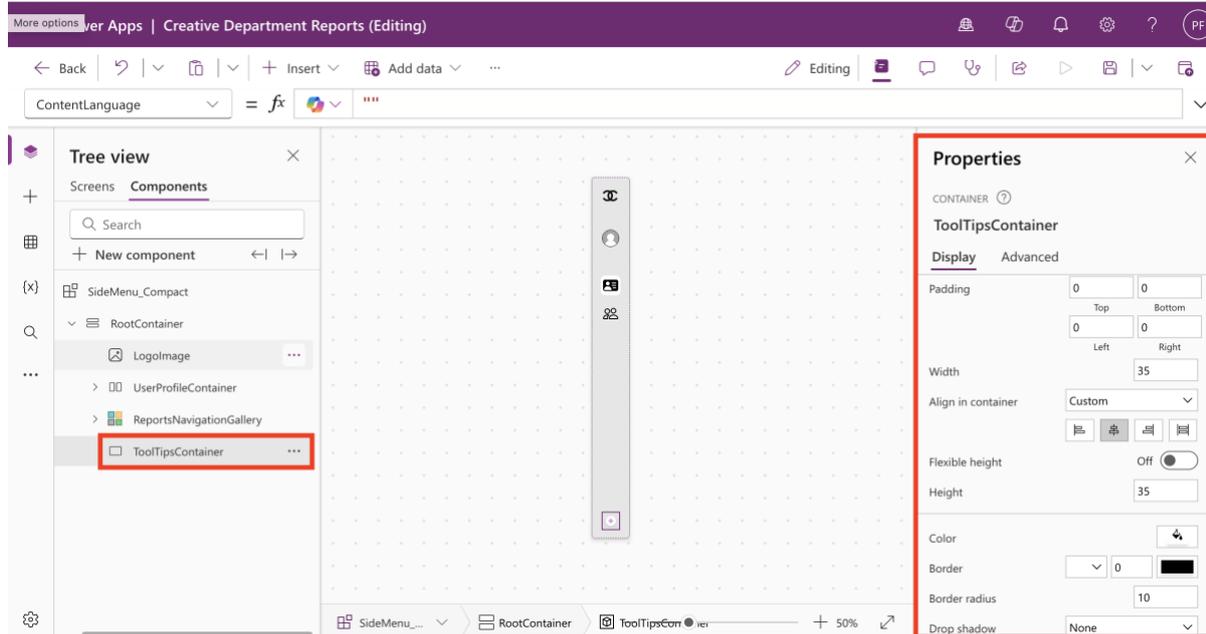
Tab index: 0

- For the same button, find the “OnSelect” property in the drop-down list and enter the following formula in the formula bar:

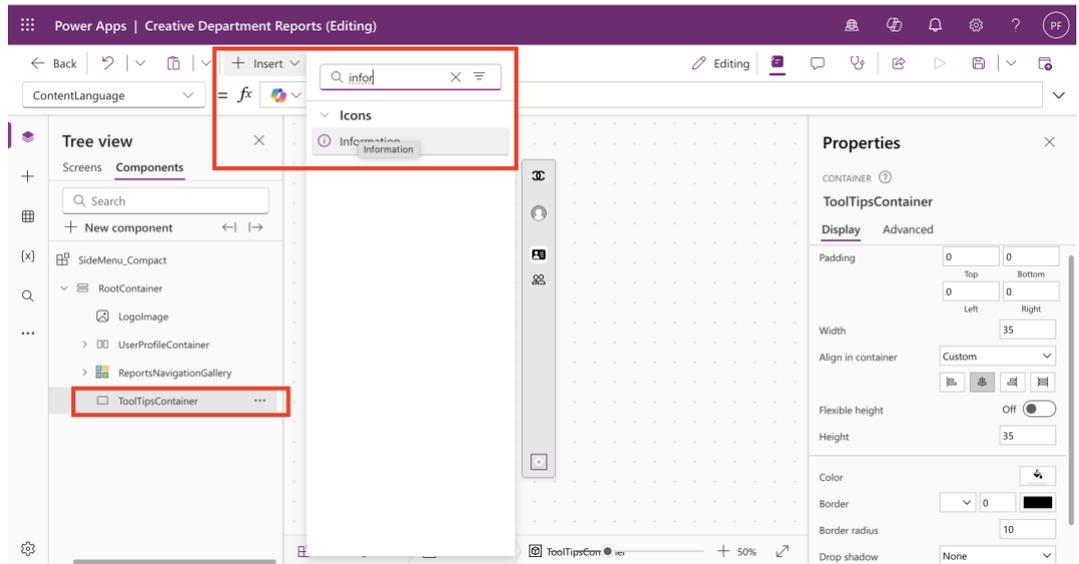


```
Navigate(ThisItem.ScreenToGoTo)
```

- Create a “ToolTipsContainer” and set its properties as shown

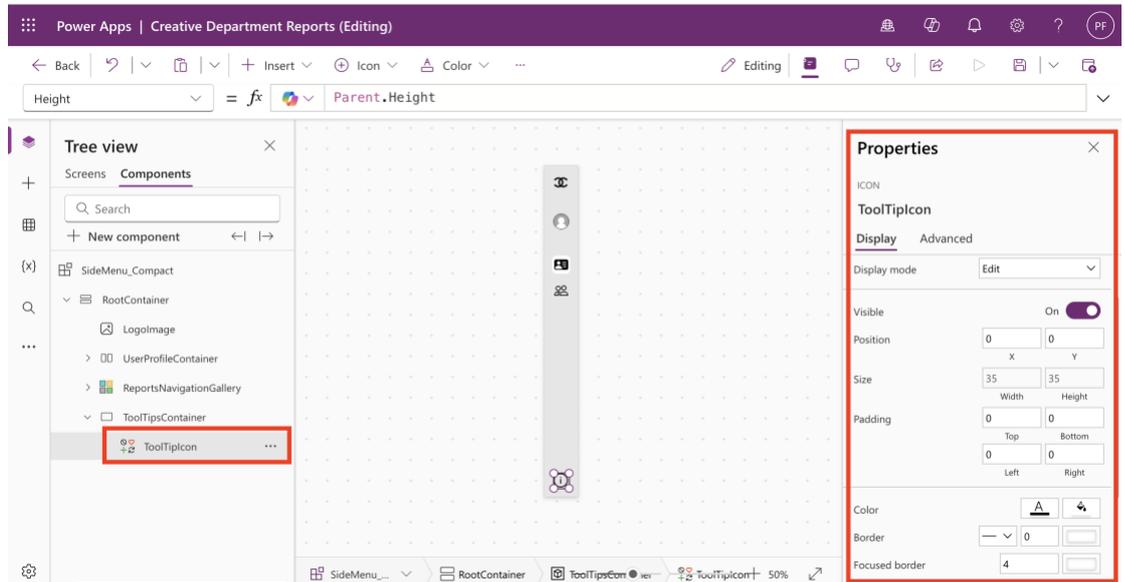


- Insert “Information” icon into it and set its properties:



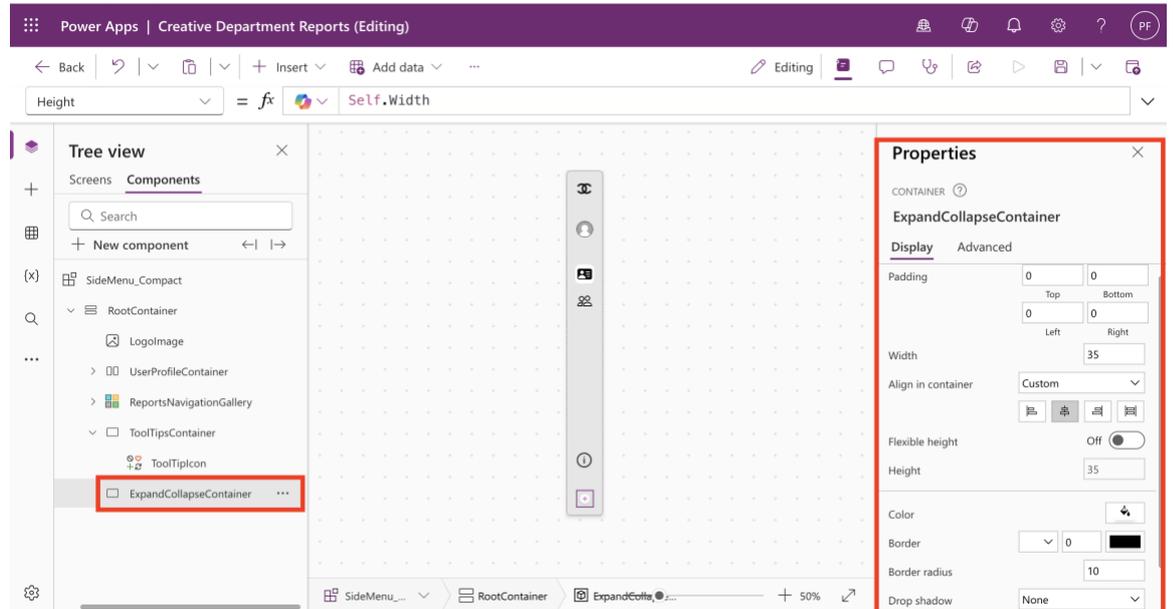
Properties:

- X = 0, Y = 0
- Width = Parent.Width, Height = Parent.Height
- Paddings all set to 0
- Color Black
- Border width 0
- Focus radius = 4

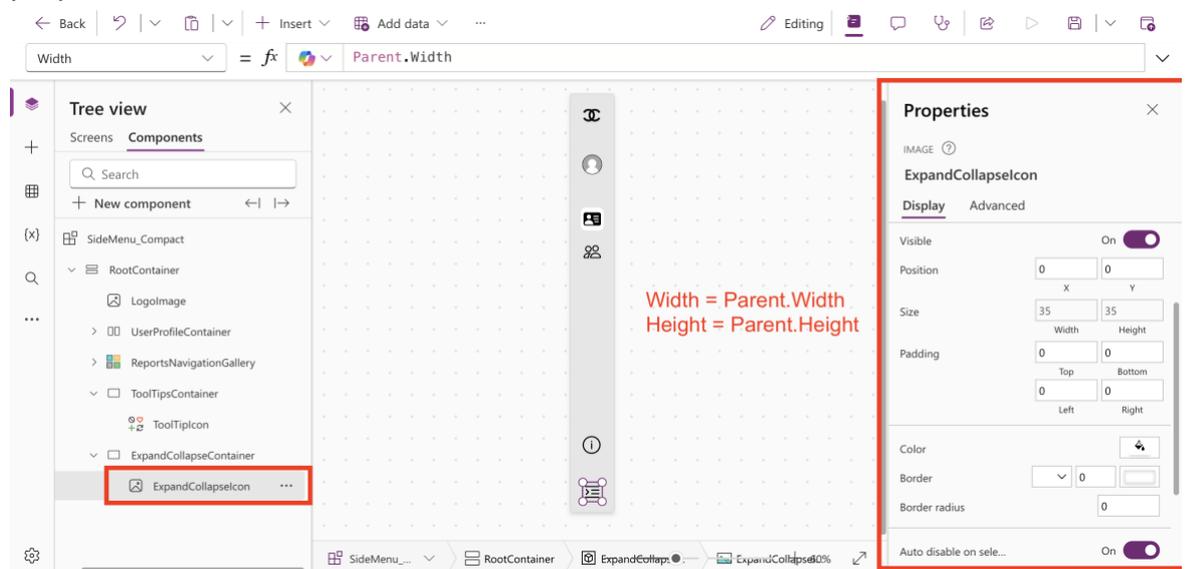


- Add a container for the button to expand and collapse the side navigation menu.

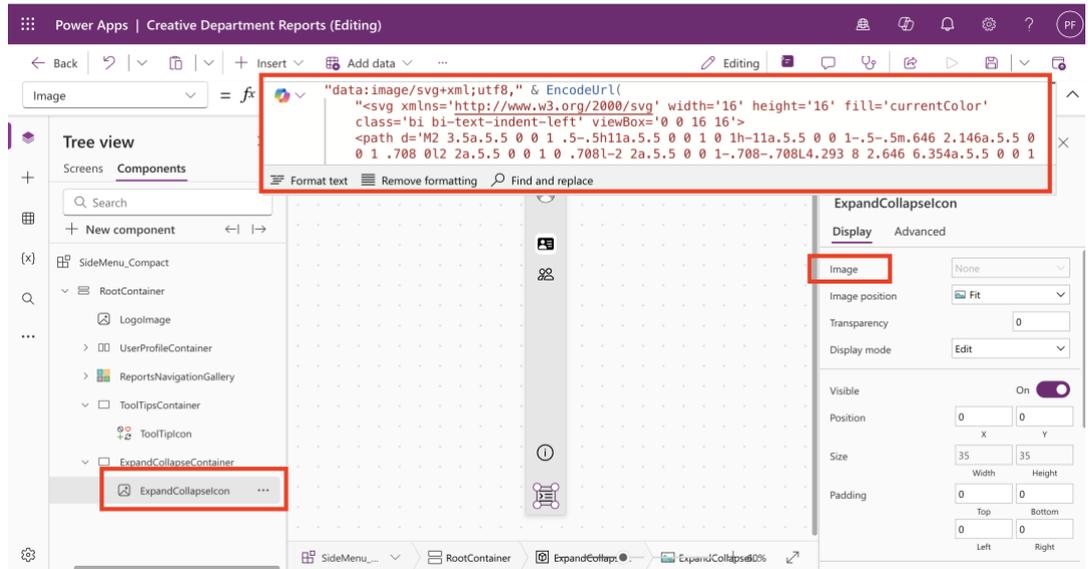
Set its properties as shown:



- Add Image control to it, call it “ExpandCollapseIcon”, and set its properties as shown:



- Set the Image field in the properties panel to the following formula:



```

"data:image/svg+xml;utf8," & EncodeUrl(
  "<svg xmlns='http://www.w3.org/2000/svg' width='16'
height='16' fill='currentColor' class='bi bi-text-indent-left'
viewBox='0 0 16 16'>
  <path d='M2 3.5a.5.5 0 0 1 .5-.5h11a.5.5 0 0 1 0 1h-
11a.5.5 0 0 1-.5-.5m.646 2.146a.5.5 0 0 1 .708 0l2 2a.5.5 0 0
1 0 .708l-2 2a.5.5 0 0 1-.708-.708L4.293 8 2.646 6.354a.5.5 0
0 1 0-.708M7 6.5a.5.5 0 0 1 .5-.5h6a.5.5 0 0 1 0 1h-6a.5.5 0 0
1-.5-.5m0 3a.5.5 0 0 1 .5-.5h6a.5.5 0 0 1 0 1h-6a.5.5 0 0 1-
.5-.5m-5 3a.5.5 0 0 1 .5-.5h11a.5.5 0 0 1 0 1h-11a.5.5 0 0 1-
.5-.5' />
  </svg>"
)

```

The SVG code for this icon was taken from the [free open source Bootstrap Icons library](#) [2].

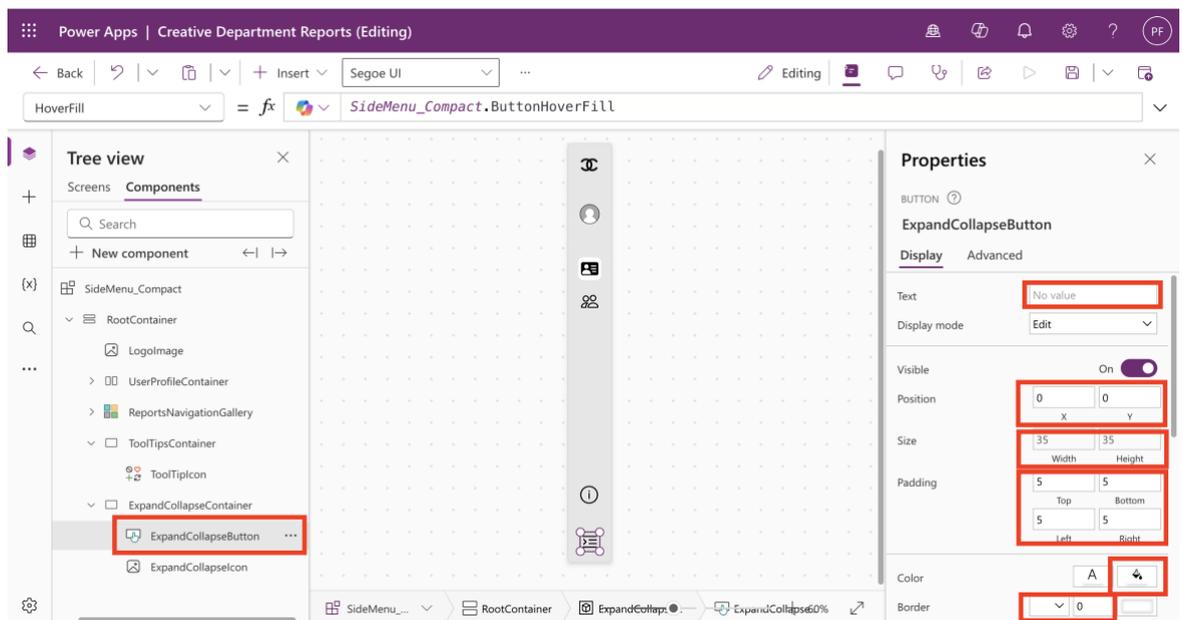
Bootstrap icons used:

Name & Link of the icon in Bootstrap	The image of the icon
Text indent left	

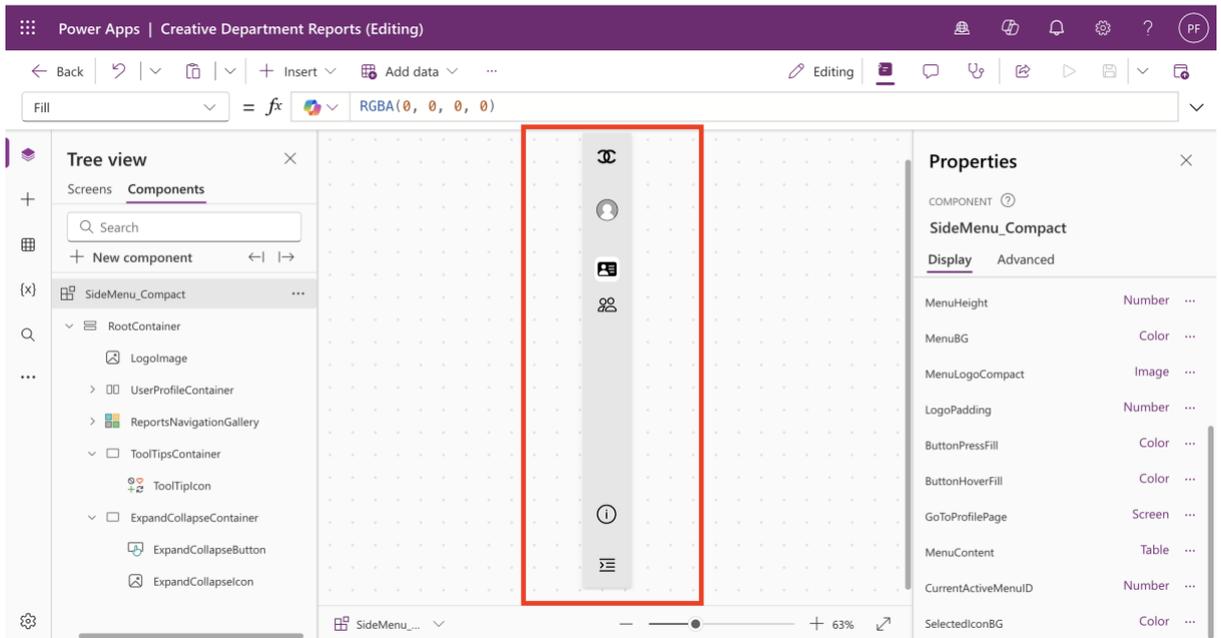
- Add button called “ExpandCollapseButton”, make sure it is the top-most item in the container.

Set its properties as shown below:

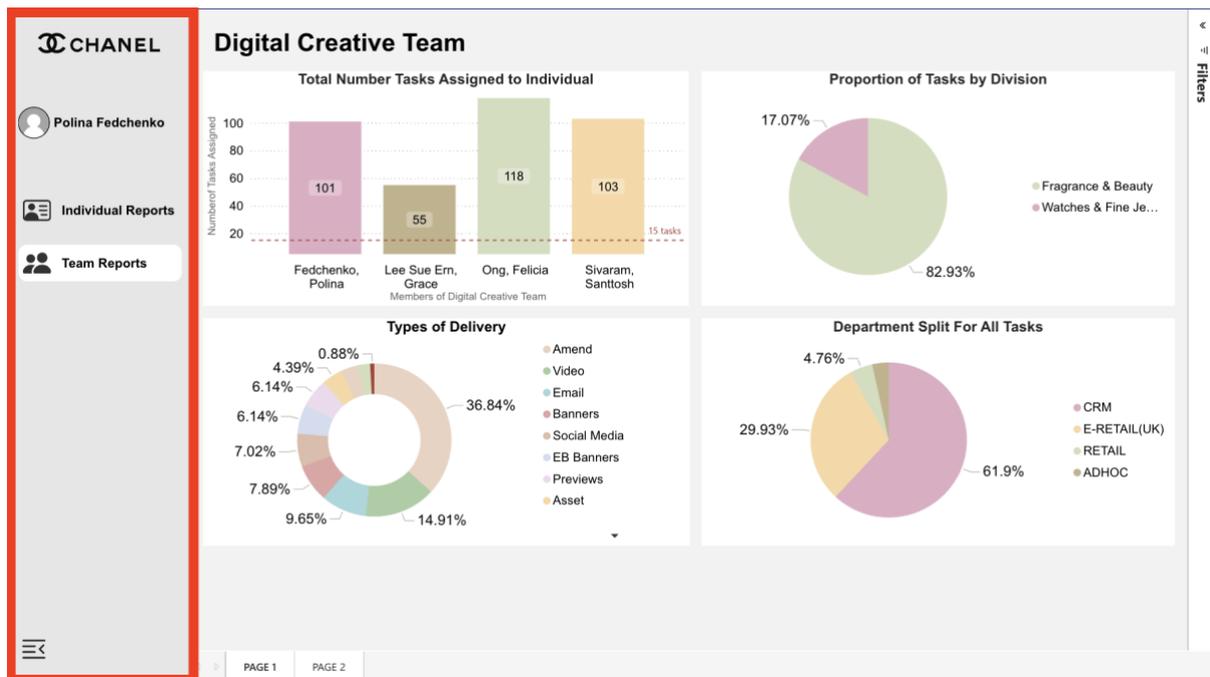
- Width = Parent.Width
- Height = Parent.Height
- Color = transparent
- Border none and width 0
- Border radius = 10
- PressFill = *SideMenu_Compact.ButtonPressFill*
- HoverFill = *SideMenu_Compact.ButtonHoverFill*



- We have finished creating the compact side navigation menu component



Steps – #2 Creating Expandable Navigation Menu component :



References

[1] [https://www.microsoft.com/en-us/power-platform/blog/power-apps/power-
fx-introducing-named-formulas/](https://www.microsoft.com/en-us/power-platform/blog/power-apps/power-fx-introducing-named-formulas/)

[2] <https://icons.getbootstrap.com/>

[3] <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/component-properties>

[4] <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/create-component>

[5] <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/controls/control-container>