Lab 2.1: Code-first development

In this lab, we focus on collaboration between professional and citizen developers.

We will guide you through creating a new Canvas App using code, illustrating how citizen developers can leverage the advantages of code without needing to understand it.

By the end of this lab, you'll gain insights into how professional developers can facilitate a code-first approach that benefits all team members.

Lab 2.1 – Tasks

In this lab, you will complete the following tasks:

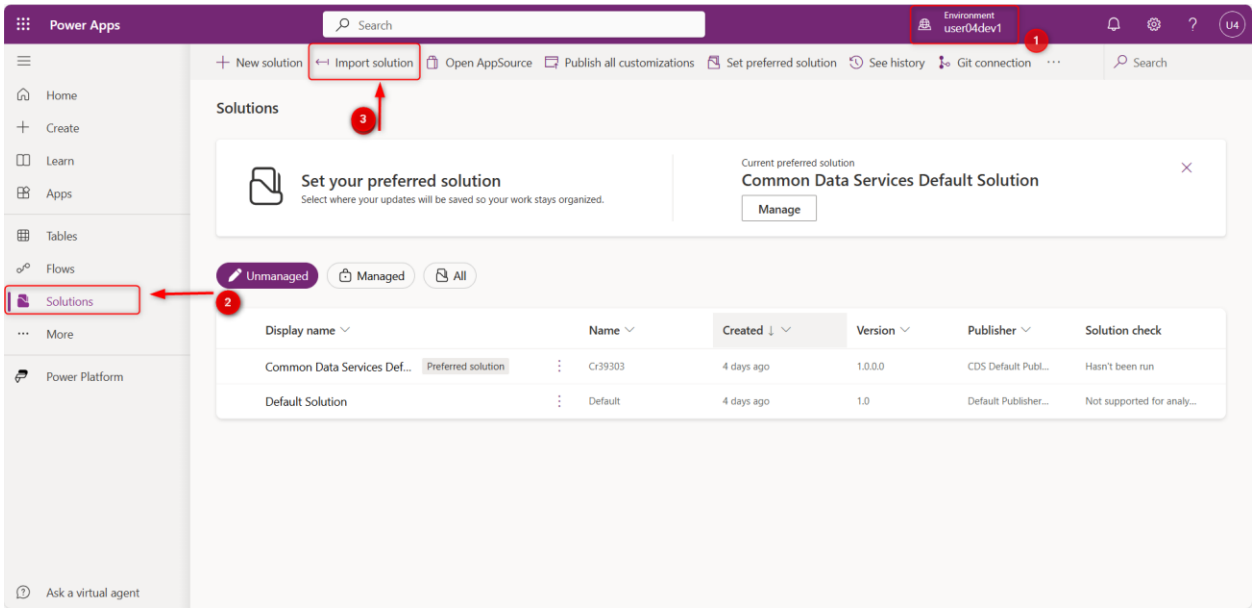
- Import and commit an existing solution.
- Create Data
- Create a new Canvas App.
- Paste YAML code snippets to create controls.
- Add a Code Component (Pro Dev)

Task 1: Import a solution

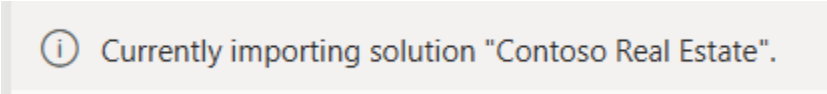
In this task, you will import an existing solution, used to manage real estate inventory. The data and pictures are stored in Dataverse. Once you import, you will commit the code, so we can enhance the solution.

1. Go to https://make.powerapps.com and sign in with your credentials.
    - Be sure to be in your dev environment (created in lab 1).
    - Be sure your environment is connected to your repository.

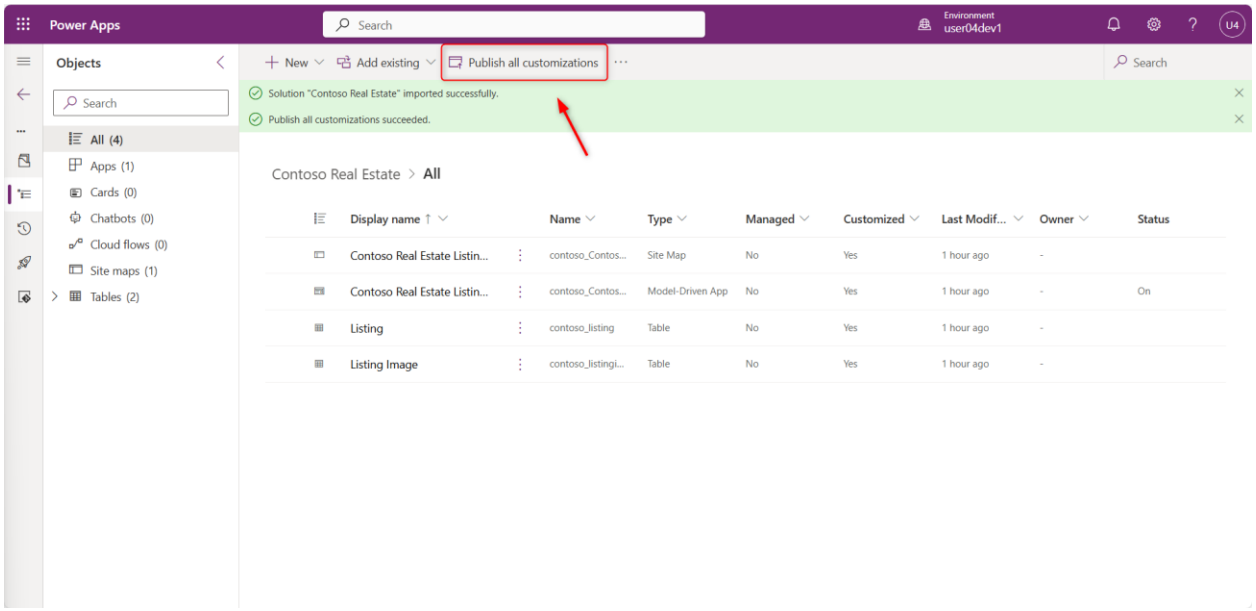
2. Go to solutions. Download and import the solution ContosoRealEstate_1_0_0_1.zip



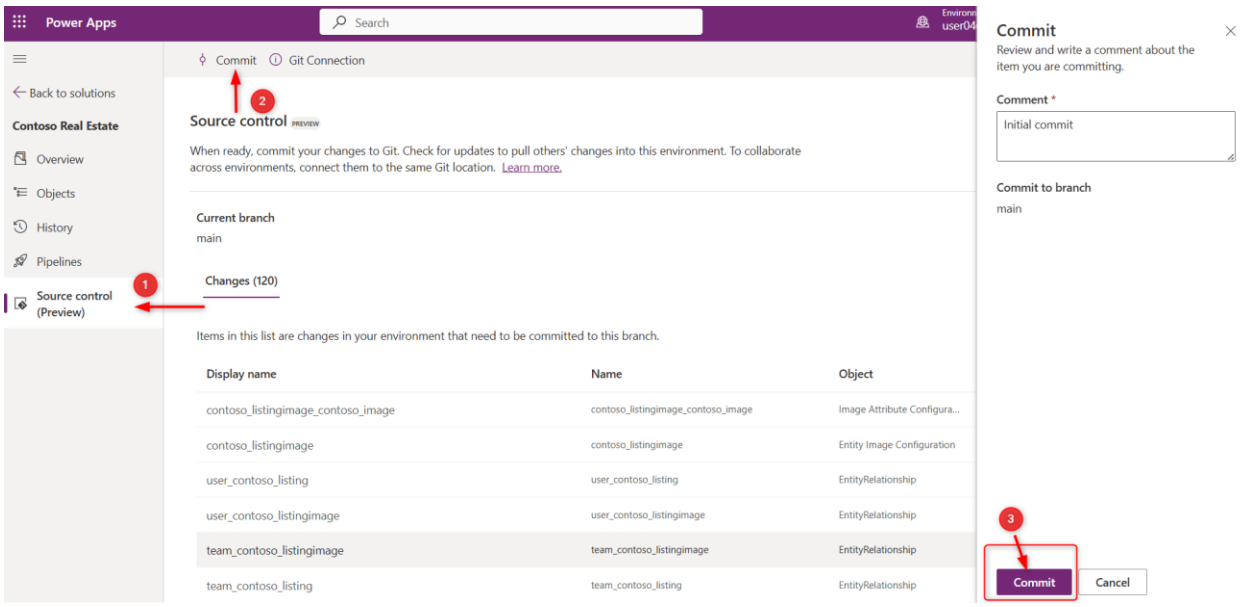
The solution will be imported. Wait until import is complete

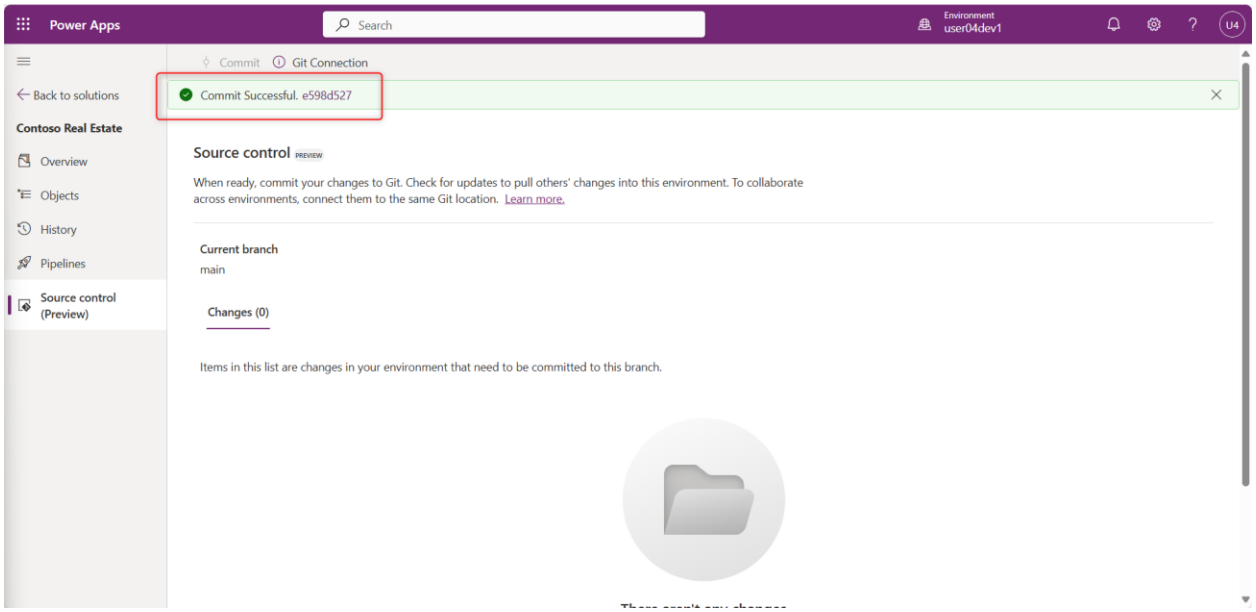


Once import is complete, open the solution “Contoso Real Estate” and select “Publish all customizations”

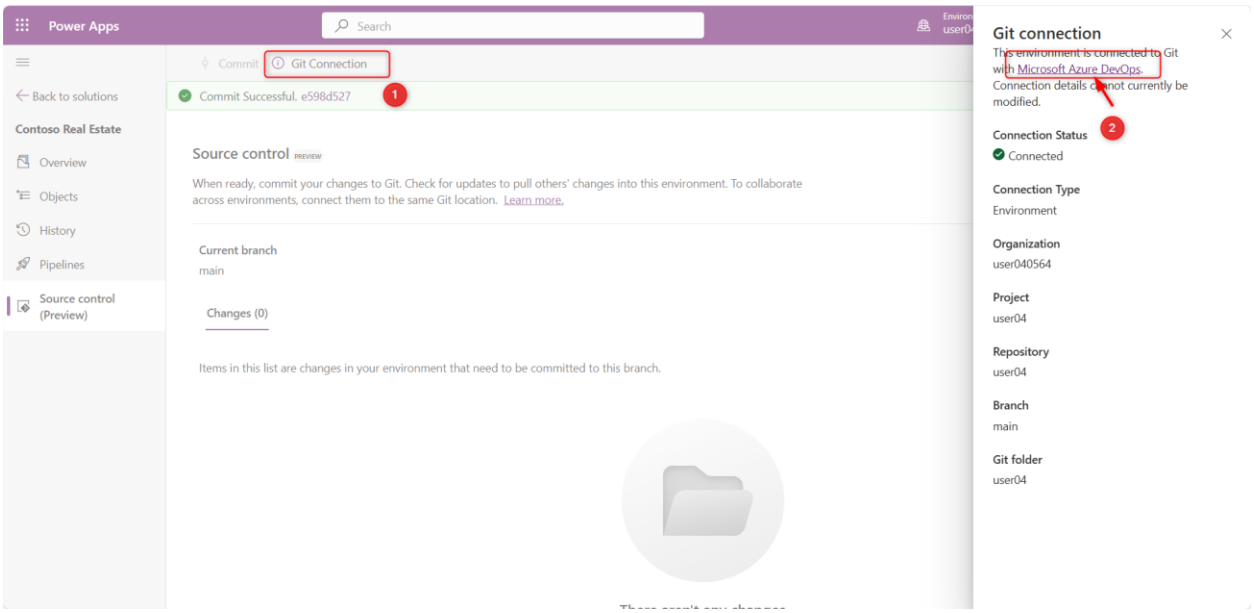


3. Click in Source Control. Notice the list of objects that will be committed. Commit the code to your repository.





4. Open Azure DevOps to see the committed code.



Before we create a new application, let's explore the existing solution and create a few properties.

Task 2 [Option 1]: Import Data using Power Platform CLI.

If you have Power Platform Installed on a Windows machine you can import the data automatically.

1. Copy this data to a local folder.
2. Run "pac data import -d [file]"
3. In the maker portal, Open the Contoso Real Estate Solution. Select the Model-Driven App, and select "Play"
4. Confirm you have properties with Pictures.

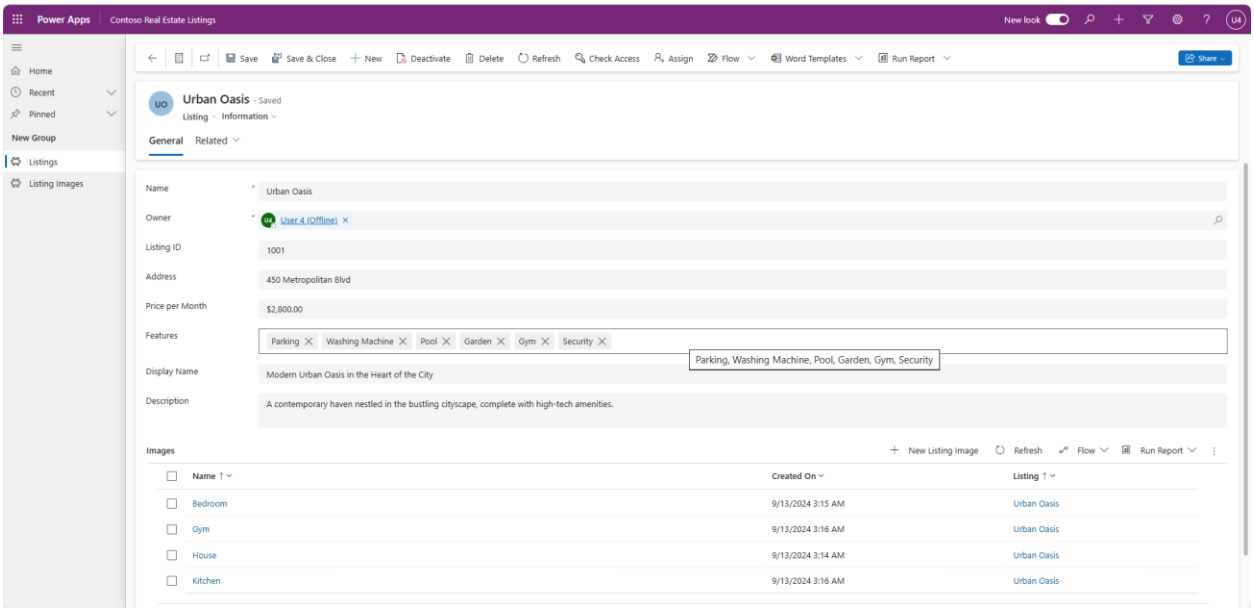
Task 2 [Option 2]: Create Data Manually.

1. In the maker portal, Open the Contoso Real Estate Solution. Select the Model-Driven App, and select "Play"
2. Click in New to add a new Property. Add 3 new properties:

Name	Urban Oasis
Address	450 Metropolitan Blvd
Price per Month	$2,800.00
Features	Parking, Washing Machine, Gym, Security
Display Name	Modern Urban Oasis in the Heart of the City

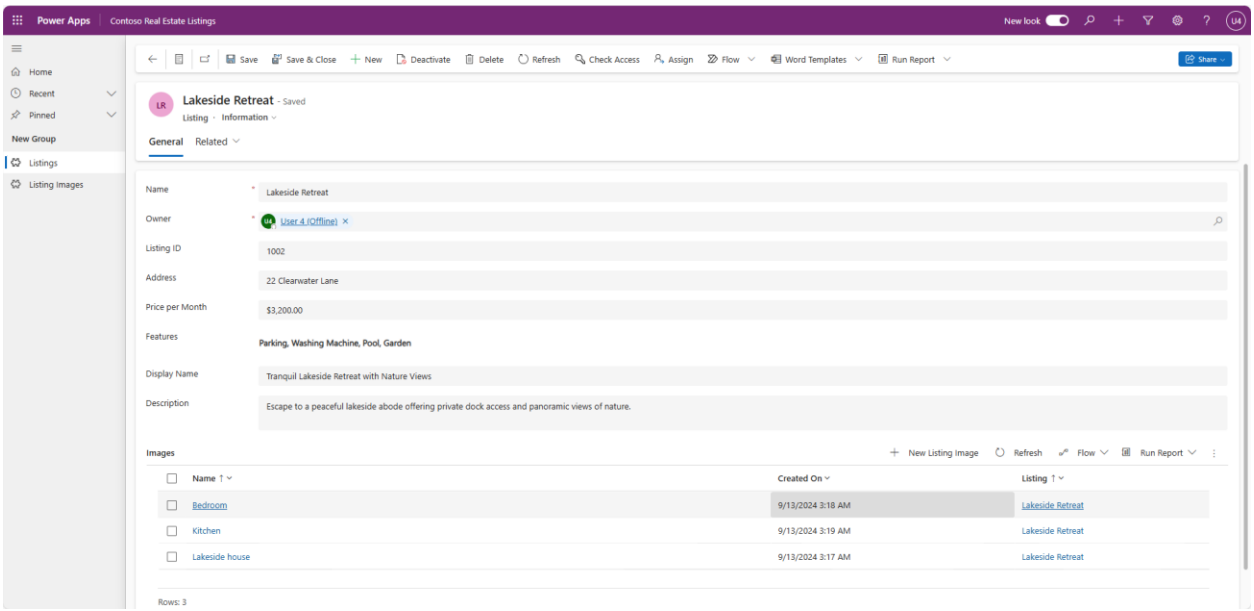
Description	A contemporary haven nestled in the bustling cityscape, complete with high-tech amenities.

click save and add pictures (Download files from repo).



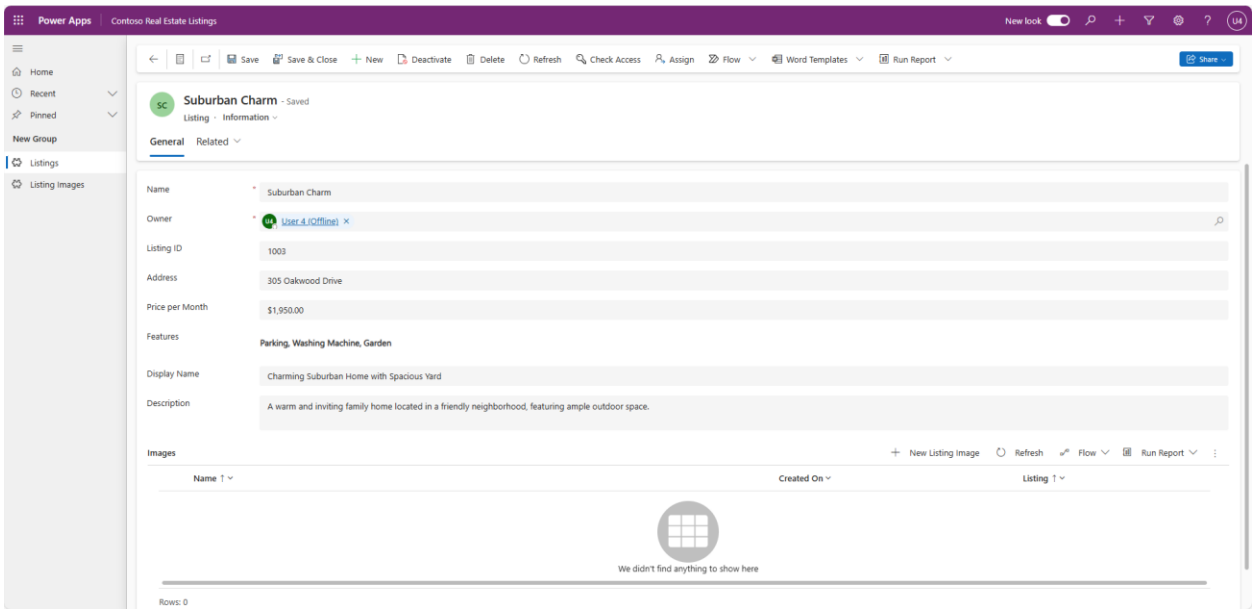
Name	Lakeside Retreat
Address	22 Clearwater Lane
Price per Month	$3,200.00
Features	Parking, Washing Machine, Pool, Garden
Display Name	Tranquil Lakeside Retreat with Nature Views
Description	Escape to a peaceful lakeside abode offering private dock access and panoramic views of nature.

Click save and add pictures (Download files from repo).



Name	Suburban Charm
Address	305 Oakwood Drive
Price per Month	$1,950.00
Features	Parking, Washing Machine, Garden
Display Name	Charming Suburban Home with Spacious Yard
Description	A warm and inviting family home located in a friendly neighborhood, featuring ample outdoor space.

Click save don’t add pictures (we will do from the new app).

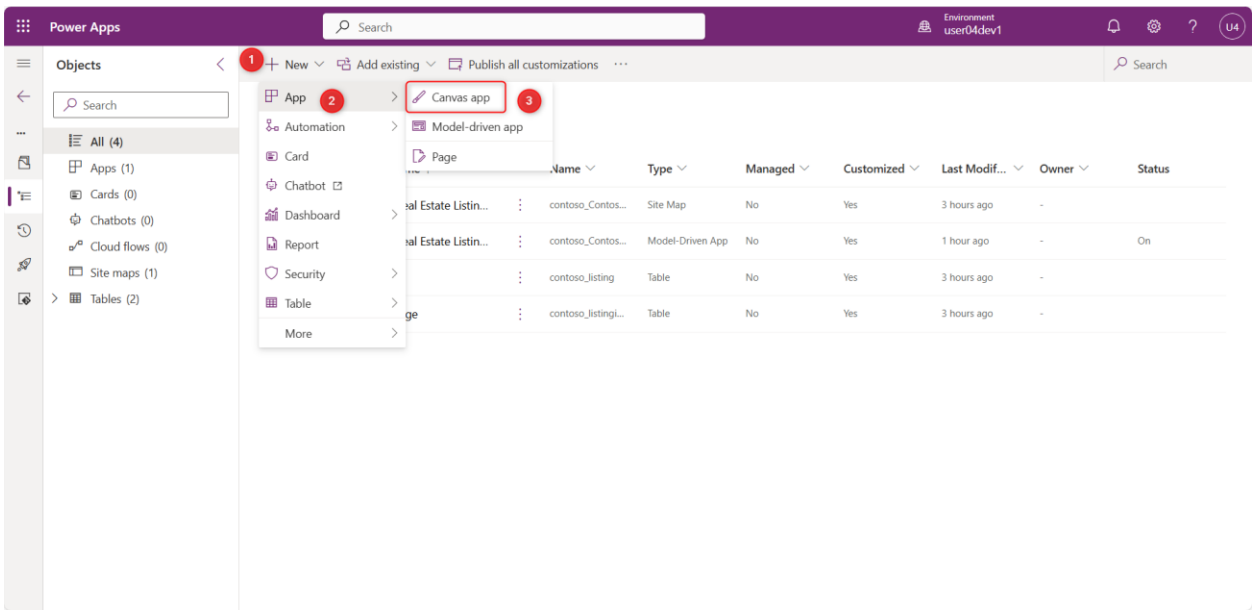


Task 3: Create a new Canvas App

In this task, you'll create a new application. This app will be used by field agents, to manage real estate inventory, create and delete pictures while in the property, from a mobile device.

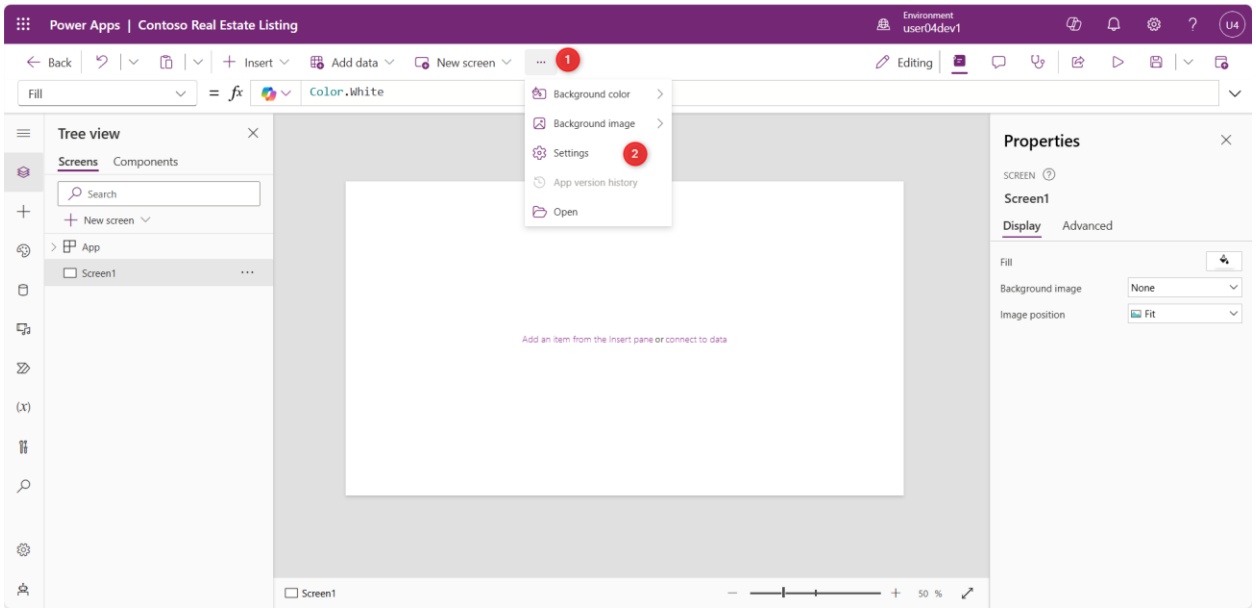
In make.powerapps.com, select Solution > Contoso Real Estate.

1. Select New, Canvas App

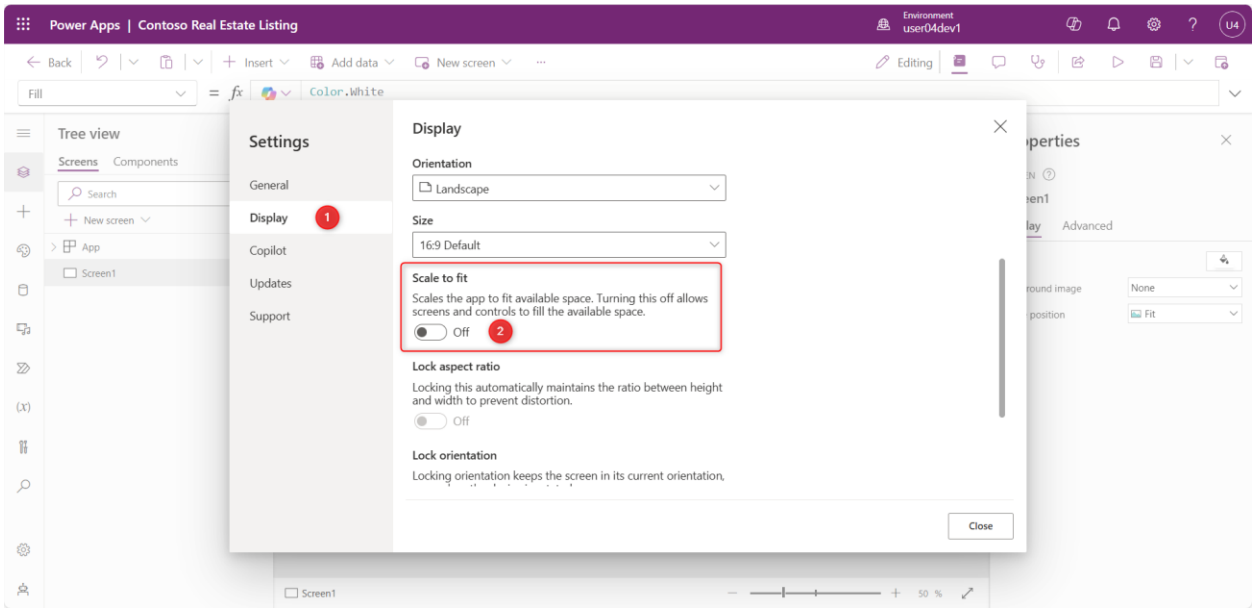


2. Type "Contoso Real Estate Listing" and click Create.

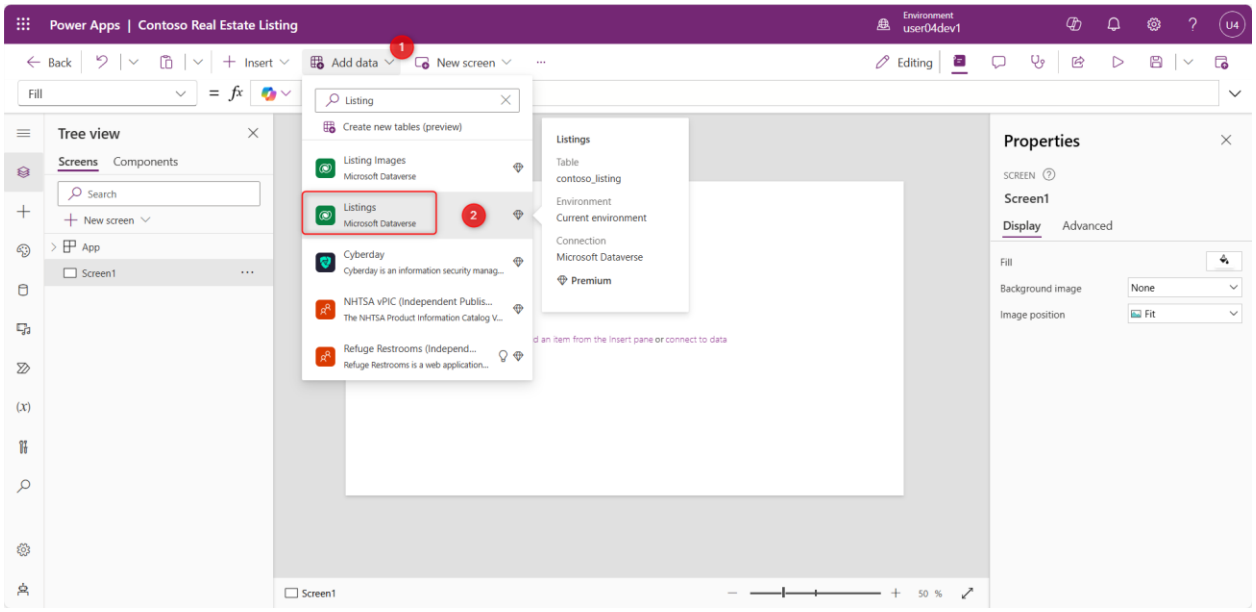
3. Open the App Settings:



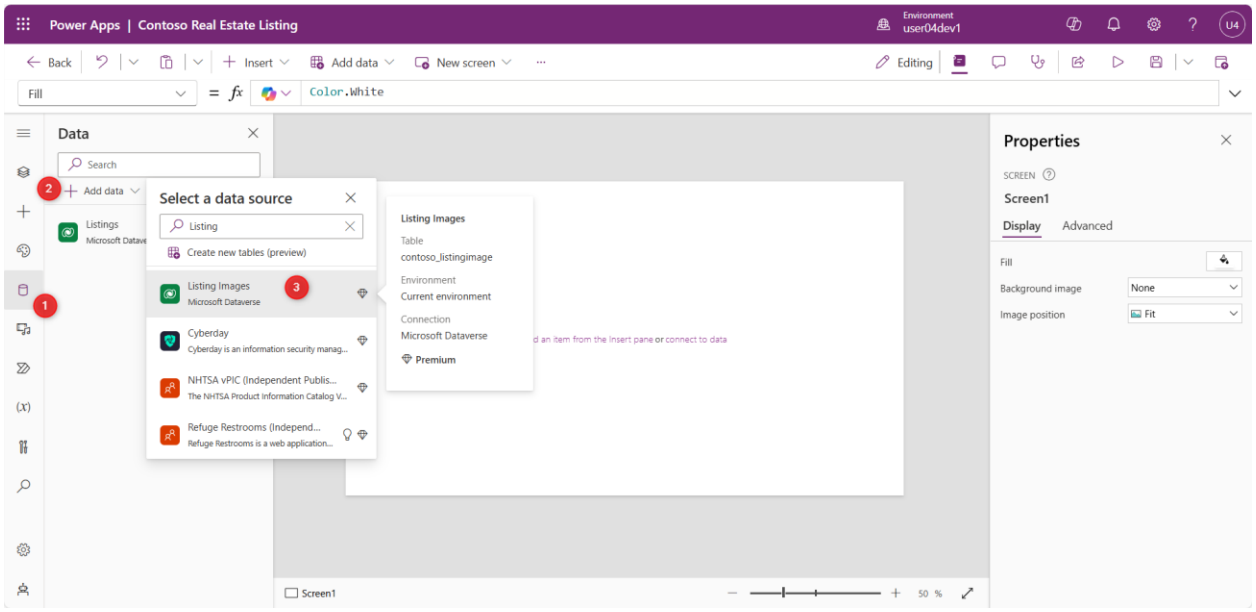
4. In Display, turn Off “Scale to fit”.



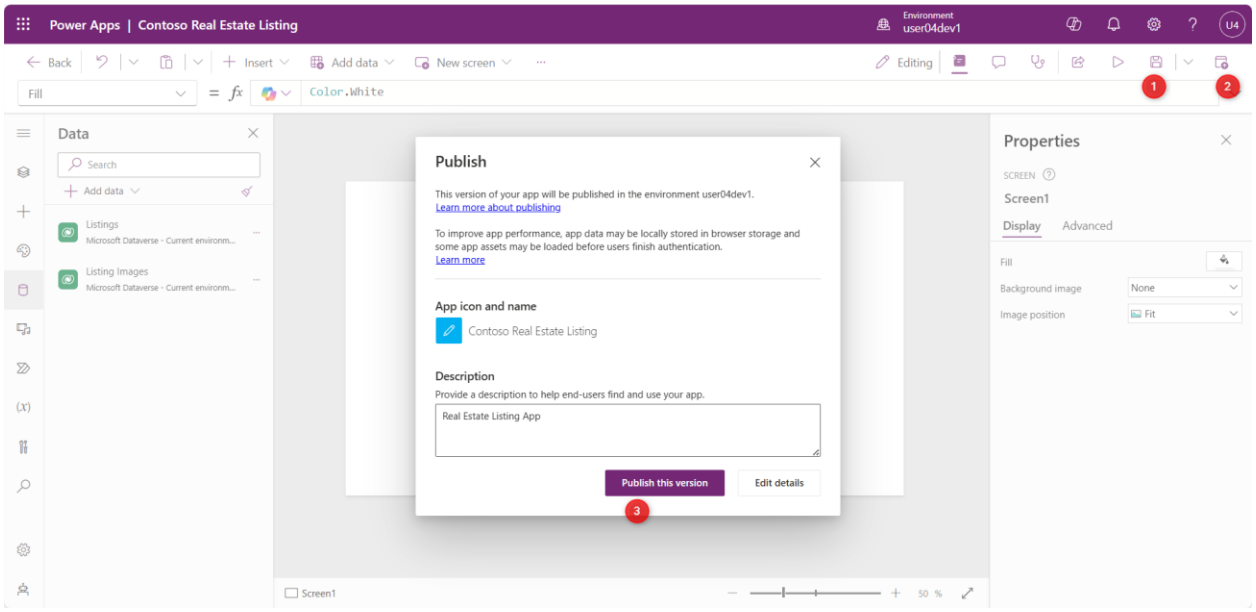
5. Click in Add data and add the “Listing”



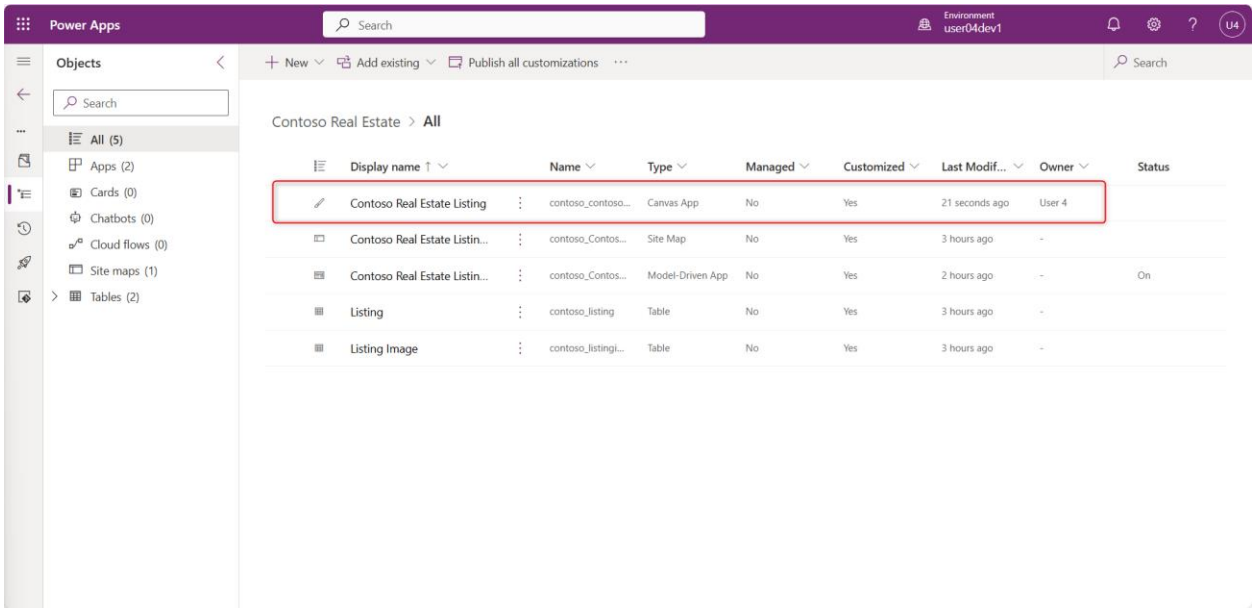
6. From the left menu, select the Database View to confirm “Listings” has been added. Click in Add data, and add the “Listings Image” Table.



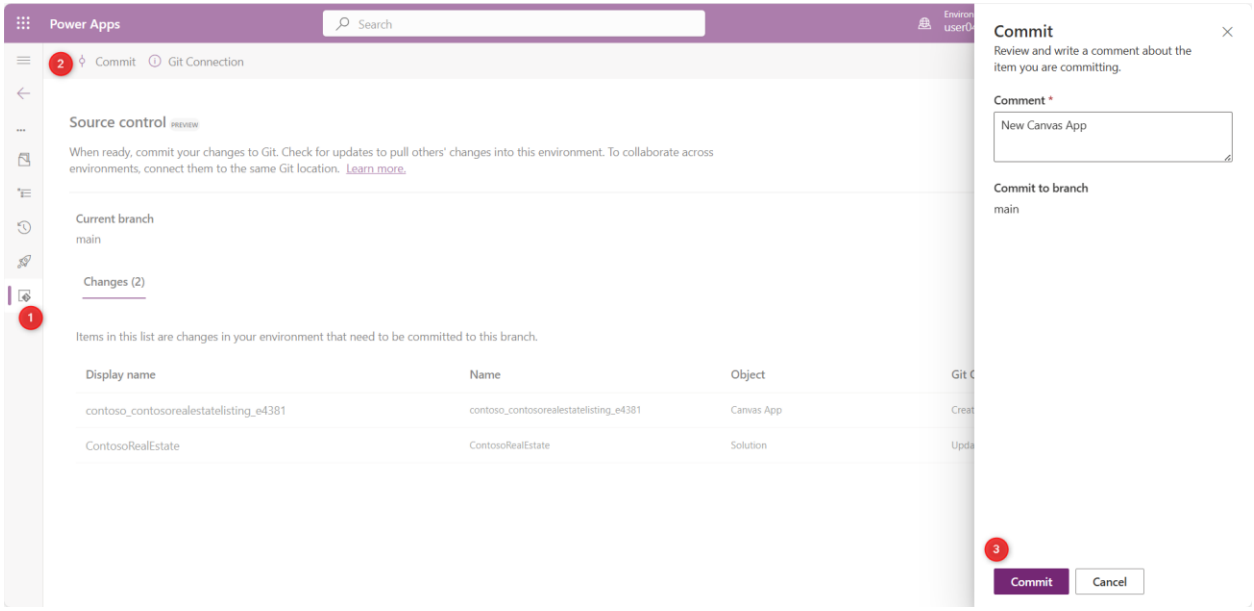
7. Save and Publish the App



8. Click “Back” to go back to maker portal. Notice the new app in the solution.



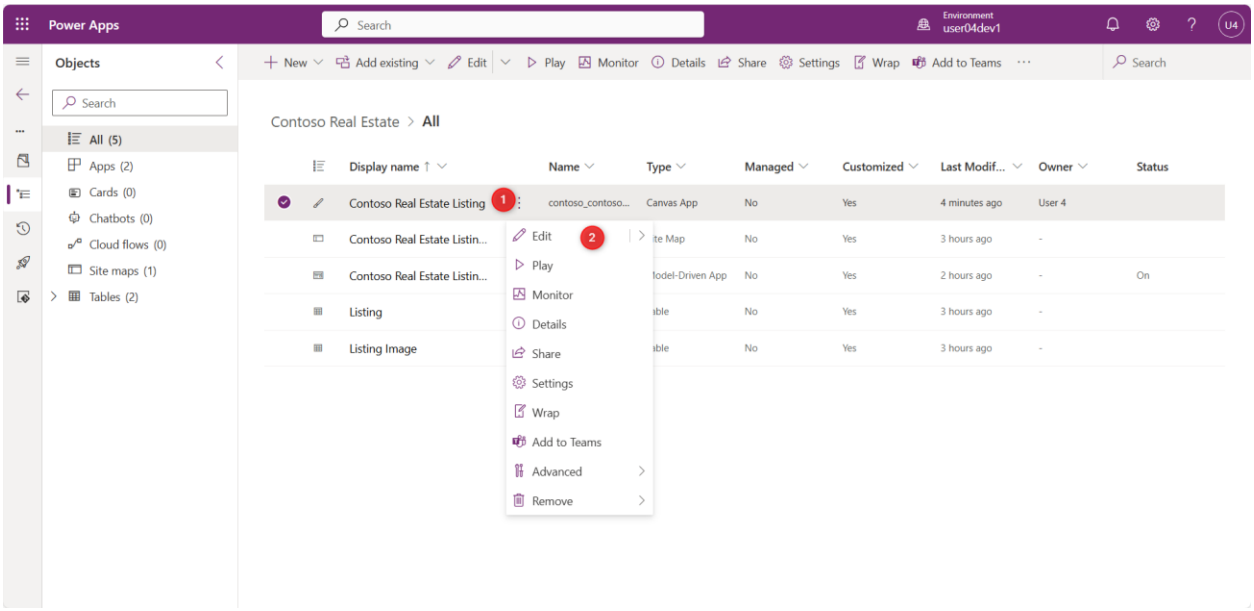
9. Click in Source Control. Notice the new Canvas App, and commit the solution



Task 4: Use YAML Code Snippets

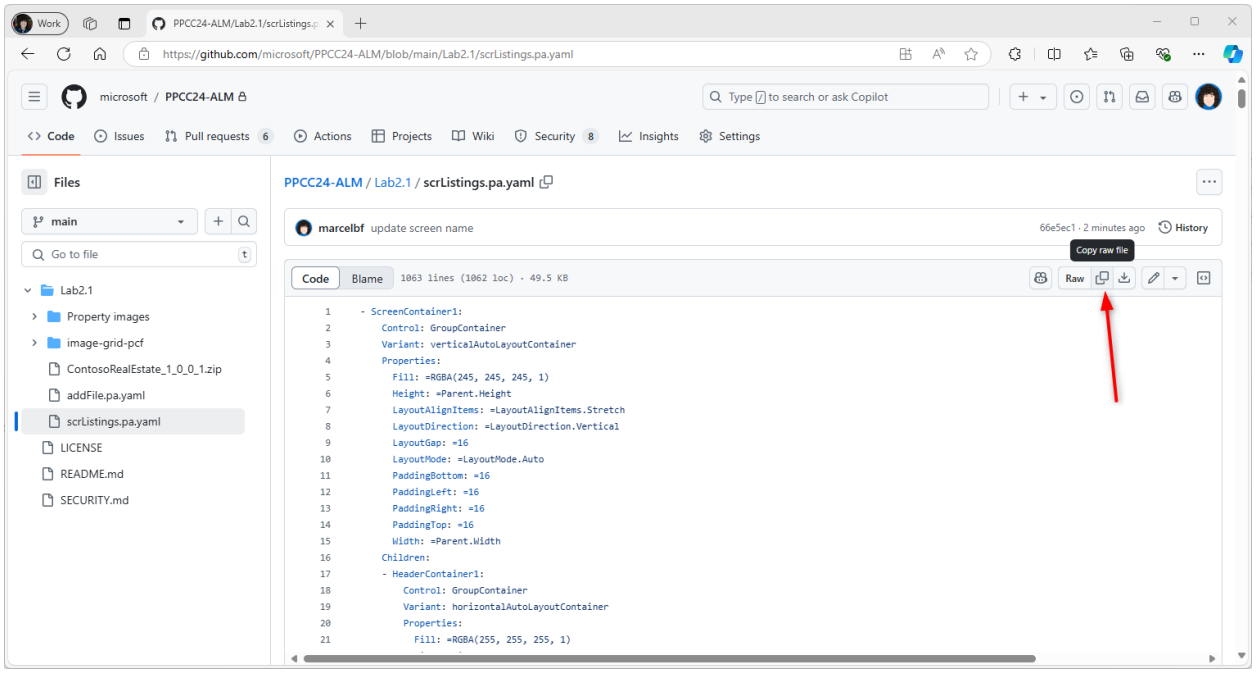
The canvas app was created, and the tables have been added. Let's use YAML code to create our user interface.

1. From Contoso Real Estate Solution, select Edit to open Power Apps Studio.

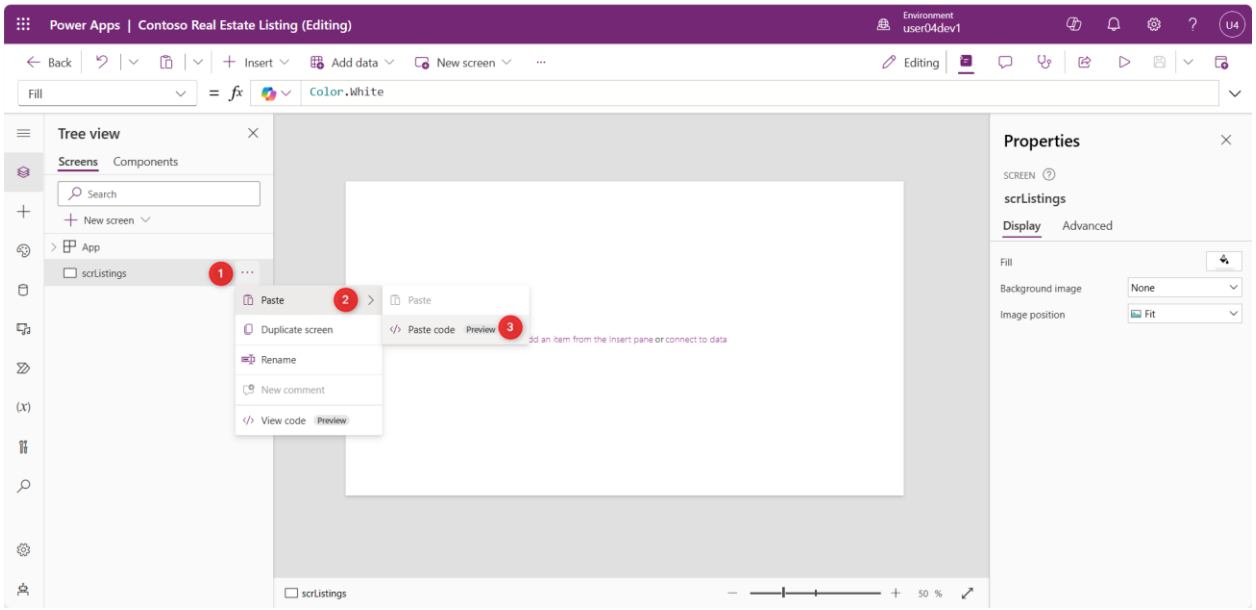


2. In Power Apps Studio, rename the screen to "scrListings" – **If you skip this step the code won't work.**

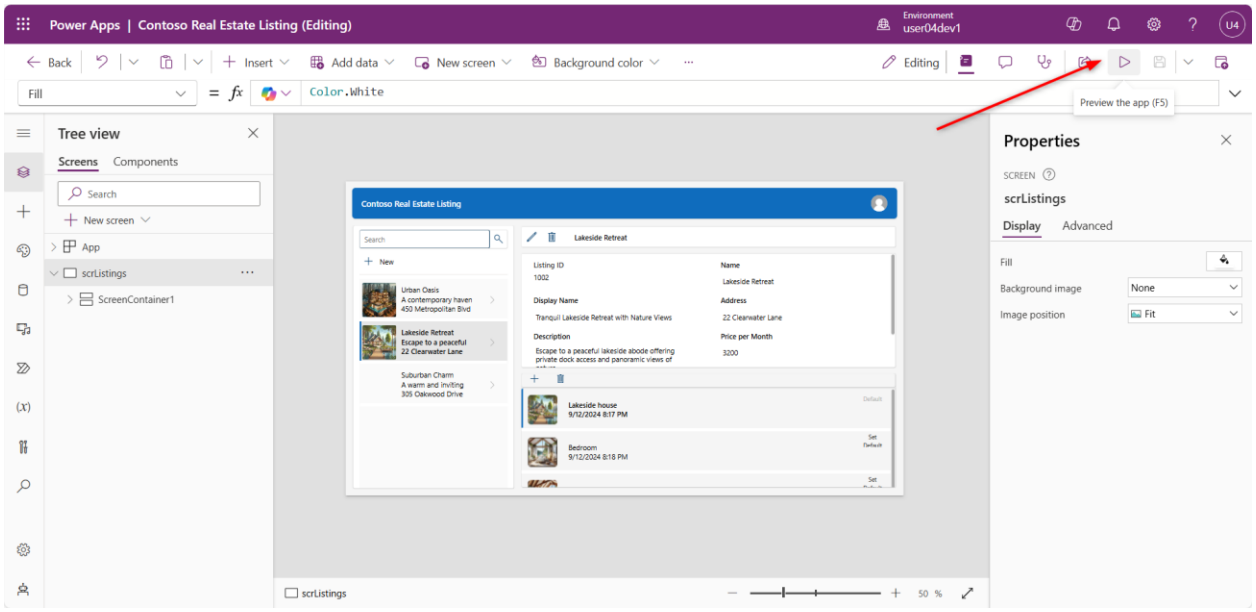
3. Open this repository, click in “copy raw”. This will copy all the YAML code.



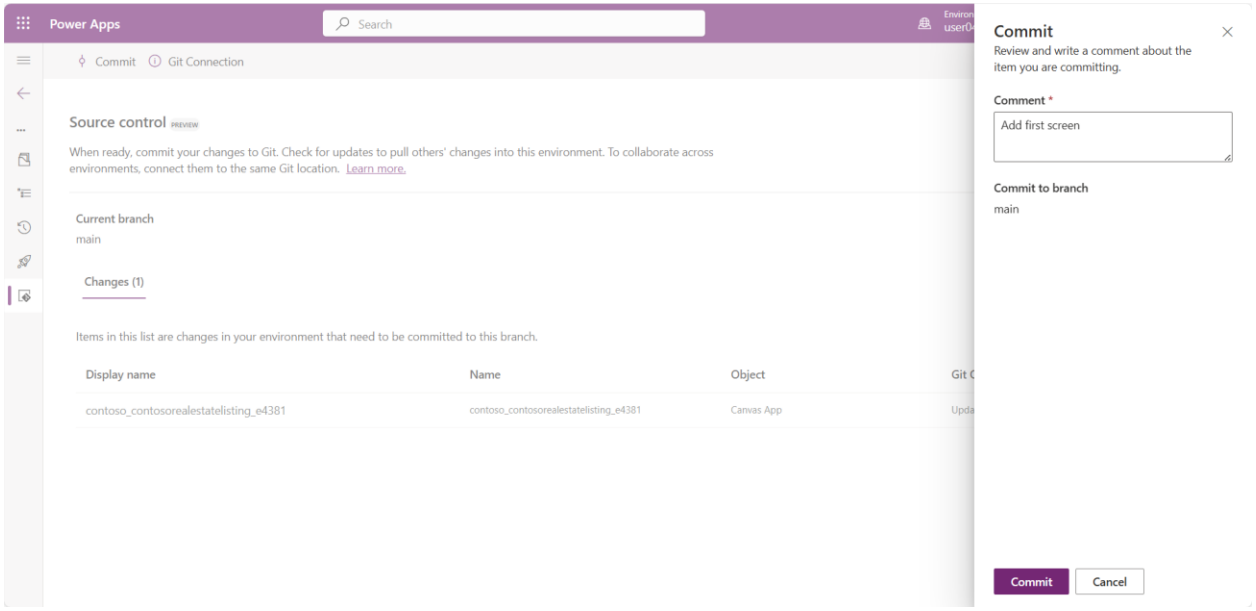
4. In Power Apps Studio, select “Paste code”



5. Run the app. For each property, you can select a picture to set as default.

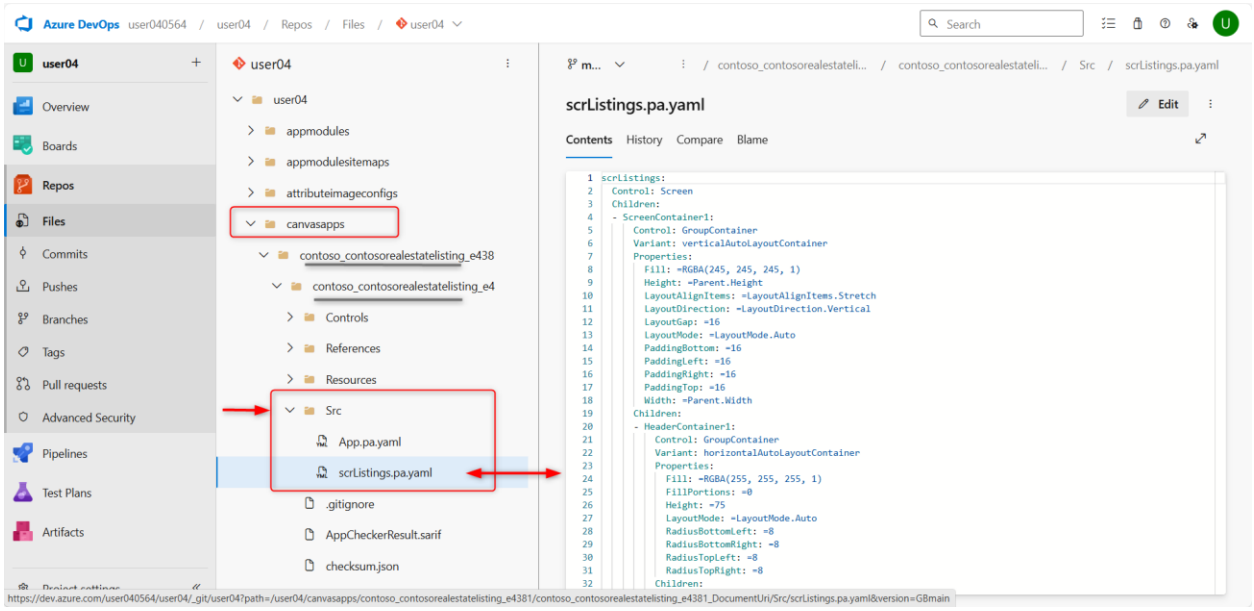


6. Save and publish the app. **You must publish your app before you can commit the code.**



7. Open Azure DevOps to visualize the code. Notice you don't have the binary .msapp file. You will find the source code for canvas within the folder src of your app as pa.yaml

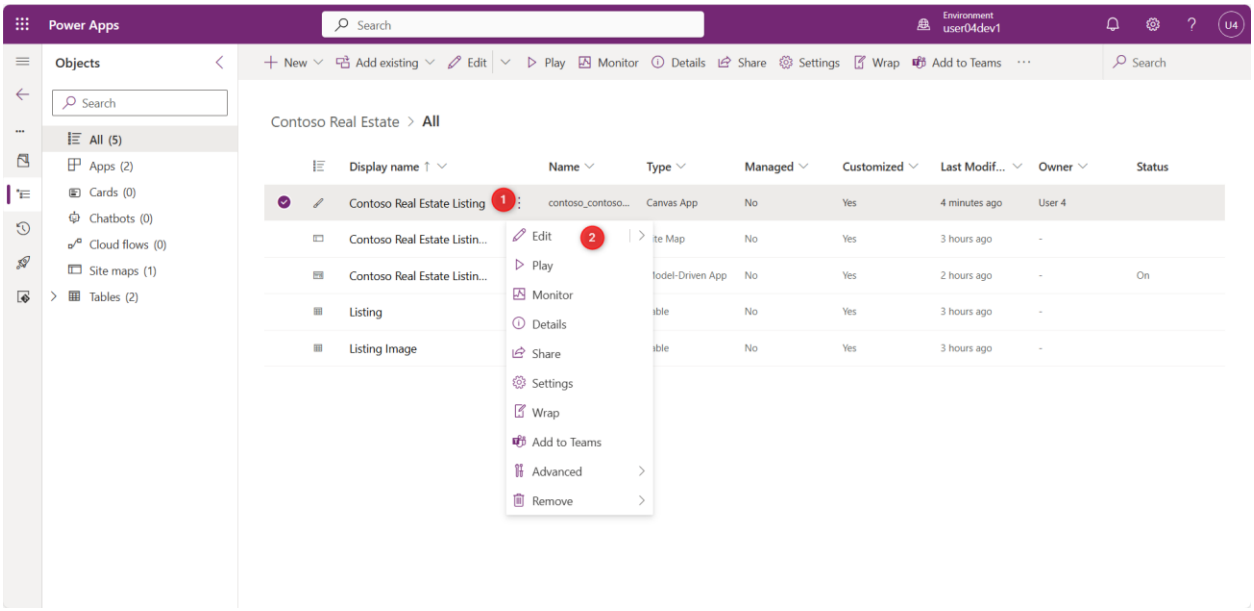
files (one file for your app, one file per screen).



Task 5: Add a screen to upload pictures

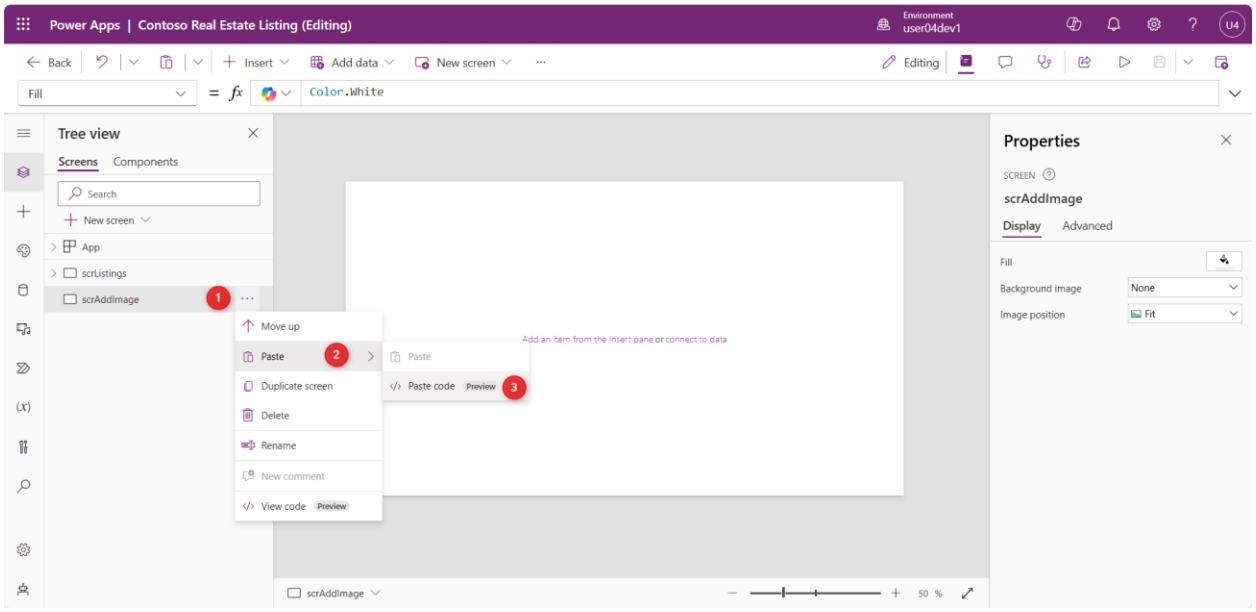
You can use YAML to create templates to reuse code. We will add a screen to upload images from another code snippet.

1. From Contoso Real Estate Solution, select Edit to open Power Apps Studio.

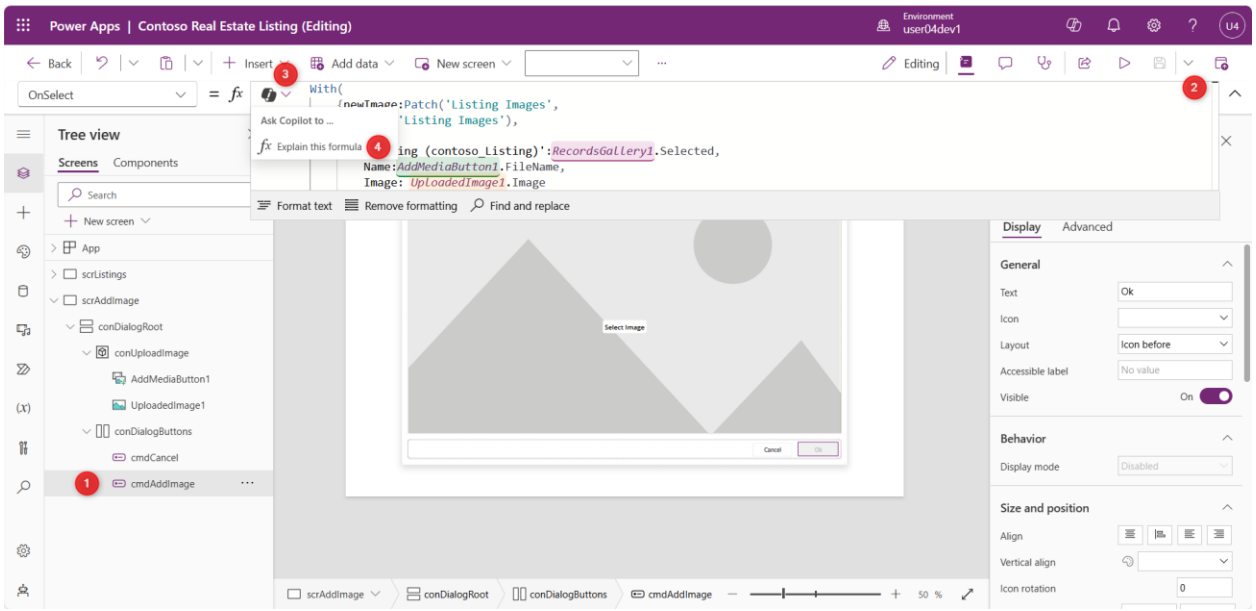


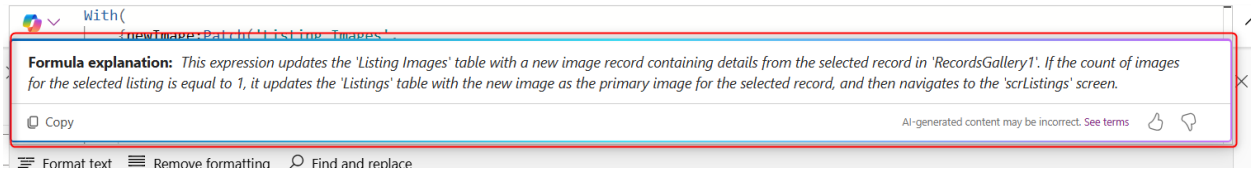
2. In Power Apps Studio, add a new Blank Screen.
3. rename the screen to “scrAddImage”
4. Open this repository, click in “copy raw”. This will copy all the YAML code.

5. Paste the code in the new screen.



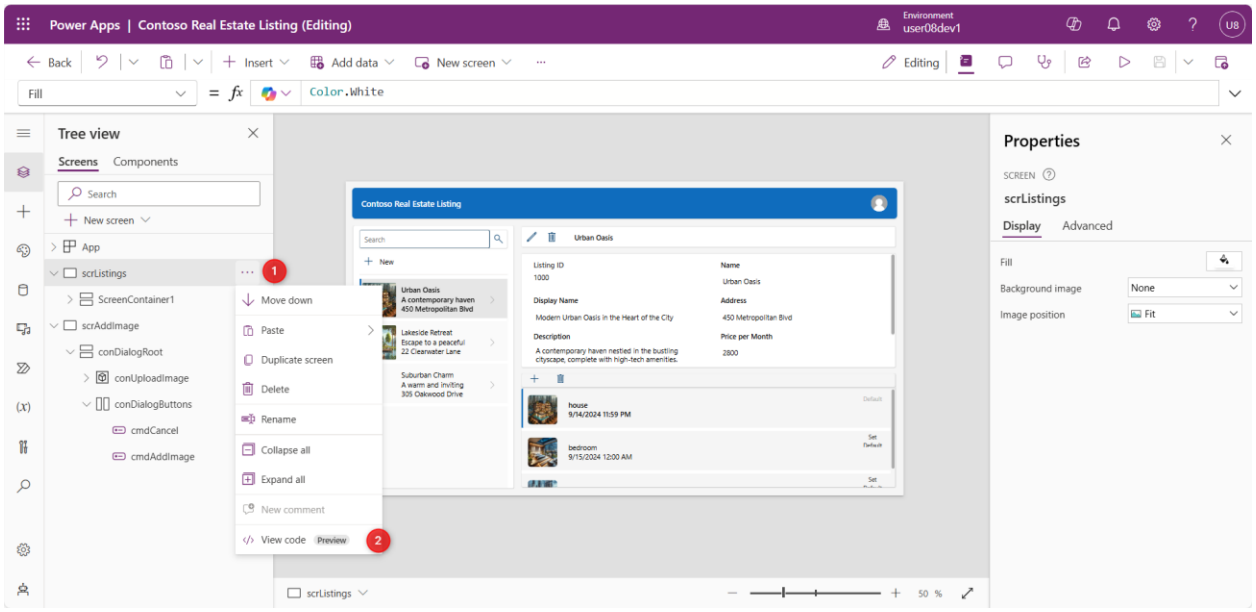
6. In the new screen, select the control cmdAddImage. In the formula bar, let's use copilot to get a plain English explanation about what it does.



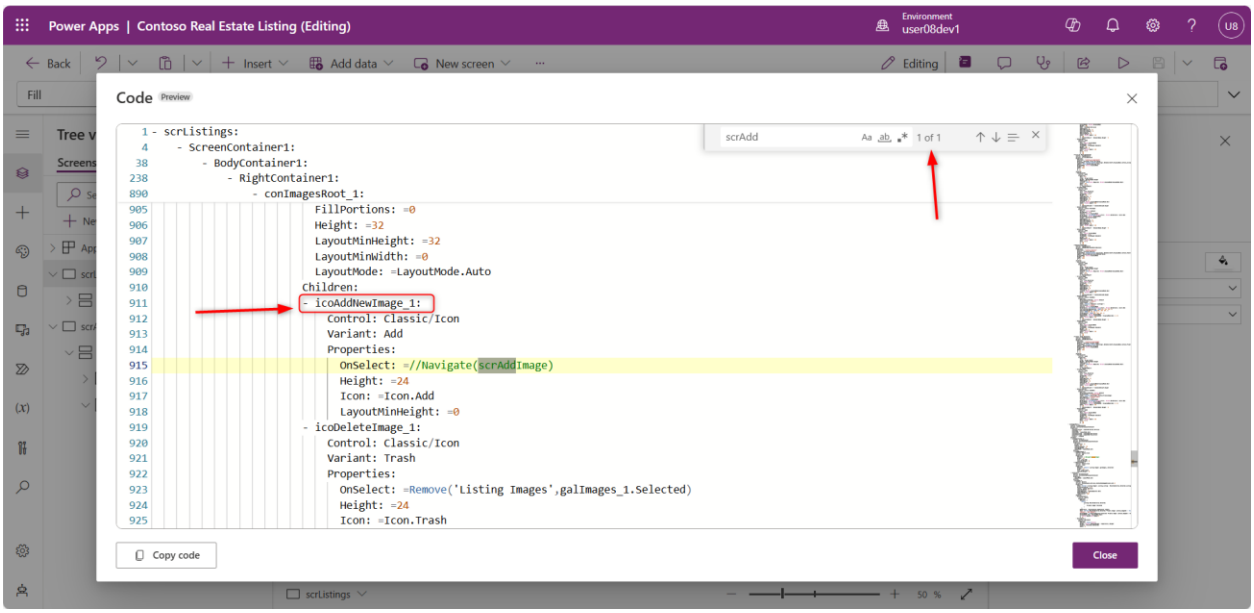


Now we need to update our code to call this screen. Our template already has the code commented. Let's find it.

7. In the tree view, in the screen "scrListings", Click in "View Code"

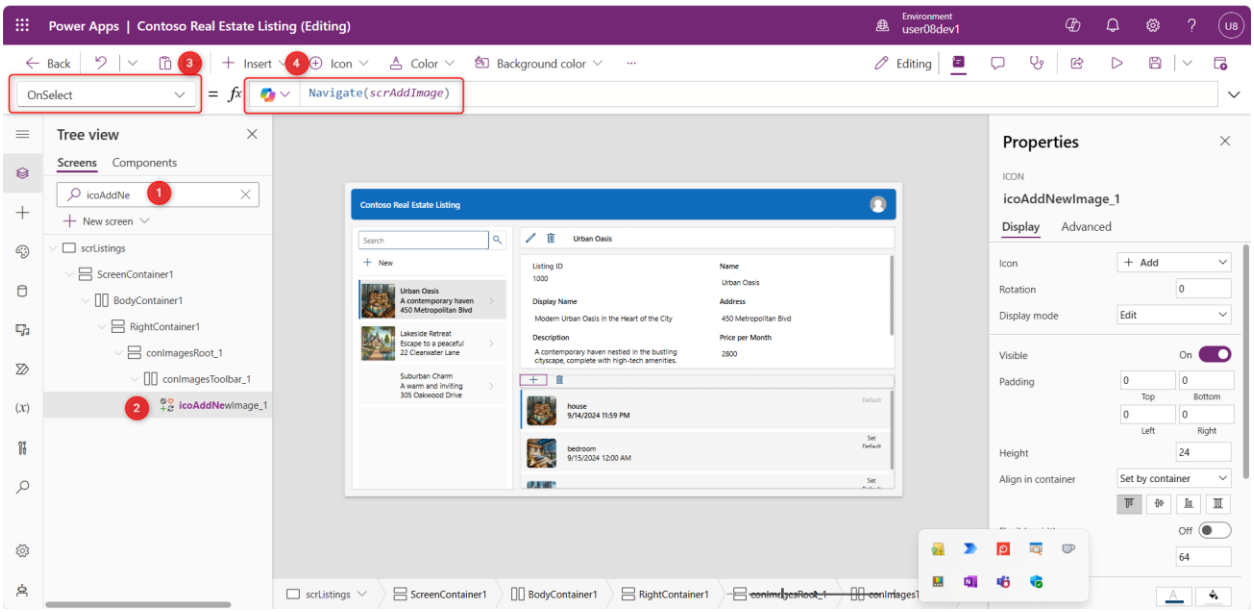


8. Press "Ctrl + F" to find, and type "scrAddImage" (this is the name of our new screen). We will find what control is using this screen.



Notice, the screen is being used by the property "OnSelect" of the control "icoAddNewImage_1"

9. In the tree view, use search to find the control “icoAddNewImage_1”. Remove the comment (“//”), so this icon will call the new screen.

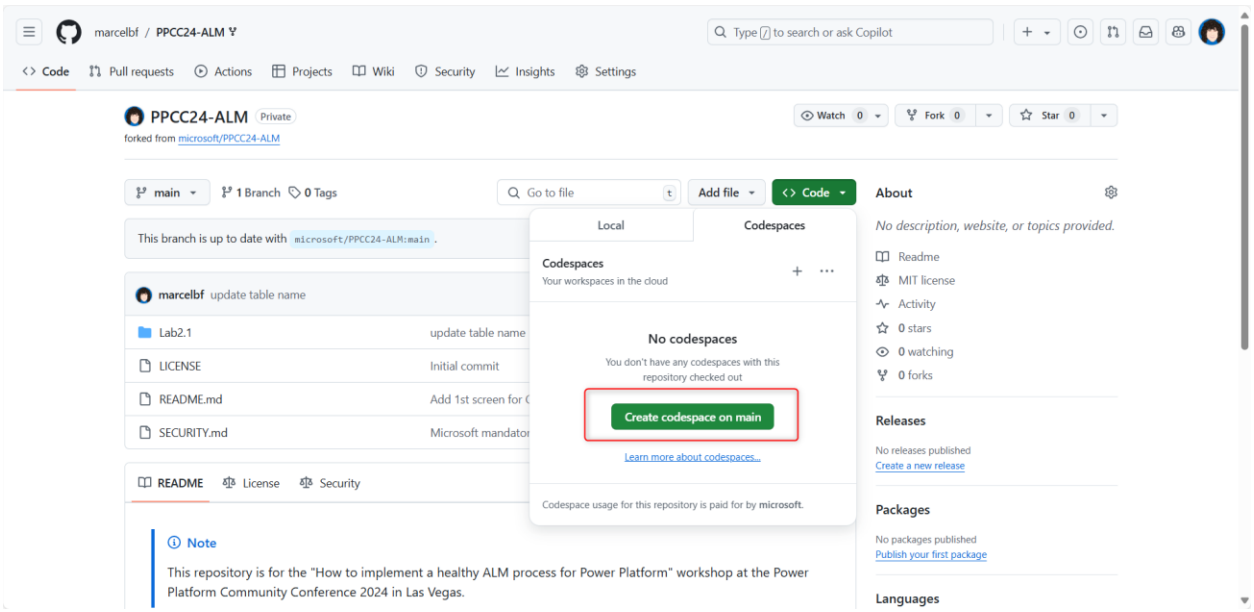


10. You can now run the app and add images to the Suburban property from PPCC24-ALM repo.
11. Save and publish the Canvas App. Commit the code to source control.

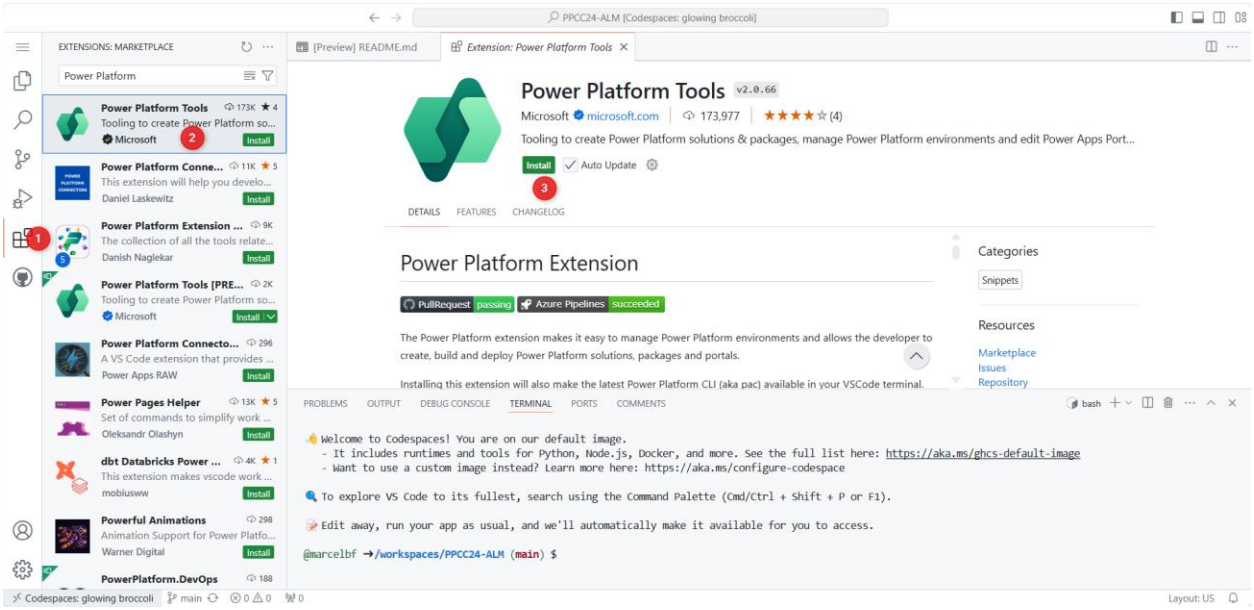
Task 6 – Build a Code Component

You need a GitHub account to complete this lab. If you need to create one follow this guide.

1. Fork the repo
2. Create a new CodeSpaces on main

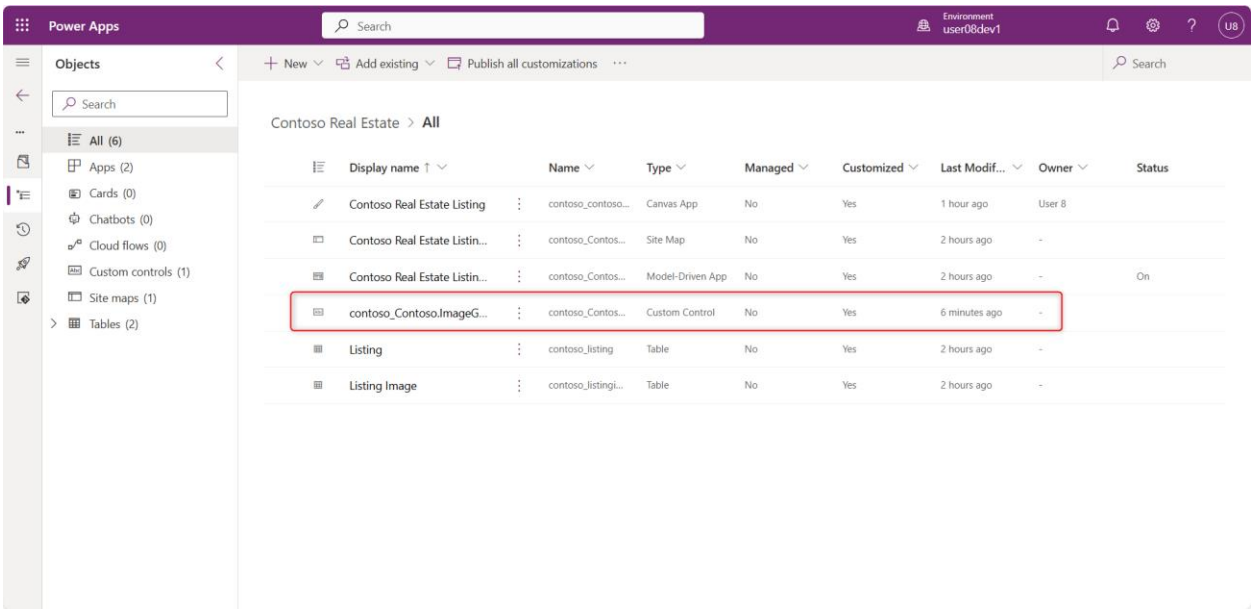


3. Install Power Platform Tools



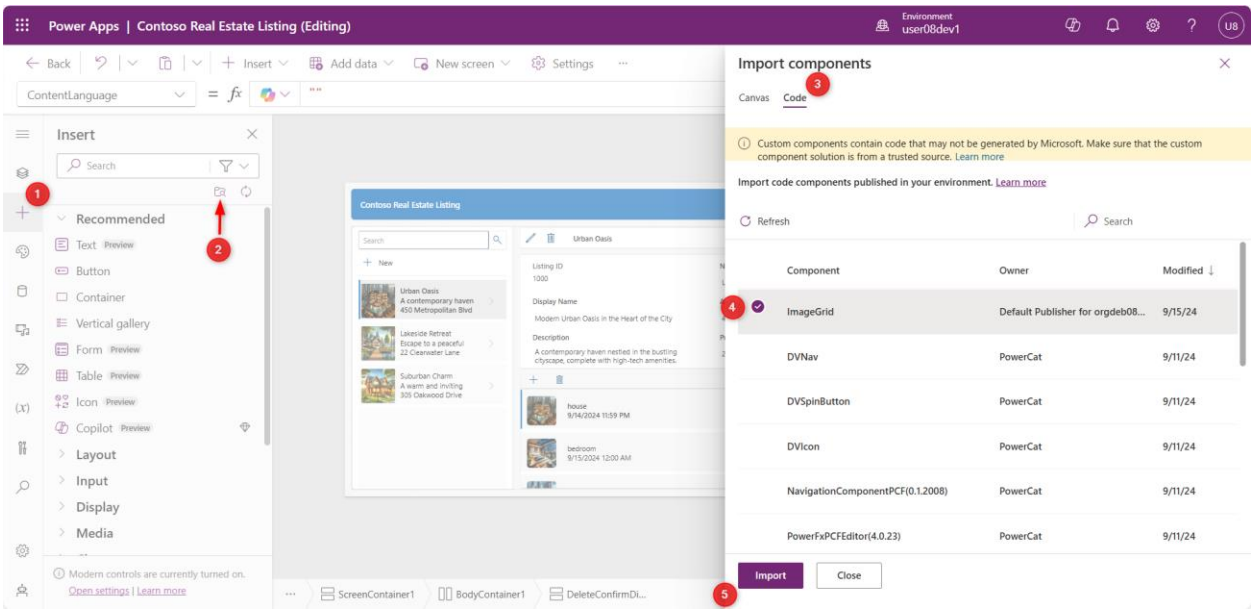
4. In the terminal, type “pac auth create” - and login with you user from the lab
5. In the terminal, type “pac env select --environment <url of your environment>“ to select your environment (you can check the url by running the model driven app)
6. In the terminal, type “pac env update-settings -n iscustomcontrolsincanvasappsenabled -v true” to enable Code Controls in your environment.
7. “cd Lab2.1/image-grid-pcf/”
8. Build the project “dotnet build”

9. Push the component to your solution “pac pcf push --solution-unique-name ContosoRealEstate”

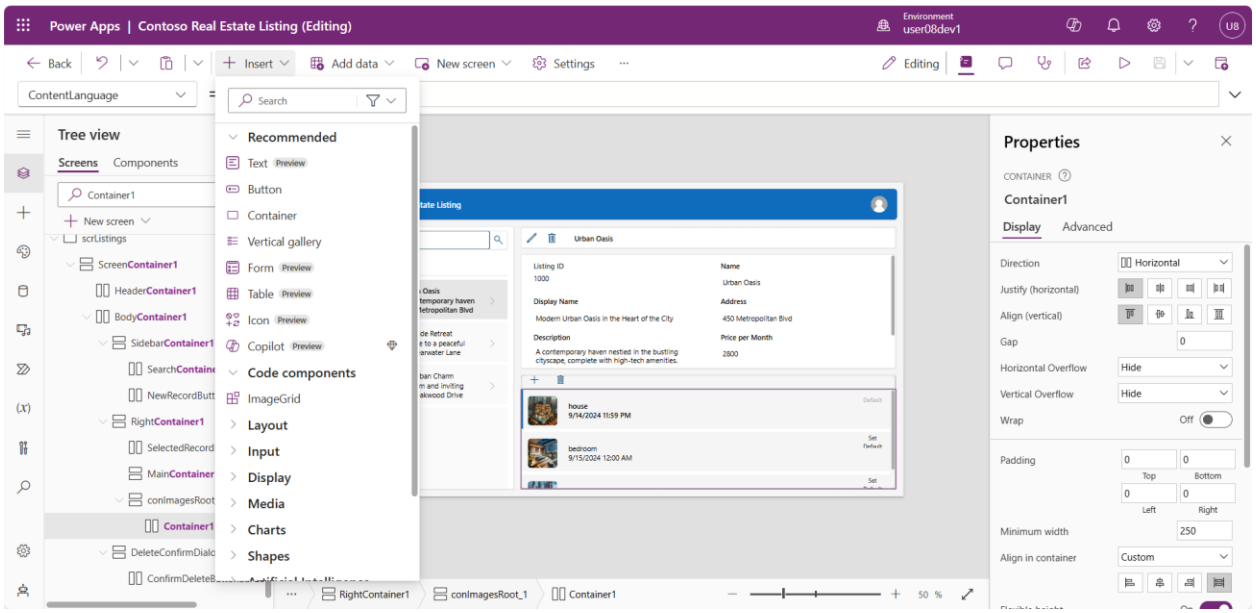


Task 7 - Add the Code Component to you Canvas App

1. Edit your canvas app.
2. Import the Code Control



3. Add the ImageGrid (code control) in the “Container1”



4. Set the properties of the control – or, delete the ImageGrid you added manually, and paste the code from the repo.
5. Commit the code – notice that you can't commit the Code Component.