

Programming Boot Camp

Bubble Basics

Tokyo Institute of Technology 2022/11/19

Naotake KYOGOKU

Table of Contents

- What is Bubble?
- First, let's create a user registration (P28 -)
- Overview of application development with Bubble (P44 -)
- Let's create a list of pets from registration (P57 -)
- Let's create a pet detail screen (P162 ~)
- Let's make a header part (P201 -)
- Let's manage your pet's weight (P273 -)
- Summary (P303 ~)

What is Bubble?

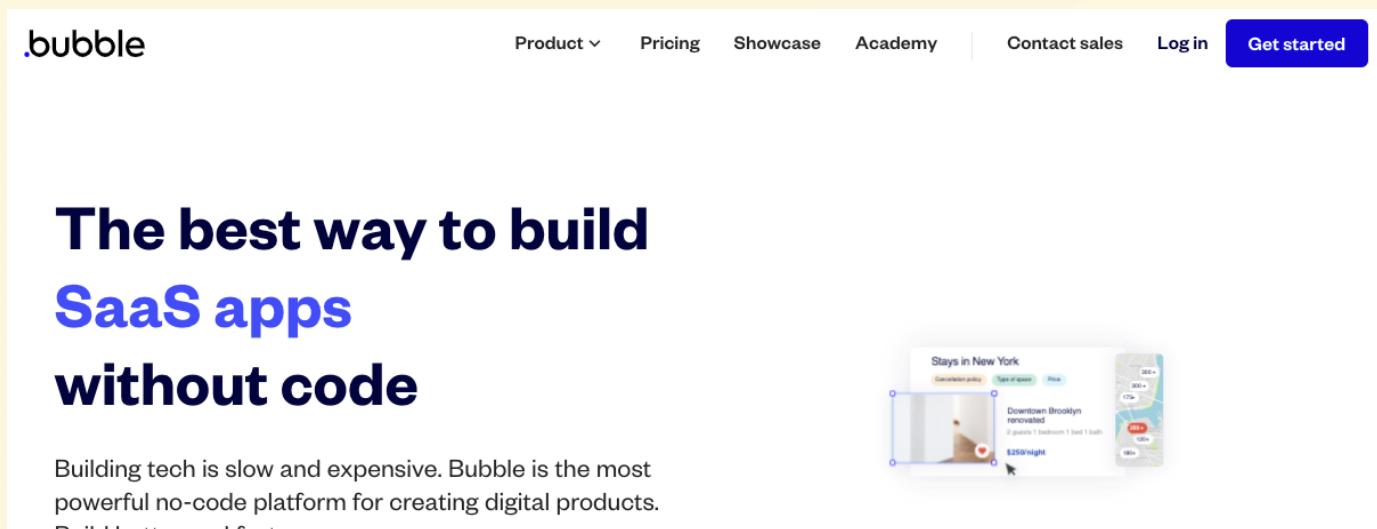
- Bubble](<https://bubble.io/>) is called a visual programming tool, and although it is a no-code tool, it requires programming ideas.
- The operations are the same as those of Adalo, such as selecting parts you want to use from the available parts and dragging and dropping them on the screen.
- However, the part of adding movement to the placed parts requires a programming mindset.
- Also, unlike Adalo, the development is done on the premise of a web application, and then it is adapted for smartphone display.

Examples of applications made with Bubble

- Ai Home Virtual Showings: <https://aihome-vr.com/>
 - Online property viewing app
 - This is an application that allows you to preview properties online in a street view-like format.
 - Reference: <https://note.com/apopopo/n/n155b0df7f78c>
 - It describes the knowledge of no-code development rather than Bubble's know-how.
- There are many more out there, look them up under "Bubble Case Studies" etc.!
 - Example: <https://nocodedb.world/archives/3337>

Register with Bubble

- Register an account by entering your email address from the Bubble's TOP screen
 - <https://bubble.io/>
 - Click "Get started"



- Register for a free account
- Enter your email address and password and click "Start building"

SIGN UP

No credit card or coding skills required.

 Log in with Google

Email

example@gmail.com



Password

By signing up, you agree to our [Terms](#) and [Privacy Policy](#).

Start building

Log in

- You will be asked what your goal is for today, select "BUILD" and Next

Hi there, what's your goal today?

Let's get you started faster!

BUILD

Create an app for myself or my business. Free to start building.

HIRE

Find someone to build my app for me. Service starting at \$200/month.

EARN

Make money by building apps for others. Typically around \$80+/hour.

..

Next

- You will be asked where you learned about Bubble.

Where did you hear about Bubble?

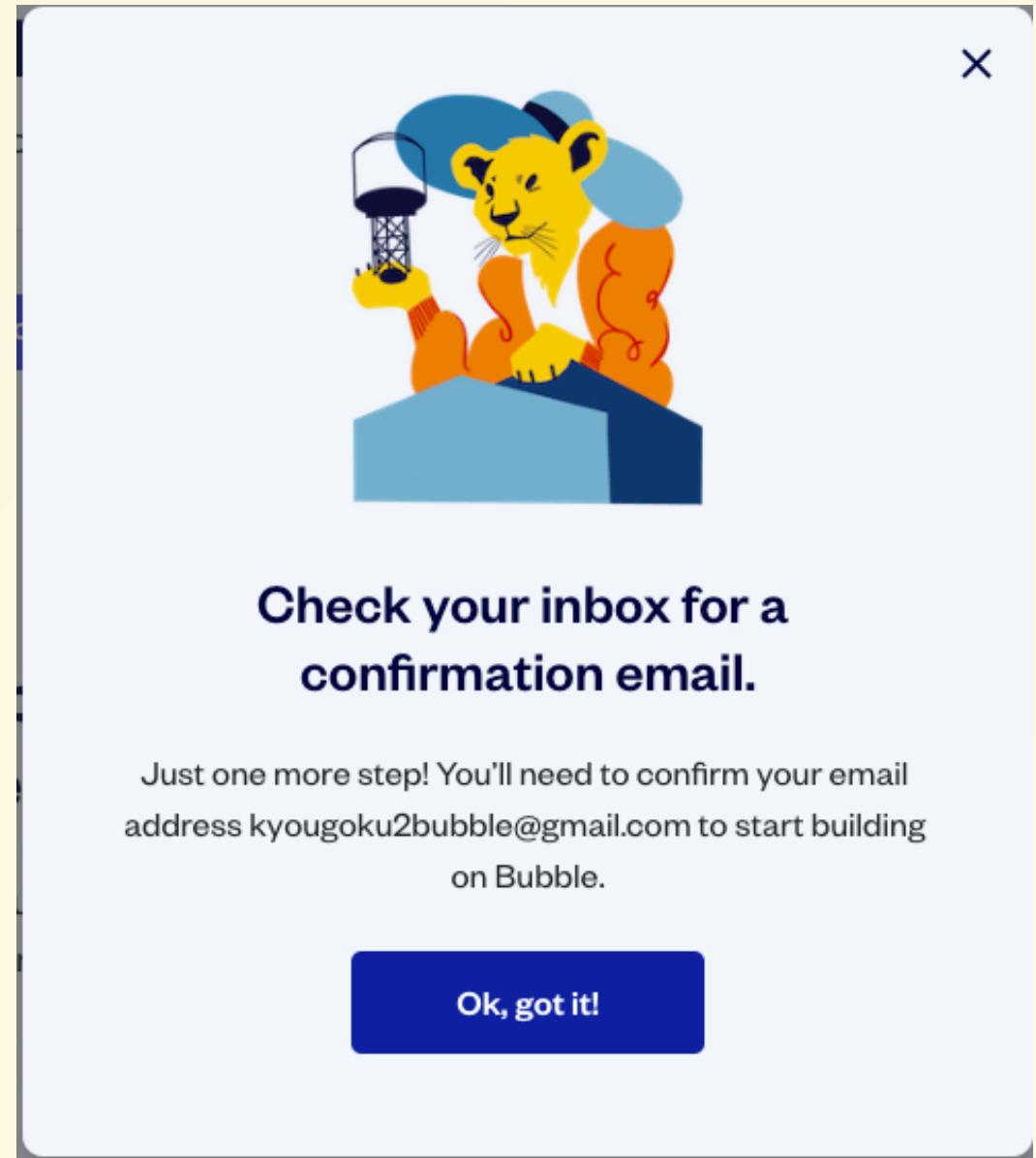
Just before you get going!

Friends or colleagues Discovery platforms (PH, Quora...) Course
 Search engines Facebook/Instagram Incubator
 Press Twitter Forums (Reddit, Hackernews...)
 Blog LinkedIn Other
 Podcast Other social networks

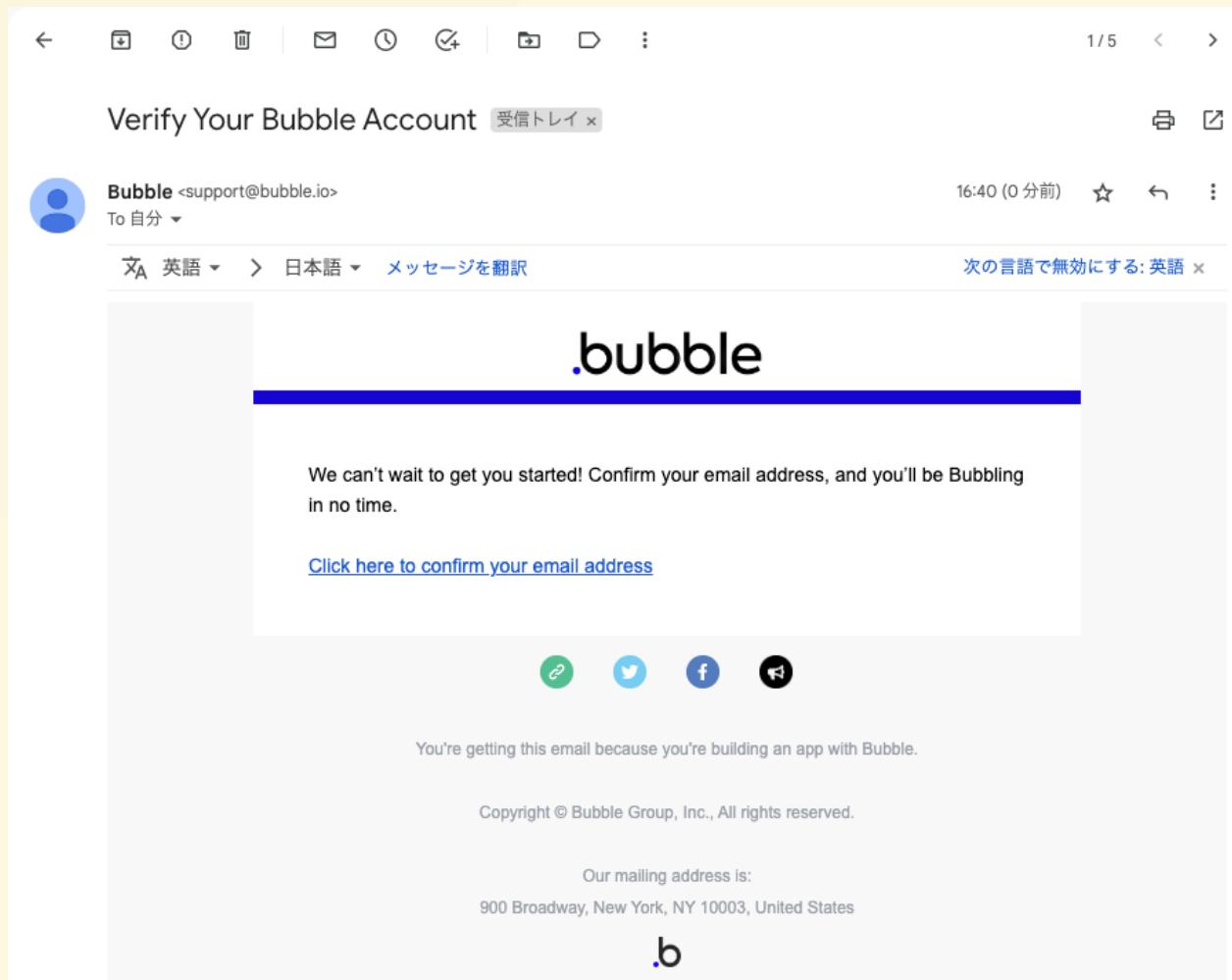
[Previous](#) • • • [Complete](#)



- A confirmation email will then be sent to the email address you entered, so check your mailbox.



- You should have received an email from Bubble titled "Verify Your Bubble Account", click on the "Click here to confirm your email address" link in the body of the email.



- If the Bubble screen appears and says "Thanks for confirming!"
- Click "Start building"

Thanks for confirming!

You're ready to build amazing web applications
with Bubble!

Start building



- The Welcome page will appear, give a name for the exercise you are going to create and click "Create app".

Welcome to Bubble

Let's set up your new app.

Create a new app

Name your app
titech2022-bubble-kyogoku

Your Bubble URL ⓘ
titech2022-bubble-kyogoku.bubbleapps.io

I agree to the [Bubble Terms of Service](#) and [Privacy Policy](#).

Create app Explore templates on the marketplace



- However, you cannot create an application with the same name as others (around the world), so use the following naming convention
 - **titech2022-bubble-{Your Name} .**
 - If no error message is displayed after entering the app name, OK.

Welcome to Bubble

Let's set up your new app.

Create a new app

Name your app

Your Bubble URL ⓘ
titech2022-bubble-kyogoku.bubbleapps.io

I agree to the [Bubble Terms of Service](#) and [Privacy Policy](#).

Create app Explore templates on the marketplace ↗



- You will be asked for a plan for the app you are about to create, click on "Start building" for Free.

The screenshot shows a comparison between two plan options for the app "titech2022-bubble-kyogoku".

Choose a plan for "titech2022-bubble-kyogoku"

Upgrade or downgrade at any time.

Free

- ✓ Core platform features
Access to basic features for testing and learning.
- ✓ Host on a Bubble domain
Preview your app on titech2022-bubble-kyogoku.bubbleapps.io.

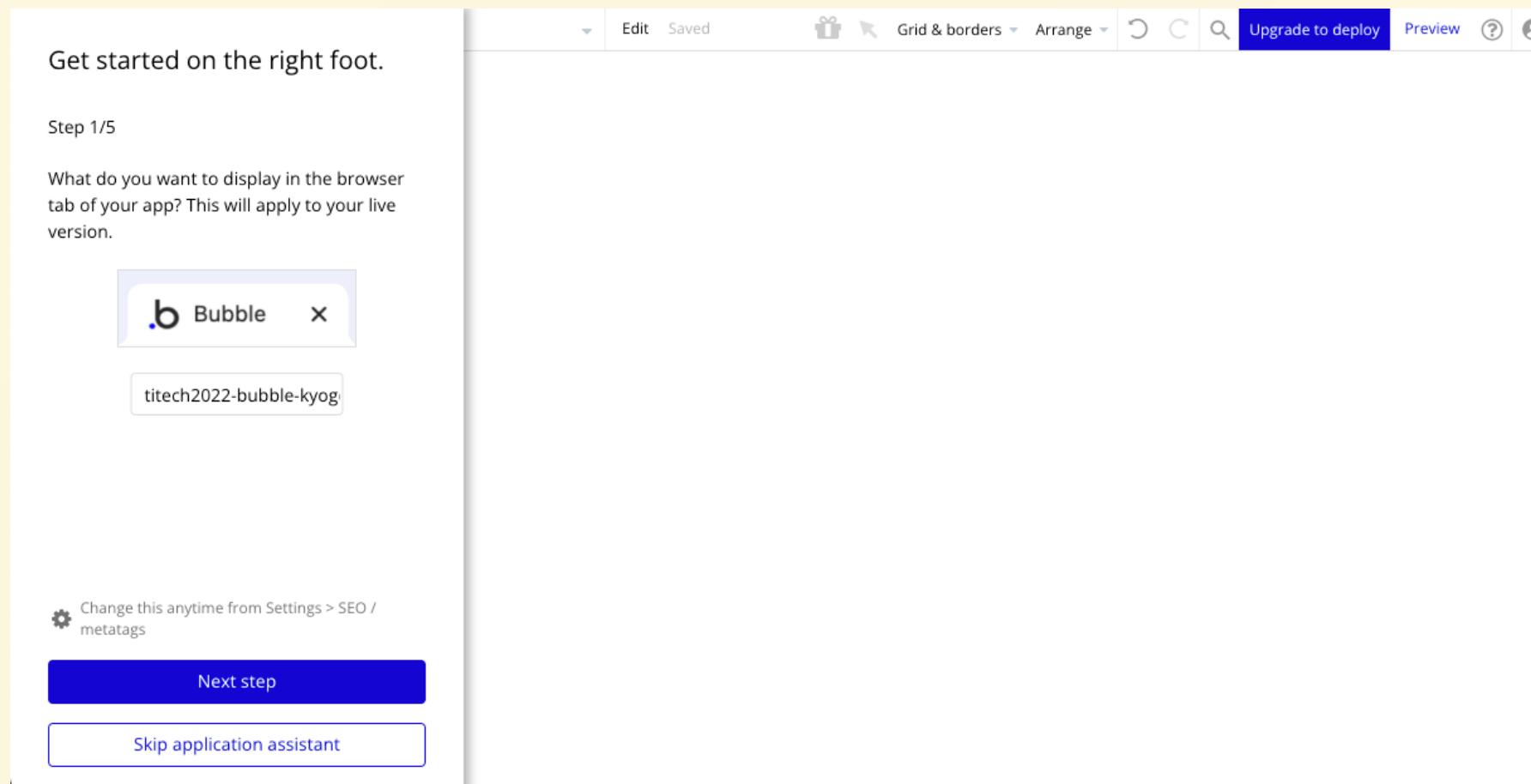
Personal

- ⊕ Custom domains
White-label your app with a custom domain name.
- ⊕ Publish, host, and scale
Launch your site to real-world users and grow with ease.
- ⊕ Infinitely extensible
Connect to Stripe, Google Maps, Sendgrid, and more.
- ✗ Automated actions
Set recurring workflows to automate backend processes.

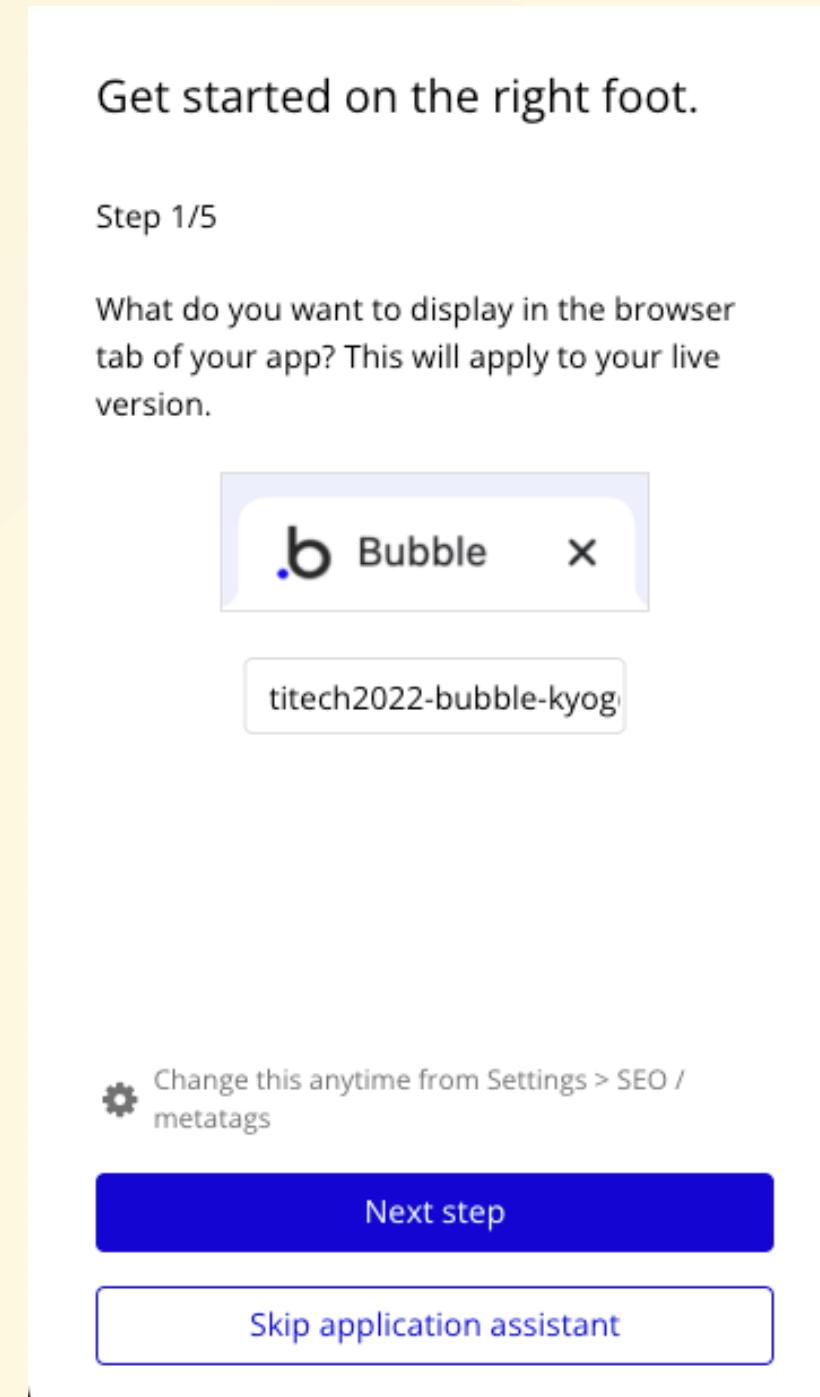
Start building Activate 14-day free trial

\$0 Due today, then \$29 per month. Cancel anytime.

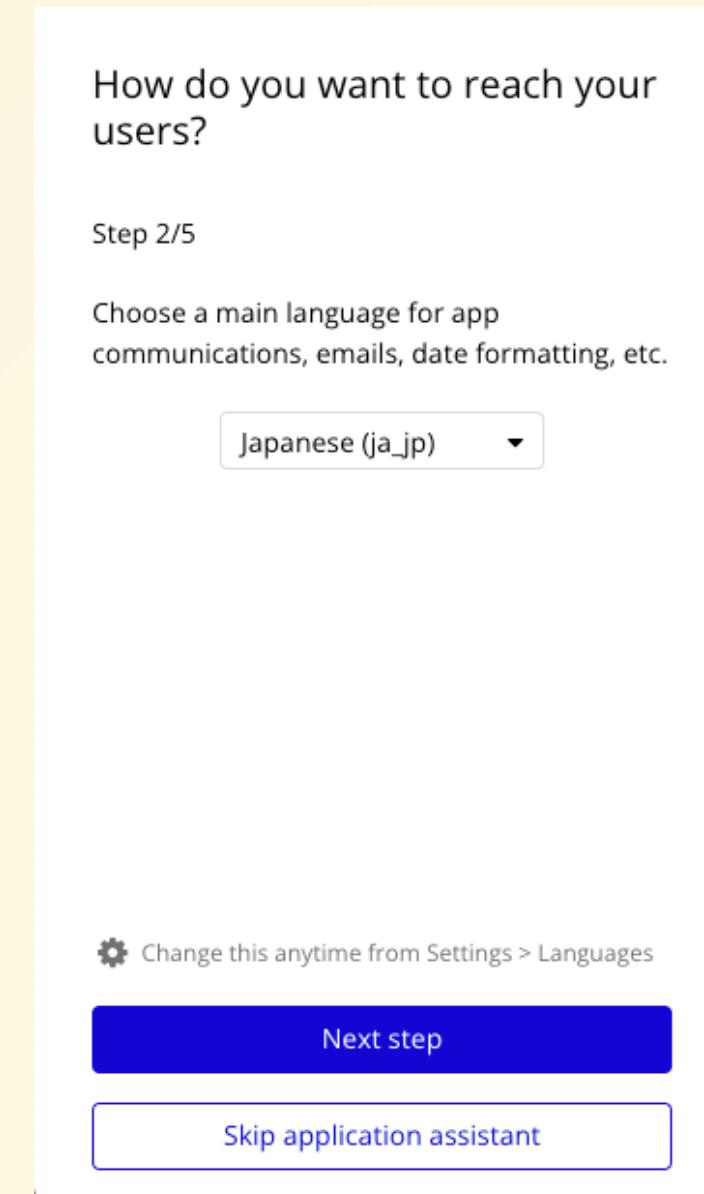
- Once created, the Bubble editor screen will appear and you will be guided through the initial setup.



- Step 1/5 asks you for the name of the web application you are going to create and display on the screen.
- The application name you have just entered is probably the default value, so leave it as it is and click OK.
- Click Next step



- Step 2/5 select the main language of the app
- Select your preferred language and click Next step

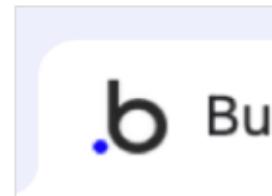


- Step 3/5 is to select an application icon (favicon).
- This time, we will proceed with the standard bubble icon, so click Next step as it is.
 - If you have enough time, you can set your favorite icon 

What icon should users see in the browser tab?

Step 3/5

Upload a favicon for users to see next to your app name.



 Change this anytime from Settings > General

Next step

Skip application assistant

- Step 4/5 selects whether the app should be connected to another application
- Select "Install API Connector" to connect in a later step

Do you want to connect your app to third party API's?

Step 4/5

Install the API connector to connect to almost any external API from Bubble.

[Install API Connector](#)

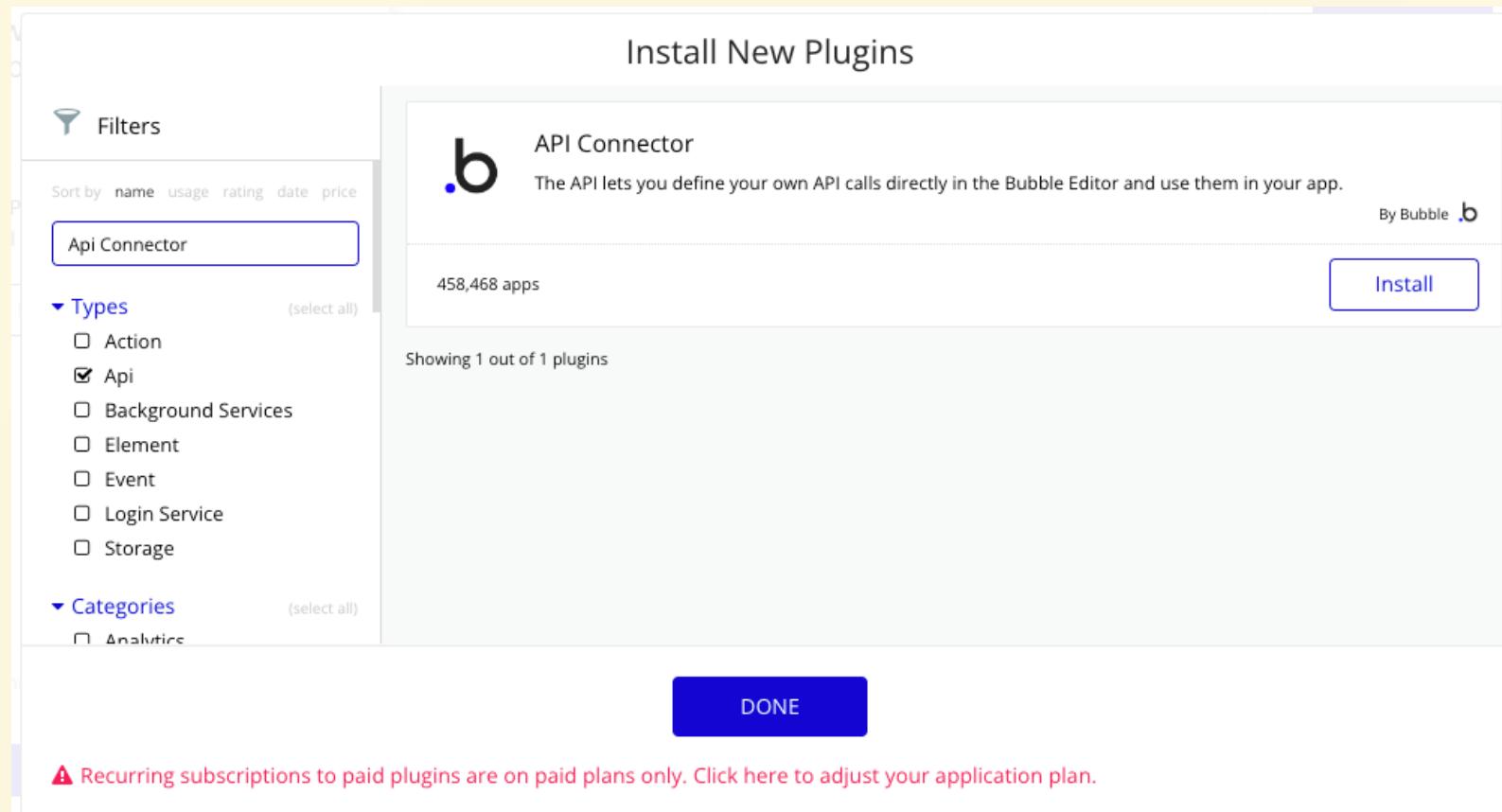


Change this anytime from Plugins > Add new plugin

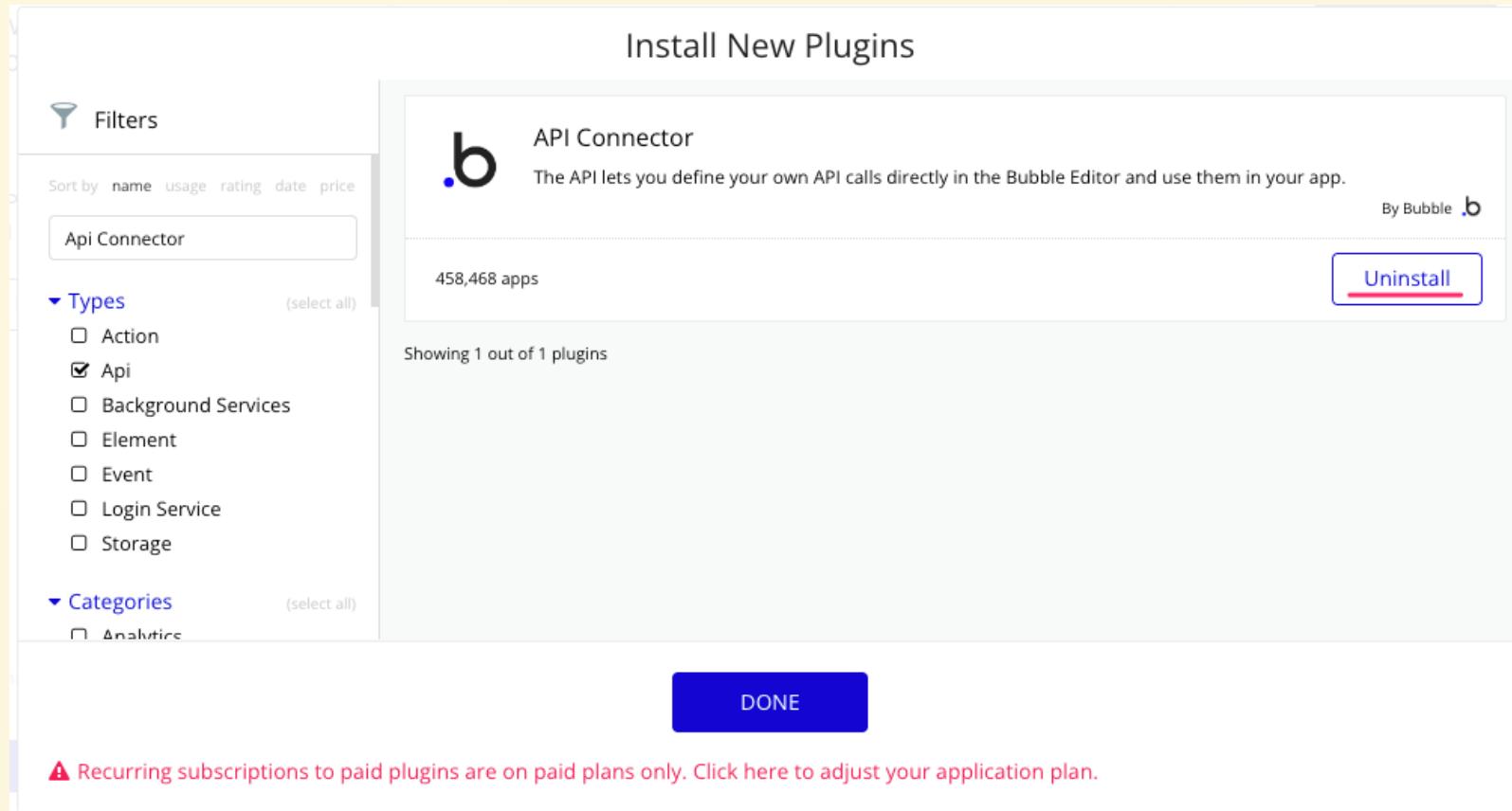
[Next step](#)

[Skip application assistant](#)

- Then a pop-up window called "Install New Plugins" will appear, where you will see "API Connector" and click on "Install".



- When the button changes to "Uninstall", click DONE.



- When you come back to the original screen, click Next step.

Do you want to connect your app to third party API's?

Step 4/5

Install the API connector to connect to almost any external API from Bubble.

[Install API Connector](#)



Change this anytime from Plugins > Add new plugin

[Next step](#)

[Skip application assistant](#)

- Step 5/5 is to choose whether to install more advanced plug-ins for the app.
- You can add any other plug-ins you may need later, so we will not install any here.
- Finally, click "Get started building".

Do you want to add advanced features?

Step 5/5

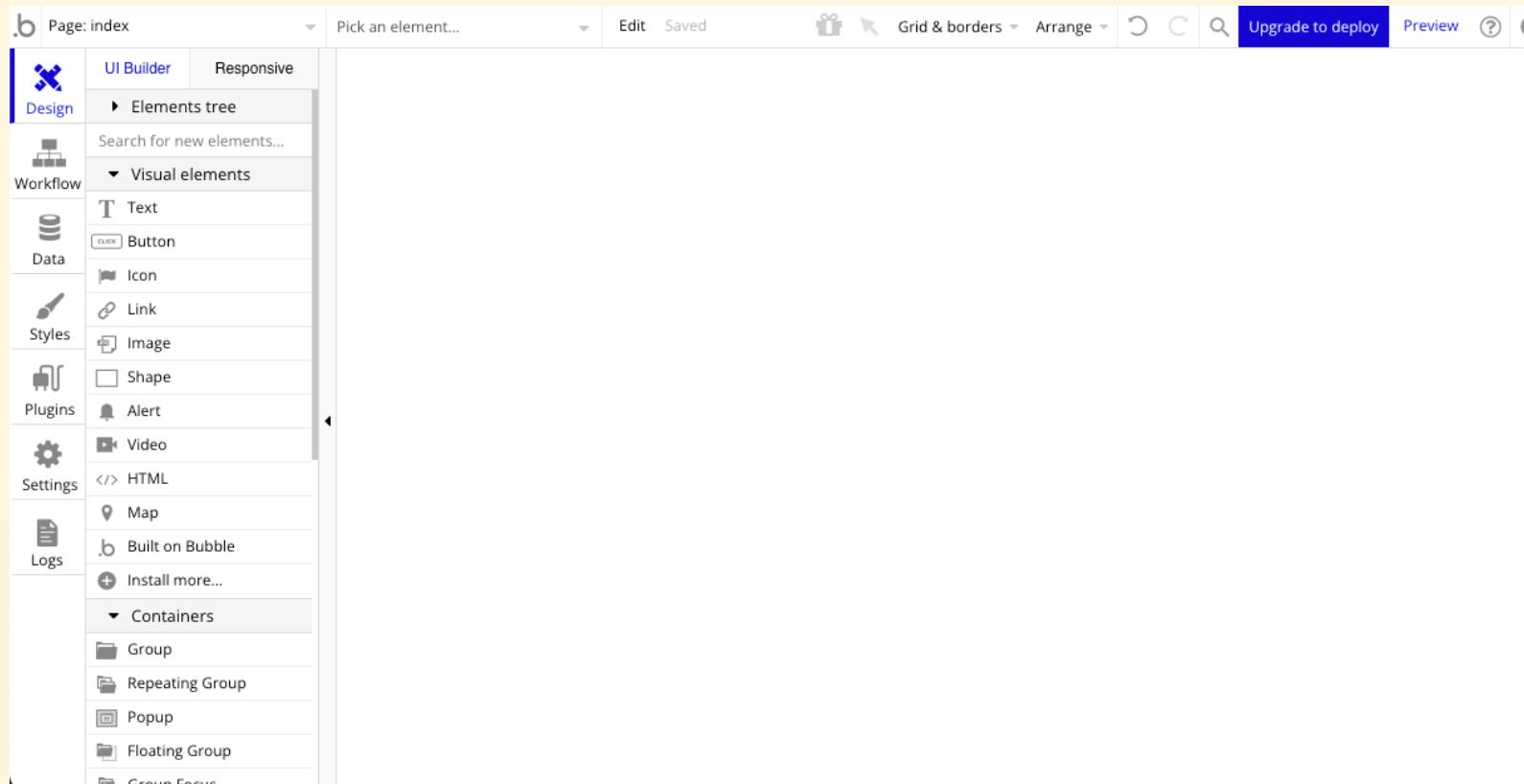
Install plugins for advanced functionality like maps, graphs, or multi-select dropdowns.

Install plugins

 Change this anytime from Plugins > Add new plugin

Get started building

- If a screen like this appears, it is OK.



まずは Bubble のテンプレートを動かしてみよう

- Click the Preview button in the upper right corner of the screen



- The preview screen starts up.
- But, only a blank screen is displayed.
- Unlike Adalo, no template has been selected, so there are no objects placed on the screen.

- As expected, we want to see it in action a bit, so let's try signing up and logging in to your account, just as we did with Adalo!
- Bubble does not provide a "screen" for signing up and logging in, but it does provide "components".
- I'd like to try to express the login screen by incorporating those parts.

First, let's create a user registration.

- We will touch various parts of the  screen, but we will explain each part later.
- Let's start with the minimum explanation of the parts and create a user registration screen.

Three main operations

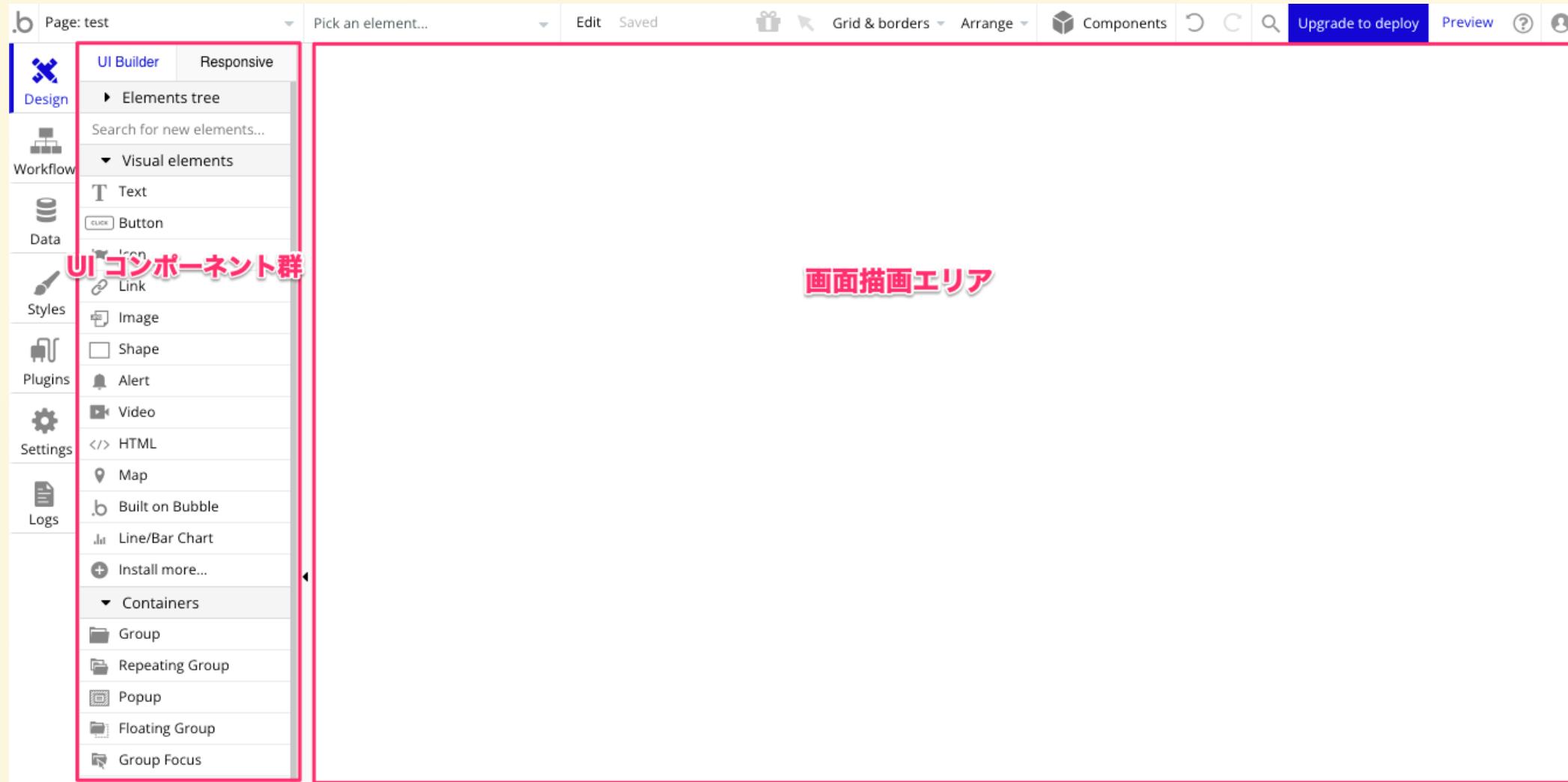
- Bubble uses three main operations
 1. design
 - Workflow
 - Data
- You can move back and forth between them from the left menu



Design

- Of these, only Design will be explained first.
- Design is the mode in which components are placed on the screen to create the user interface.
- The left panel contains the UI components and the right panel is the actual screen editing area.

- The taste of the screen is the same as Adalo's.



- Now let's create your user registration!

Sign up for My App

Email

Password

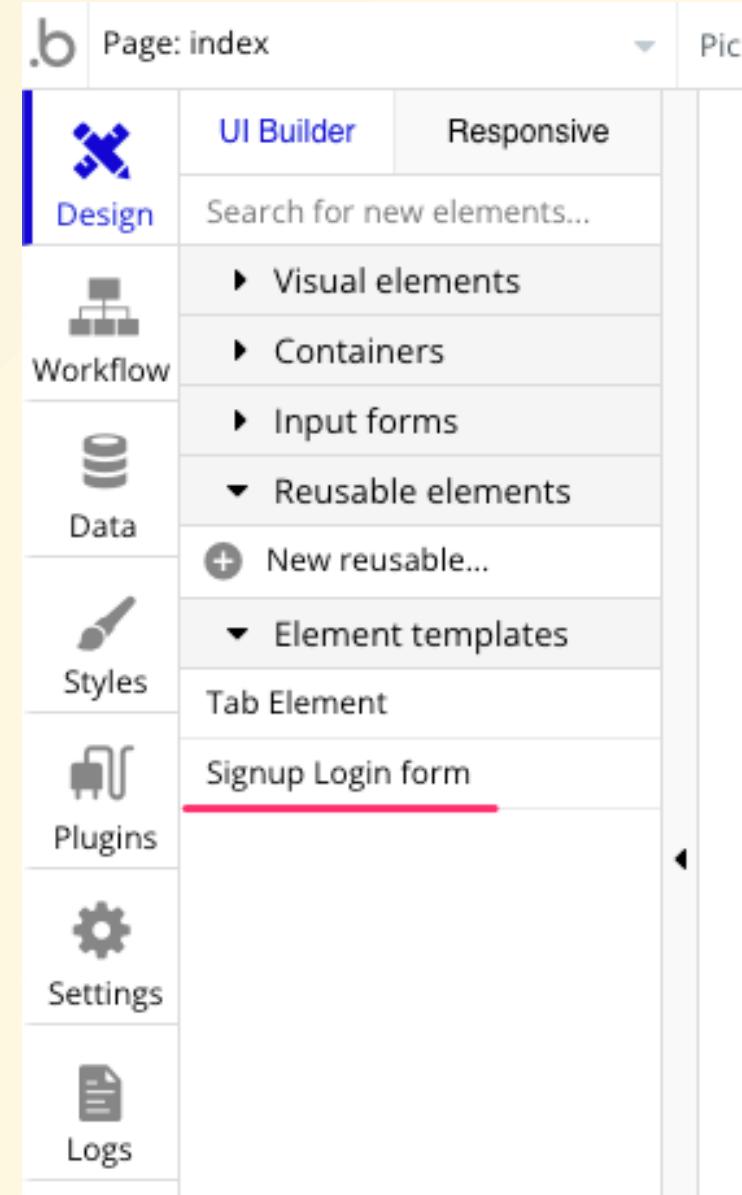
Confirmation

By signing up, I agree to the Privacy...

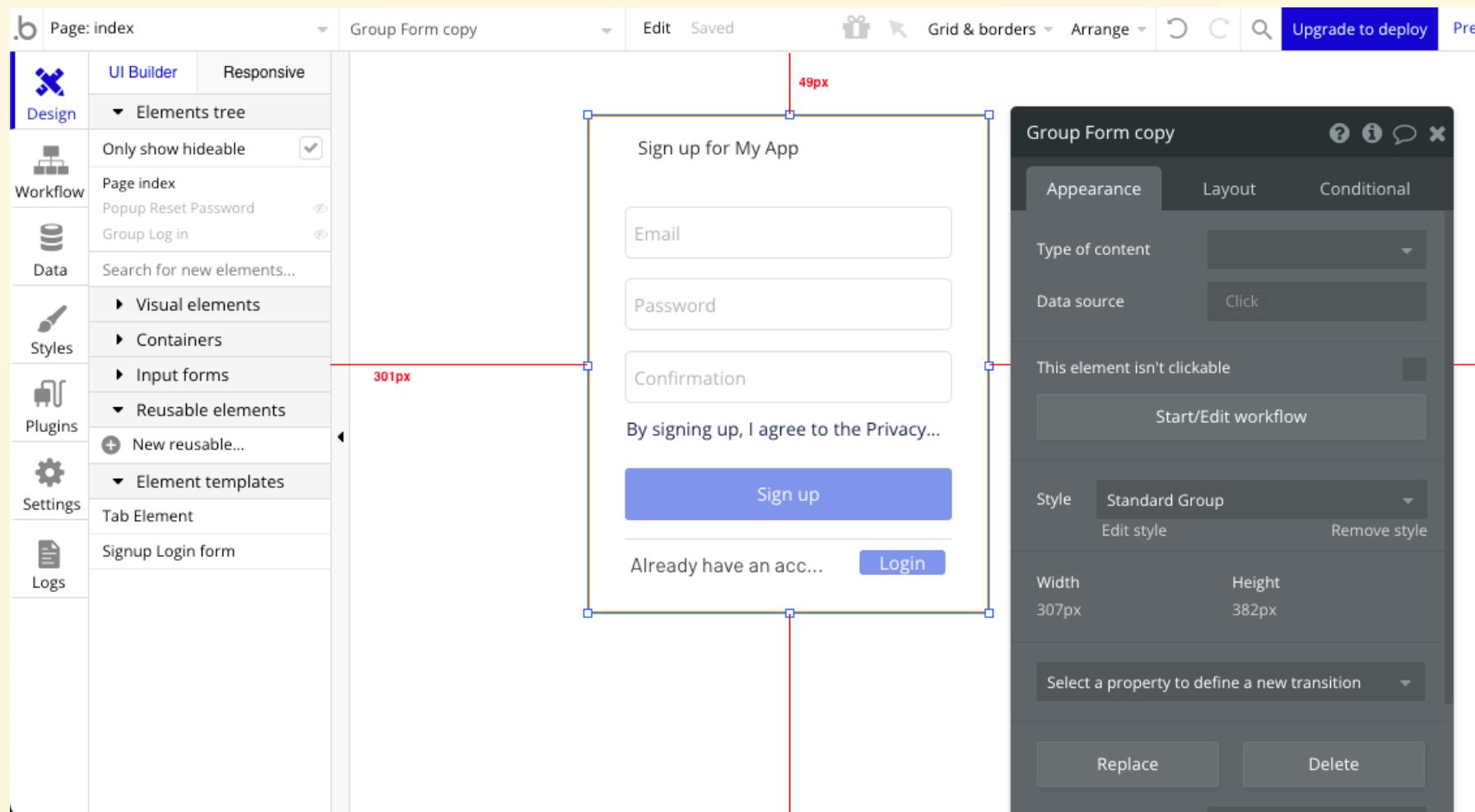
Sign up

Already have an account? [Login](#)

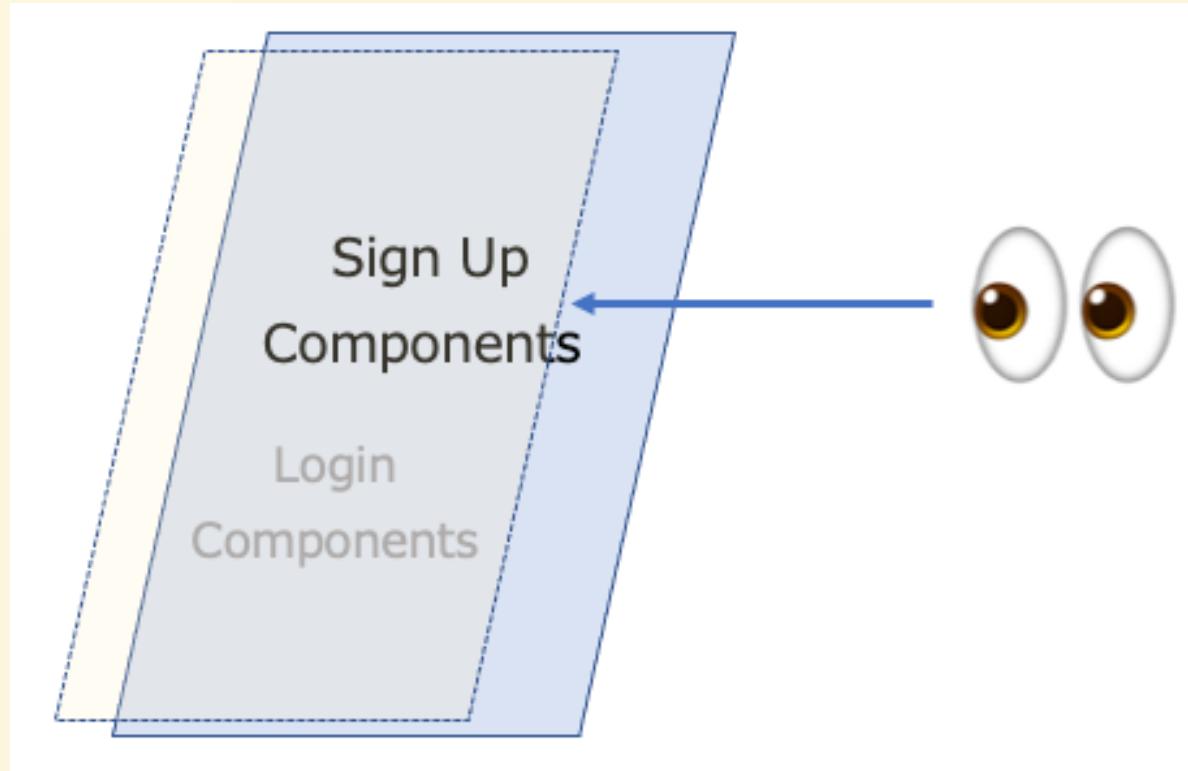
- First, let's place the signup and login components that come standard with Bubble.
- Click on "Signup Login form" in the Element templates in the UI Builder on the left panel.



- Click at the appropriate location in the right panel to display the user registration form



- ⚠️ This part is the "sign up" and "login" part together, so be careful when moving it around, etc.
- The way it works is that there is a "login" part on top of the "sign up" part that you can see now, but it is hidden.



- Let's run the preview here.
- You will then see a sign-up screen. Enter your email address and password to sign up.
 - Please make a note of the Email and Password you entered (you will use them later).

Sign up for My App

test@example.com

.....

.....

By signing up, I agree to the Privacy...

Sign up

Already have an account? [Login](#)

- Probably nothing will happen, but we're good to go for now.
- Now we can use the sign up and login, so let's move on 

- Close the preview and select the Data tab, then select the "App data" tab from the right panel and click on "All Users"
- Then you should see the data of the list of registered users in the right panel, and the user you just "signed up" should be there.
- Now you have confirmed that the screen and the database are linked.

The screenshot shows the Bubble app interface with the following details:

- Header:** Page: pet_weight_register, Edit, Help, Saved, 8 issues, Development, Preview, User icon.
- Left Sidebar:** Design, Workflow, **Data** (selected), Styles, Plugins, Settings.
- Top Navigation:** Data types, Privacy, App data (selected), Option sets, File manager.
- Main Area:**
 - Database views:** Application data - All Users - Development version, Copy and restore database, Switch to live database.
 - Actions:** New view, Primary fields, Search, 1 entries (displaying 1), New entry, Delete (0), Upload, Modify, Export, Bulk.
 - Table:** Shows a single entry for a user.

		Email	Created Date	Modified Date	Slug
<input type="checkbox"/>	<input type="checkbox"/>	kyogoku+bubble_test@guildwo	Oct 31, 2021 8:51 pm	Oct 31, 2021 8:51 pm	
 - Links:** All Pets, All PetWeightLogs, **All Users** (selected).

About Smartphone Support

- As we mentioned at the beginning of this document, Bubble's application is based on the premise of a web application.
- When adapting Bubble's application for smartphones, we will adopt the concept of **responsive design**.

- Let's see how it works first
- Try shrinking the width of your browser

<https://matsushitahome.com/>

- The layout is not broken, and the elements are automatically rearranged to be suitable for the smartphone size!
- This is what responsive design is all about!
- Google explains it like this

Regardless of the user's device type (PC, tablet, mobile, non-visual browser)
Deliver the same HTML code with the same URL, but change the rendering method based on screen size.

参考資料

<https://developers.google.com/search/mobile-sites/mobile-seo/?hl=ja>

- The key points are as follows
 1. the URL of the page being displayed is the same
 2. the rendering method is automatically changed according to the screen size

- To put it a little more simply, it controls whether the element will stretch/shrink, wrap/not wrap, and display/not display according to the screen size.
- By setting these at the item level, you can automatically make the design responsive.
 - When actually developing, it is a good idea to preview the settings and check the operation.
- (The story of making Bubble applications responsive will be explained in the second Bubble lecture.)

Overview of Application Development with Bubble

Next, we will give an overview of application development with Bubble

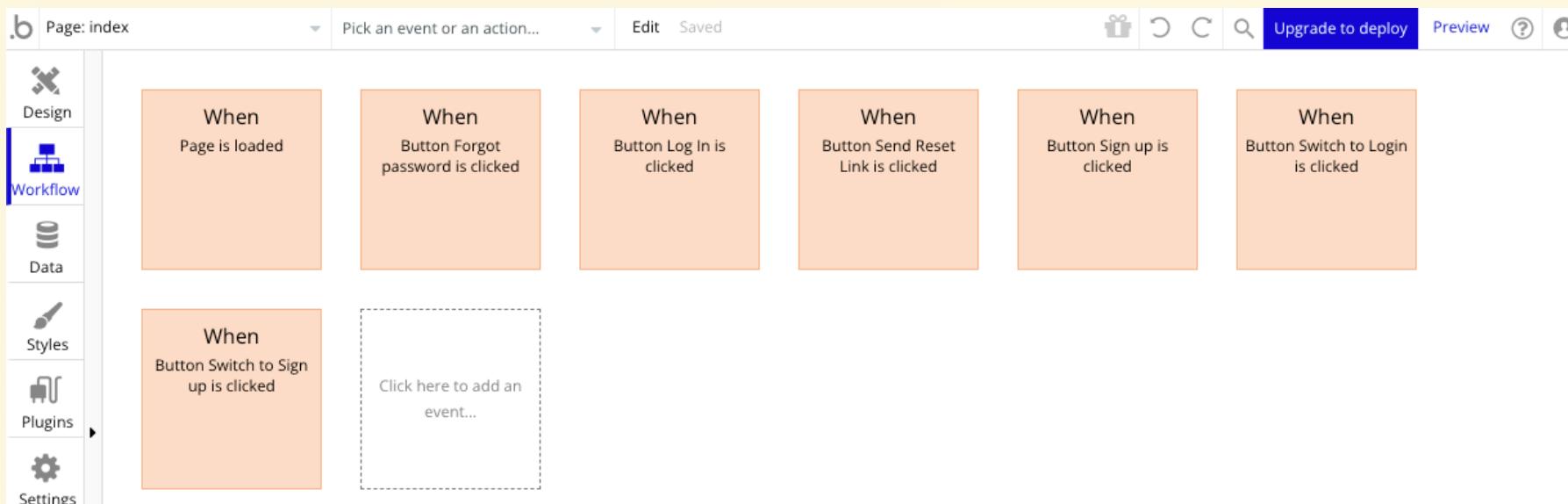
Three main operations

- We will look at the remaining two operations
1. design
 - Workflow
 - Data

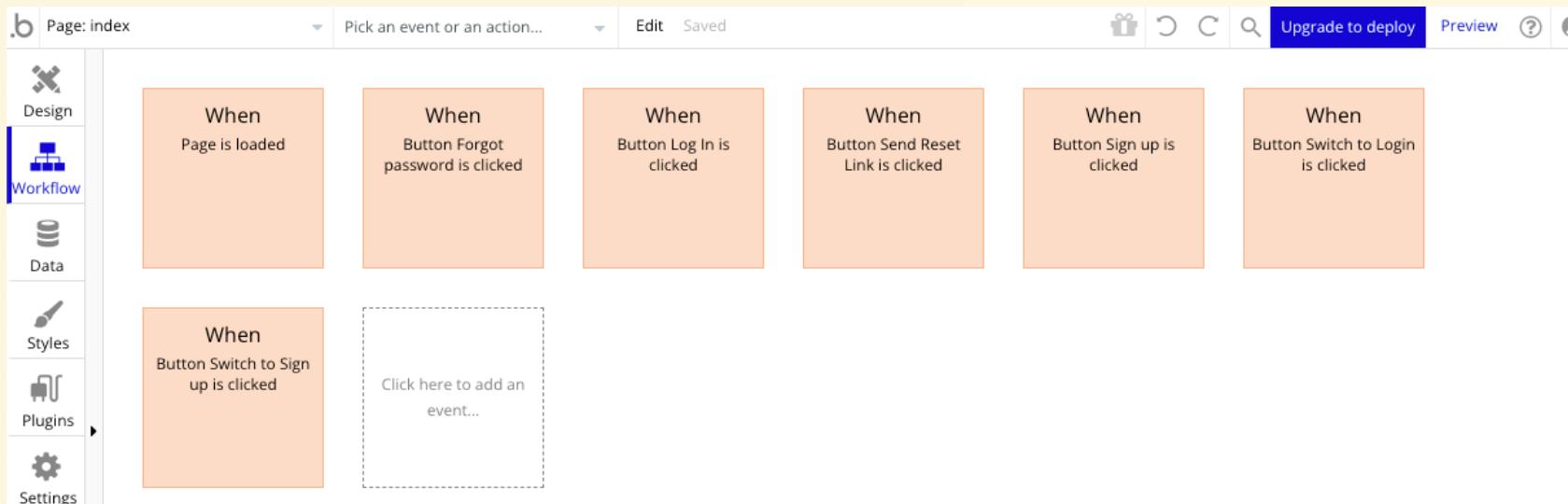


Workflow

- Mode for adding movement to the screen
 - Example: Screen transitions occur when a button is pressed.
 - Example: Data is manipulated when a button is pressed.
- This is the part that requires a bit of programming thinking.



- By the way, I think you already have a few boxes
- This is the standard workflow definition provided by Bubble.
 - This is the standard workflow definition provided by Bubble, which also includes the sign-up and login workflows we have just included.
 - We'll take a closer look at the contents later.



Data

- Mode of defining and manipulating data

The screenshot shows a software interface for managing data types. On the left, a vertical sidebar lists navigation options: Design, Workflow, Data (which is selected and highlighted in blue), Styles, Plugins, Settings, and Help. The main area has a header with tabs for 'Page: index', 'Edit', and 'Saved', along with icons for gift, copy, trash, search, and upgrade.

The central part of the screen displays a 'Data types' section. A sub-section for 'Custom data types' is shown, specifically for the 'User' type. The 'User' type is currently selected, indicated by a blue border around its name. To the right, a list of fields for the 'User' type is displayed, each with a type name, field type, and status as a 'Built-in field'. The fields listed are:

Type name	Field Type	Status
email	text	Built-in field
Modified Date	date	Built-in field
Created Date	date	Built-in field
Slug	text	Built-in field

Below the field list is a button labeled 'Create a new field'. On the left side of the main area, there is a form for creating a new data type. It includes fields for 'New type' (with a text input box), a checkbox for 'Make this data type private by default' (unchecked), and a note stating 'Things will be visible to everyone'. At the bottom of this form is a 'Create' button.

- Select the "App data" tab and click on the "All Users" link
- You should now see a list of registered users in the right panel, and the user you just signed up should be there.
- You have now confirmed that the screen and database are linked.

The screenshot shows the Backstage application interface. On the left is a vertical sidebar with icons for Design, Workflow, Data (which is selected), Styles, and Plugins. The main area has a header with 'Page: index' and buttons for Edit, Saved, and various deployment options like Upgrade to deploy, Preview, and a help icon. Below the header, there are tabs for Data types, Privacy, App data (which is active and highlighted in red), Option sets, and File manager. Under the App data tab, there are sections for Database views, Application data - All Users - Development version, and buttons for New view, Primary fields, Search, New entry, Delete (0), Upload, Modify, Export, and Bulk. A note indicates '2 additional fields'. Below these are buttons for Copy and restore database and Switch to live database. At the bottom, there is a search bar for view names or data types, and a table showing user data with columns for Email, Created Date, Modified Date, and Slug. The row for the user 'test@example.com' is highlighted with a red border. The 'All Users' link at the bottom of the page is also highlighted with a red underline.

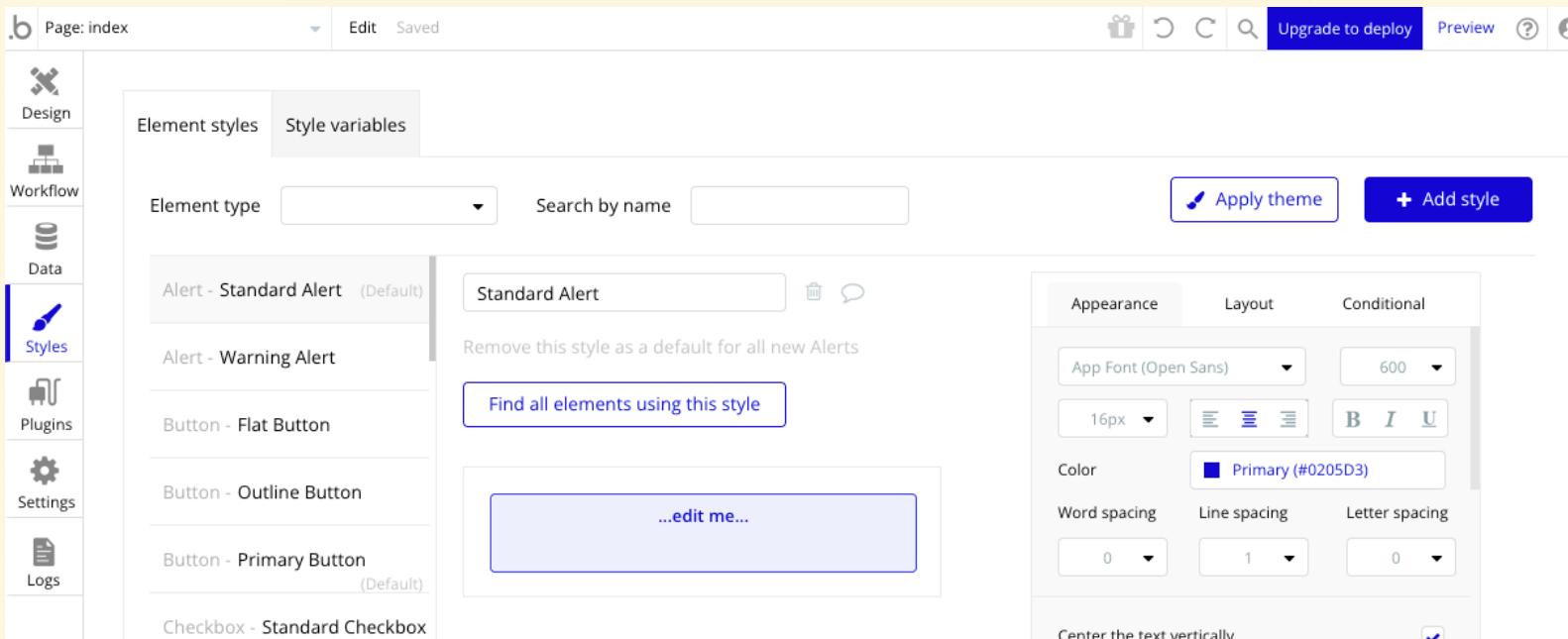
		Email	Created Date	Modified Date	Slug
<input type="checkbox"/>	<input type="checkbox"/>	test@example.com	Nov 6, 2022 5:02 pm	Nov 6, 2022 5:02 pm	

Other Operations

Other operations (menus) are also briefly explained

Styles

- By naming styles, they can be used universally throughout the application (or individually for each part)



Plugins

- Just as Adalo has added a line chart component, Bubble has a variety of components available to extend the application
- In Bubble, we call them "plug-ins".
- Bubble plug-ins can be free or paid for, so be sure to check before you use them!

Settings

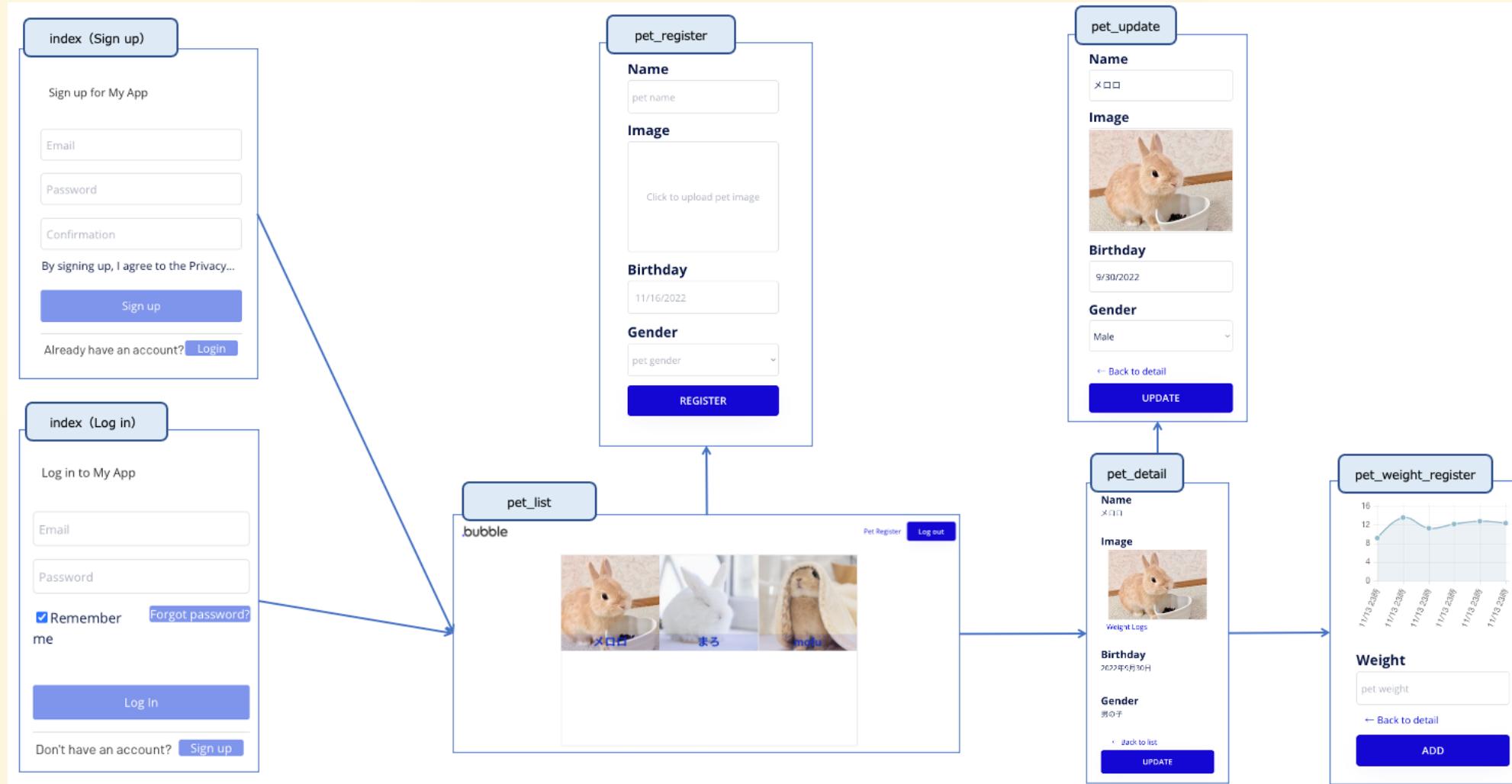
- Bubble plan changes and account operations
- Not covered in this lecture.

- Languages in Settings allows you to centrally manage the messages provided by Bubble
 - Change the locale (language setting) for the main user of the application
 - By changing these messages, you can unify message management throughout the system.
 - You can change the messages that are commonly used throughout the application, or change the default messages.
 - You can also add application-specific messages.
- Not covered in this lecture.

Logs

- You can view the log of the application when it is running.
- You can also view the log while previewing the application
- Not covered in this lecture.

これから作っていく画面の全体像

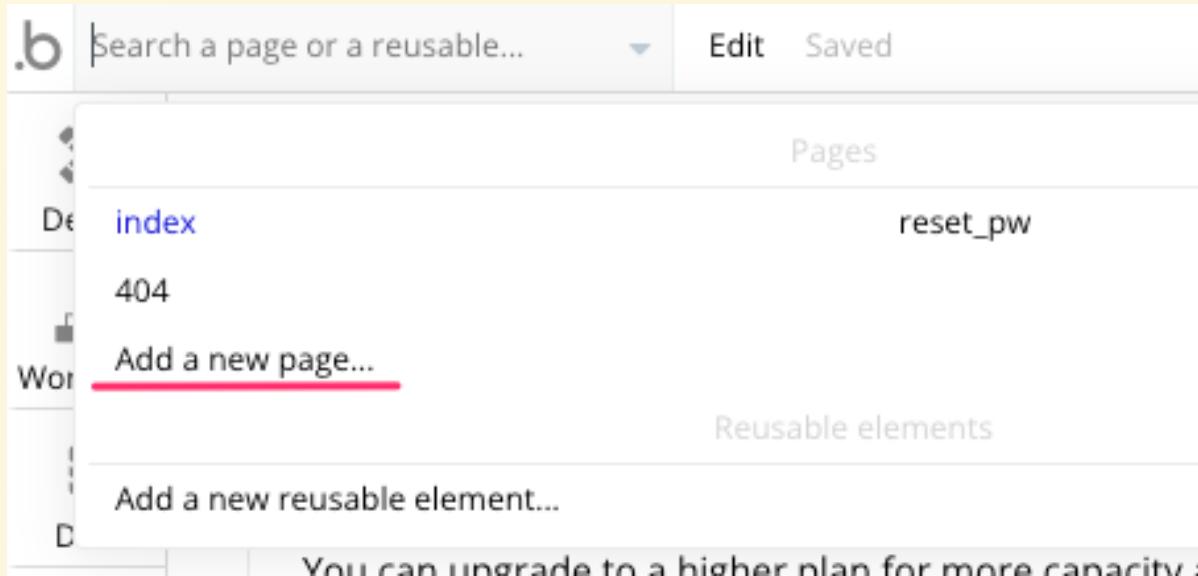


ペット登録から一覧表示まで作ってみよう

- We will create the same screen elements and data structures that we created in the meeting at Adalo
- Therefore, we will create the screen and database operations together at the Bubble meeting ! !
- First, let's create a screen for registering pets, and then let's create a list of those pets ! !

We will create a pet registration screen.

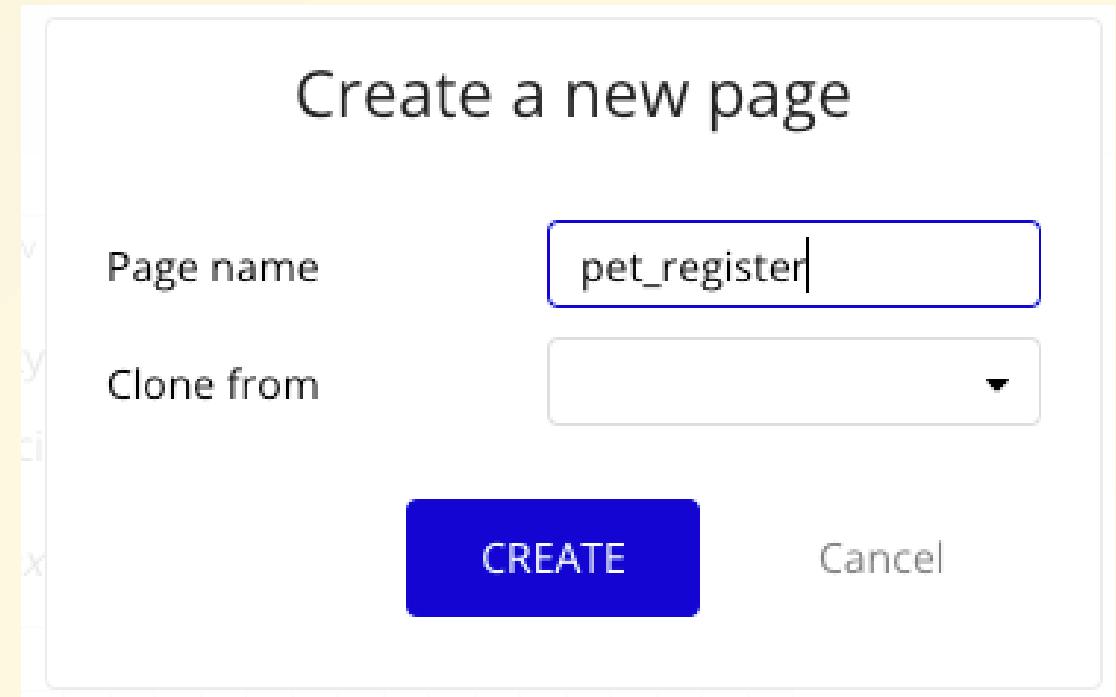
- Click **Page: index** in the upper left corner of the screen and a popup will appear.
- Click **Add a new page...** in the pop-up window.



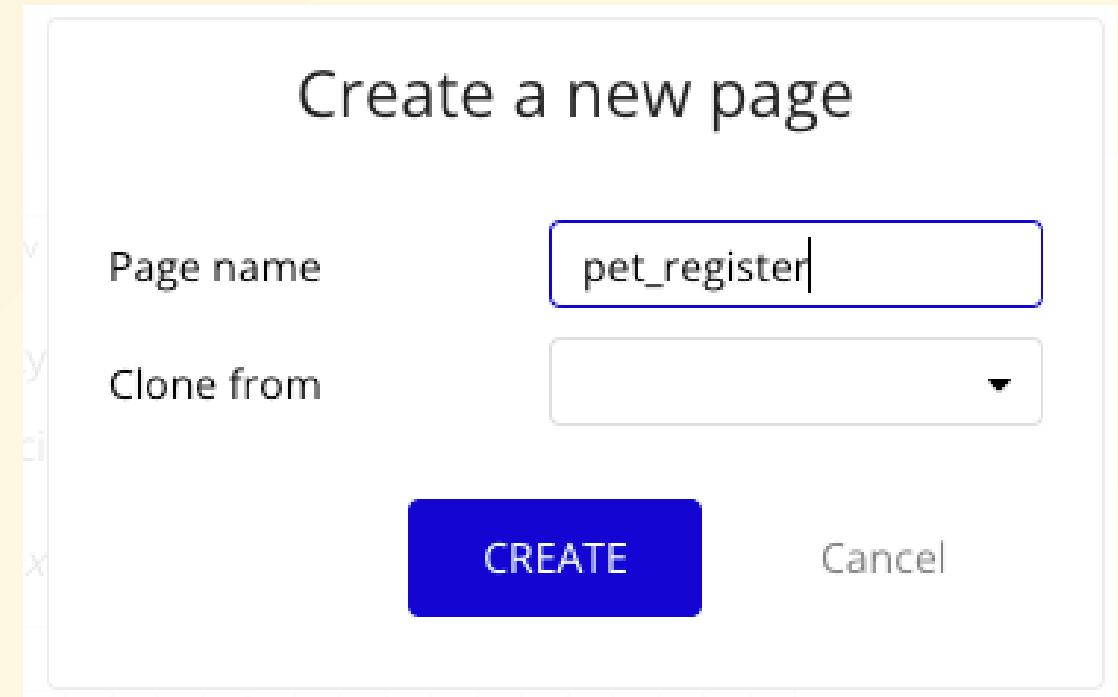
By the way...

- Here you can see the list of "pages" and "common components" in the application you are currently creating
- In Adalo, all the screens were displayed on one canvas, but in Bubble, you can only manipulate one screen on one canvas.
- Therefore, it is important to remember that you need to change the screen to move from one screen to another from here 

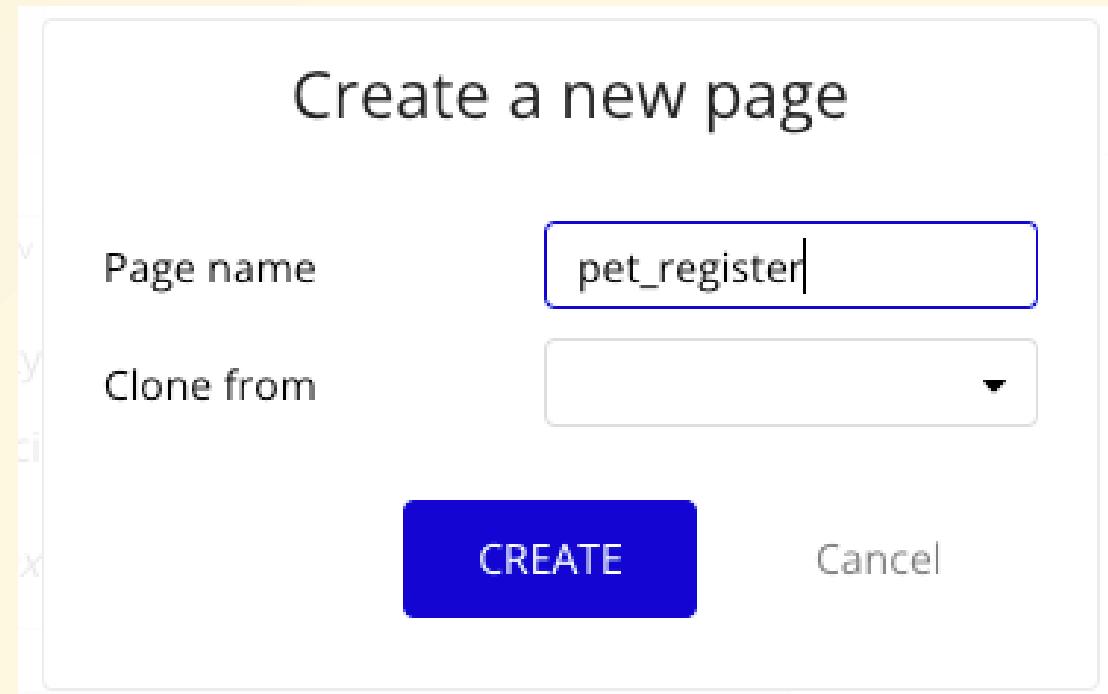
- Add a new page... to pop up a new page...
- The **Page name** is the name of the page, so enter only alphanumeric characters without spaces.
 - In this case, we will use **pet_register**.



- **Clone from** allows you to create a copy of a screen by selecting a screen from which you want to create a similar screen.



- For example, since the elements of the registration screen and the edit screen are almost the same, you can reduce development man-hours by selecting the registration screen as "Clone from" when creating the edit screen after the registration screen has been created.



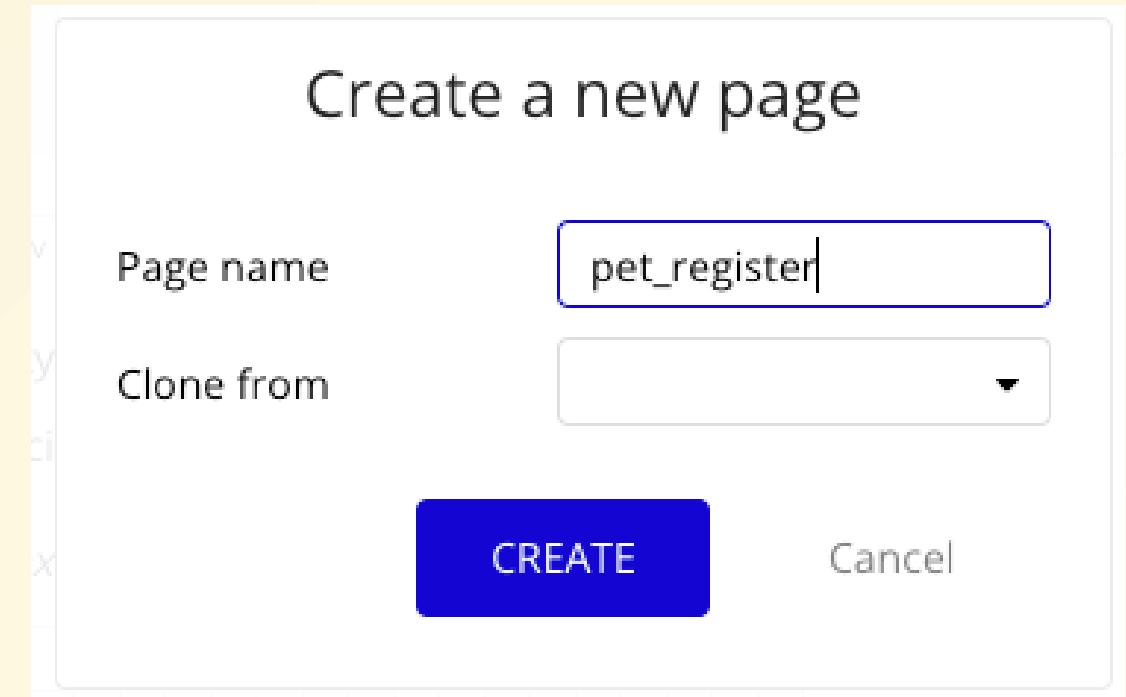
- Leave unselected for this time
- Click on the "CREATE" button

Create a new page

Page name

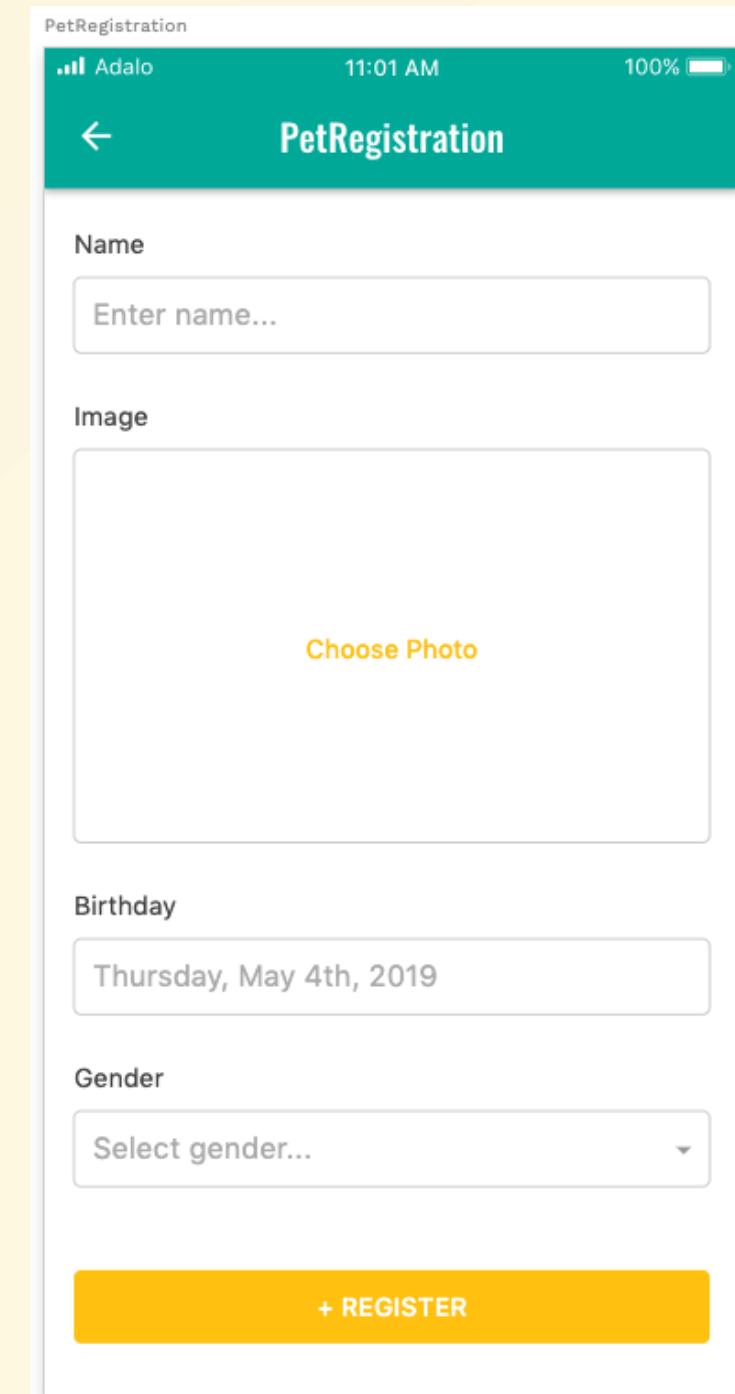
Clone from

CREATE Cancel



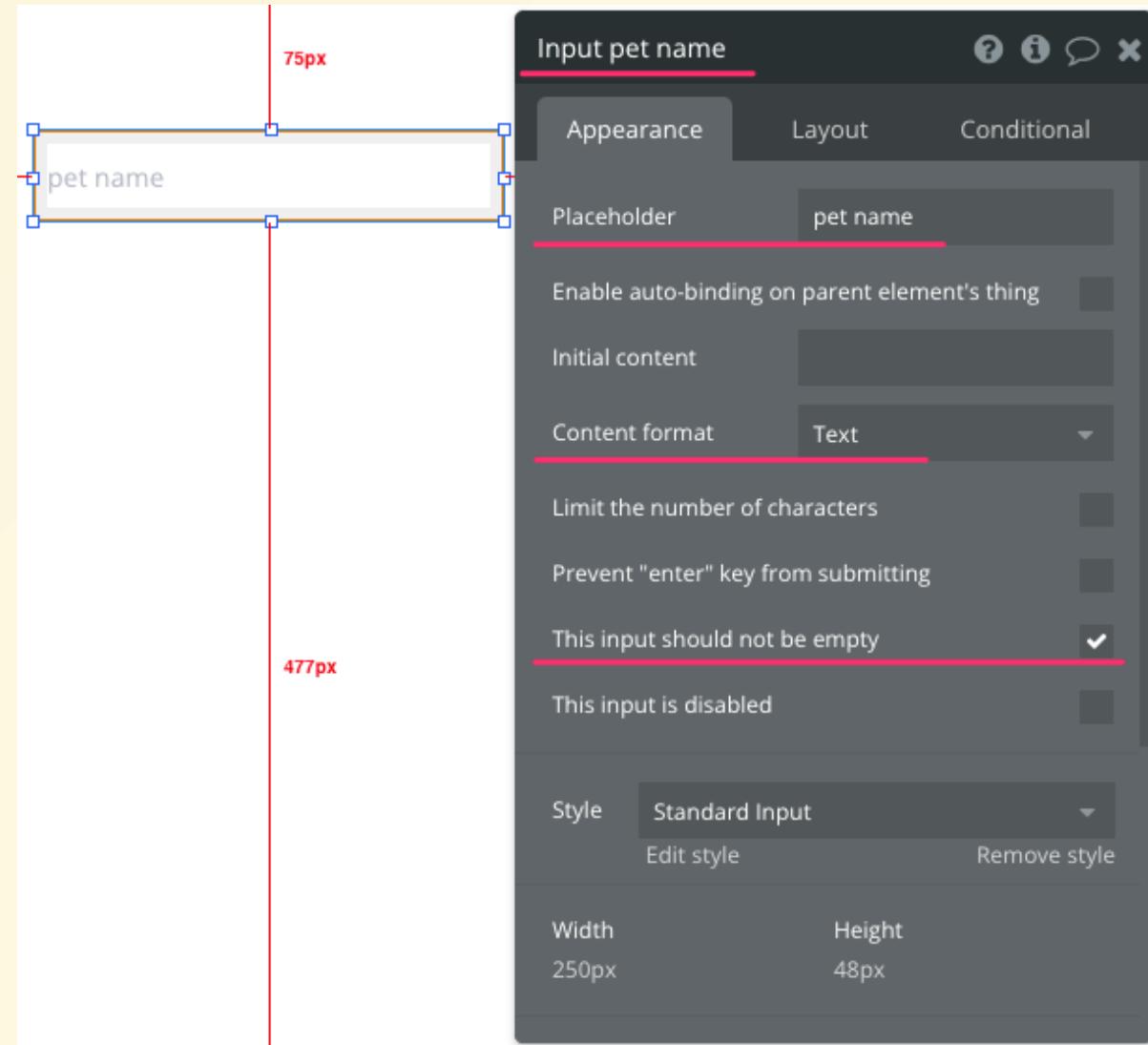
Assembling the Pet Registration Screen

- Let's build a screen based on the image of the pet registration screen created by Adalo.

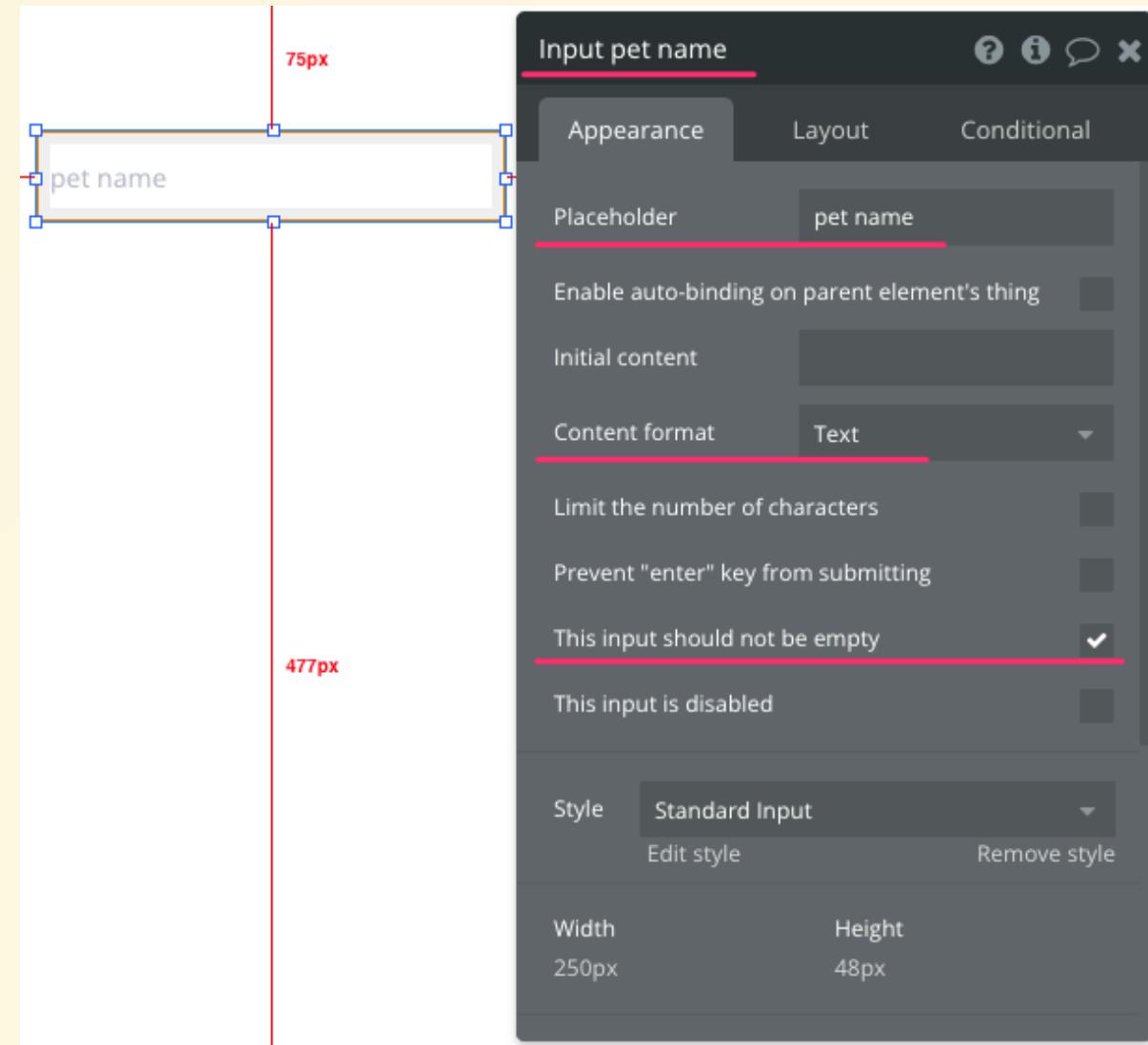


Pet Name

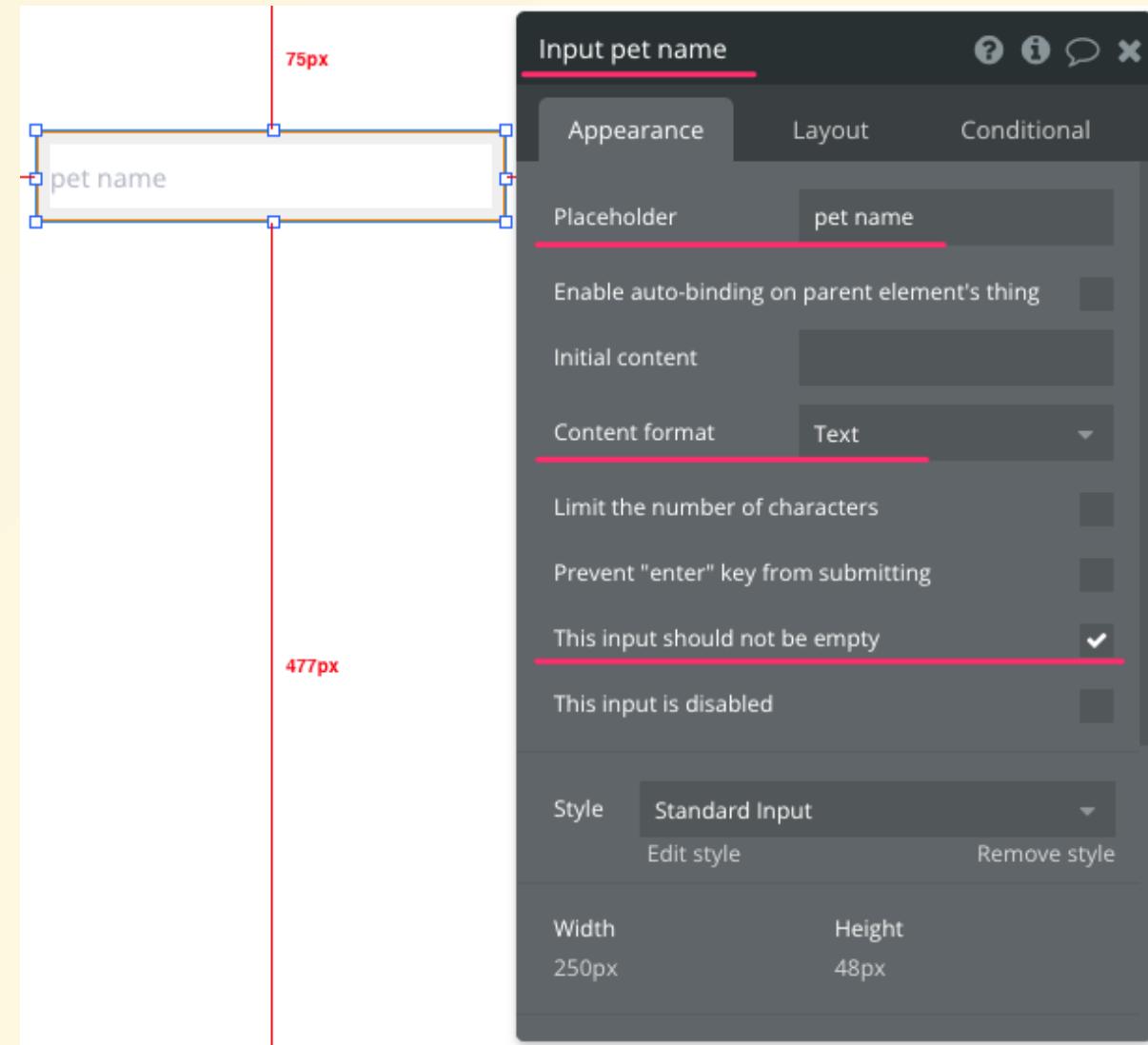
- First, place a text box for the pet's name
- Select **Input** in **Input forms** in the "UI Builder" in the left panel.
- Then, click on the place where you want to place the text box in the right panel, and the input box will appear.



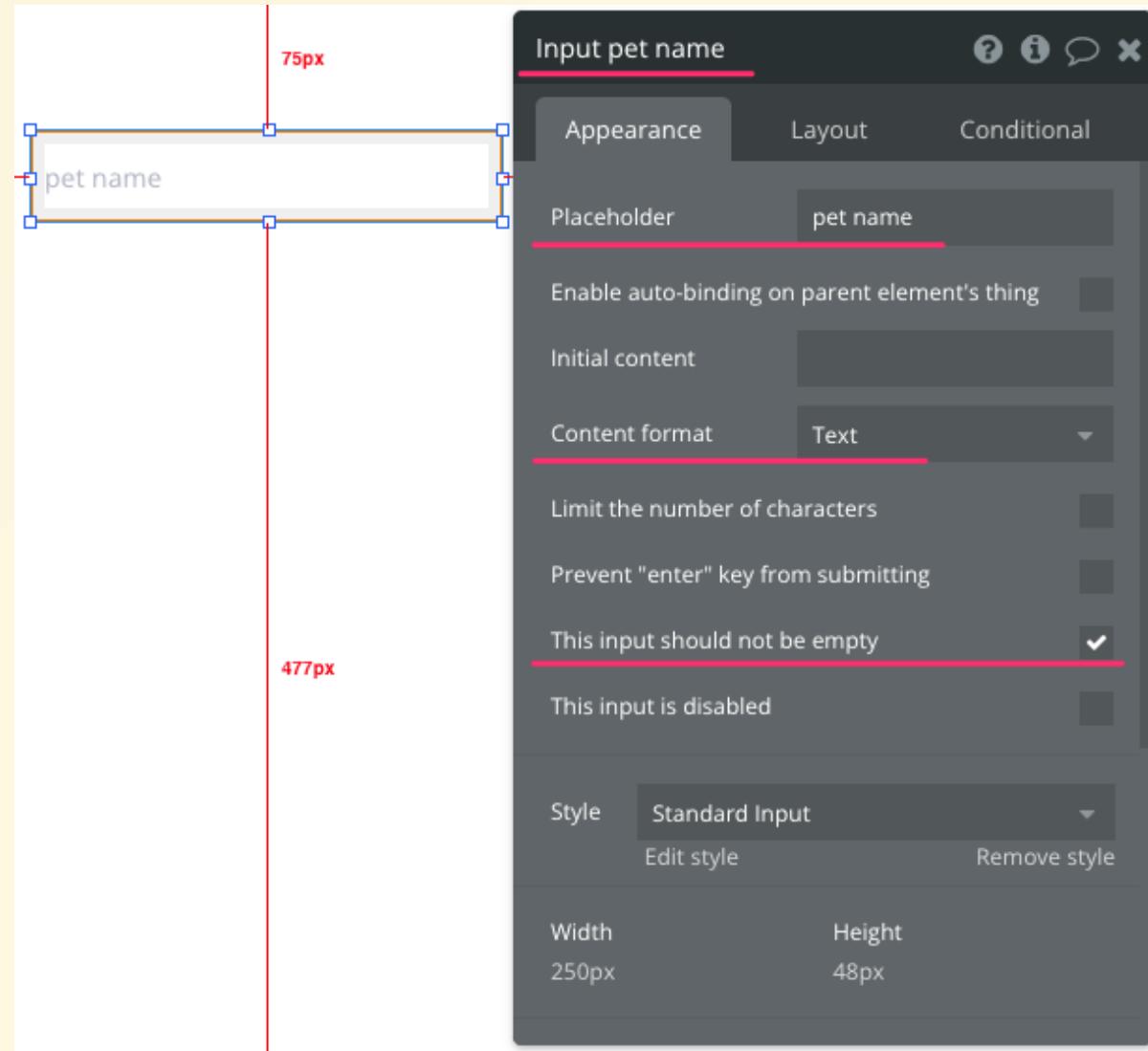
- Double-click on an element to open a dialog box where you can set various information about the element.
- Placeholder` is an auxiliary text that will be displayed when the text box is not filled in.
 - In this case, we will use "pet name".



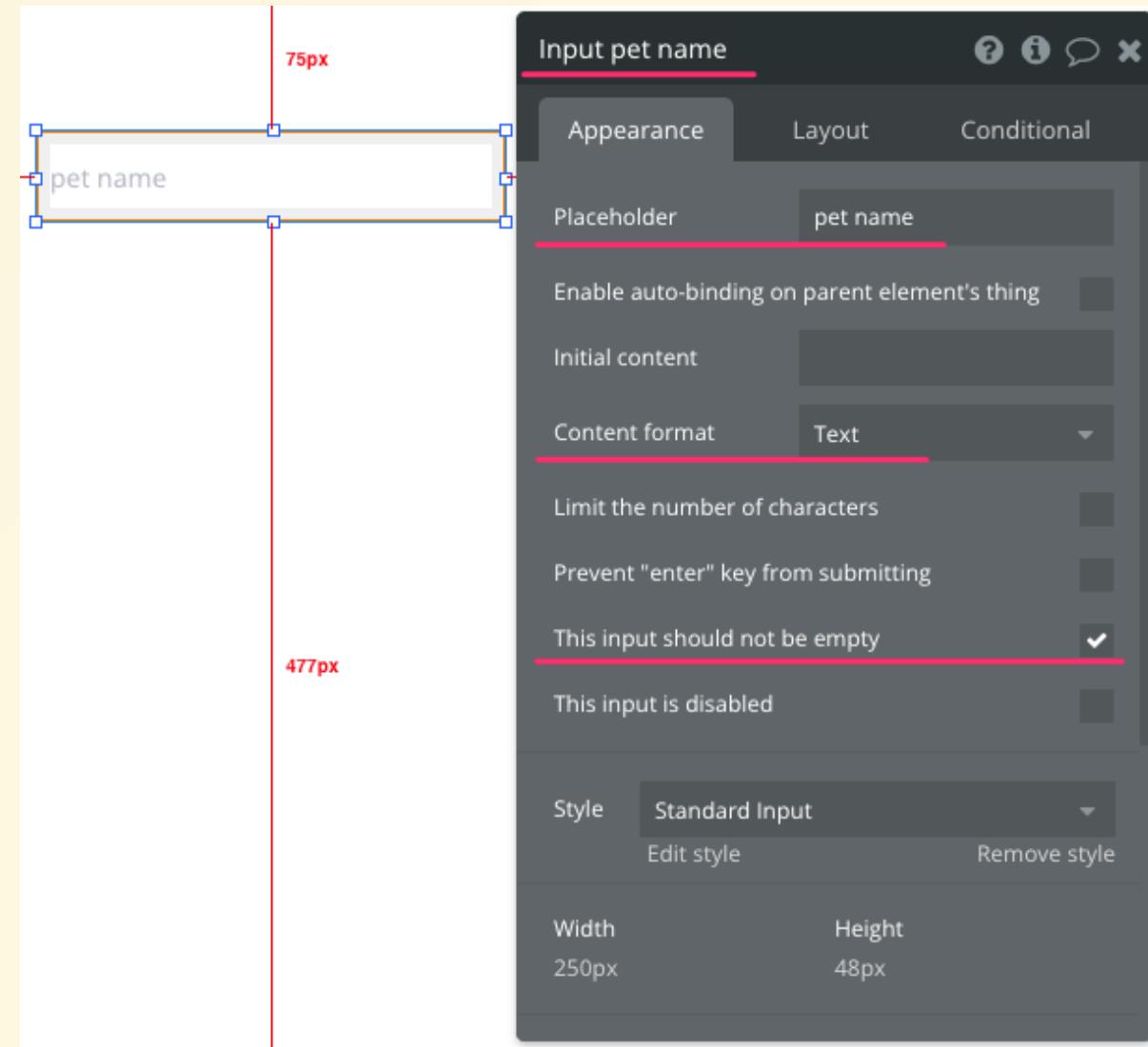
- The Content format allows you to specify the format of the values that can be entered into the textbox
 - In this case, we will leave it as "Text" since we will be entering a string.



- The **This input should not be empty** option allows you to specify whether or not the input is required.
 - In this case, we want to make it mandatory, so we check the box.



- If you click on the top of the dialog, you can name the elements in this text box
- This will come in handy later when defining the workflow, so let's specify "Input pet name" here.



- After setting up to this point, you are ready to go.
 - You can specify other details, but I won't explain them here.
 - If you are interested, please take a look 

Pet Image

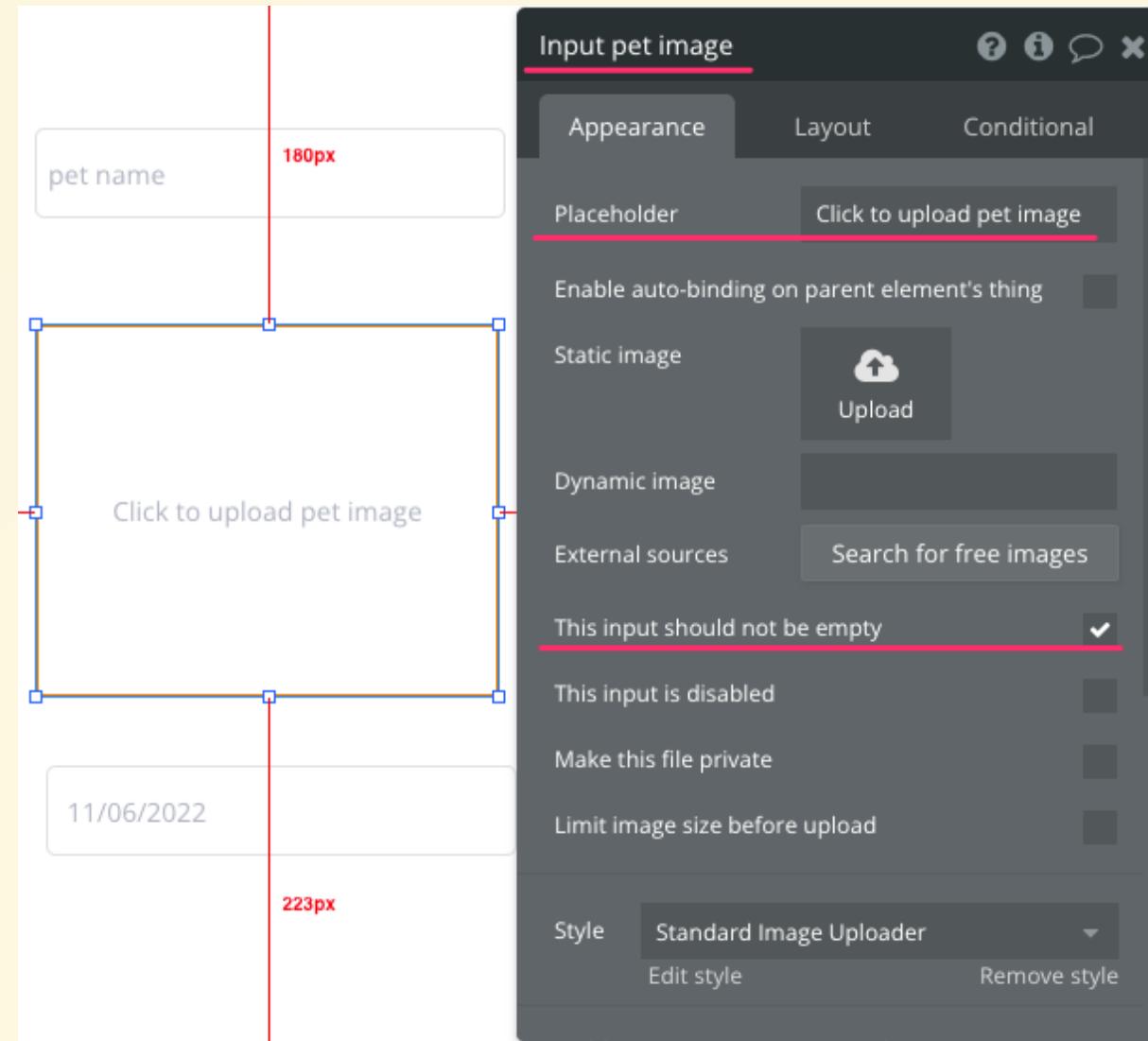
- Next, place the element that will upload the pet's image
- Select **Picture Uploader** from the left panel and **drag** the element under the pet's name on the right panel

The screenshot shows the UI Builder interface with the following components:

- Page: index**: The current page name.
- UI Builder**: A tab labeled "UI Builder".
- Responsive**: A tab labeled "Responsive".
- Design**: A section header with a blue icon. It contains a link to the **Elements tree** and a search bar for new elements.
- Workflow**: A section header with a grey icon. It contains links to **Visual elements** and **Containers**.
- Data**: A section header with a grey icon. It contains a collapsed section for **Input forms**, which includes **Input** and **Multiline Input**. It also includes **Checkbox**, **Dropdown**, **Search Box**, **Radio Buttons**, **Slider Input**, **Date/Time Picker**, and **Picture Uploader**.
- Styles**: A section header with a grey icon.
- Plugins**: A section header with a grey icon.
- Settings**: A section header with a grey icon.
- Logs**: A section header with a grey icon.

The **Picture Uploader** item is highlighted with a red underline, indicating it is selected or the current focus of the tutorial.

- Double-click on this element as well to set the element information.
 - In this case, enter "Click to upload pet image" in the Placeholder field.
 - Specify "Input pet image" as the element name.
 - Check This input should not be empty to make it mandatory.



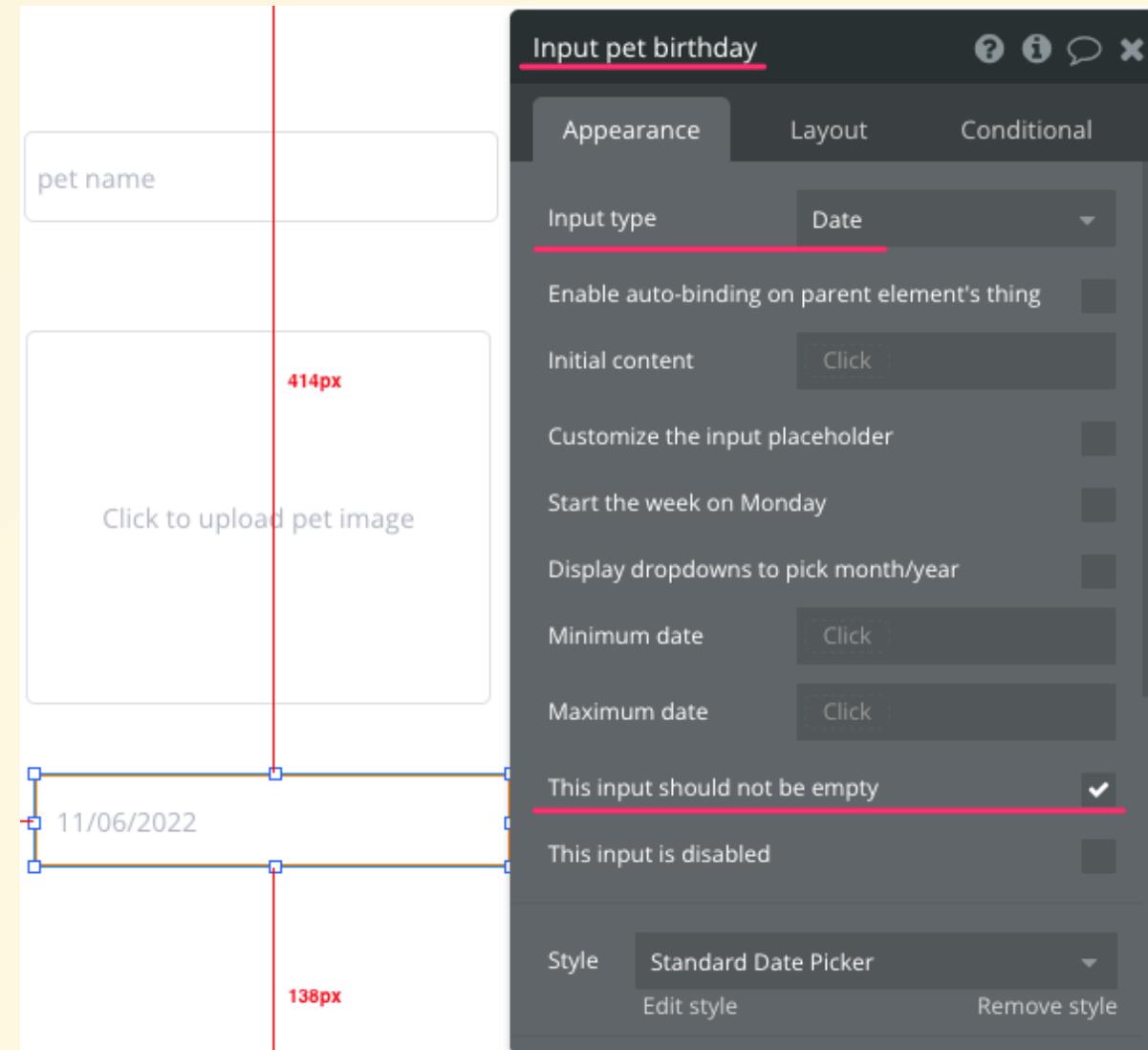
Pet's Birthday

- Next we will place the elements for entering the pet's birthday
- Select the **Date/Time Picker** from the left panel and drag the element on the right panel to place it

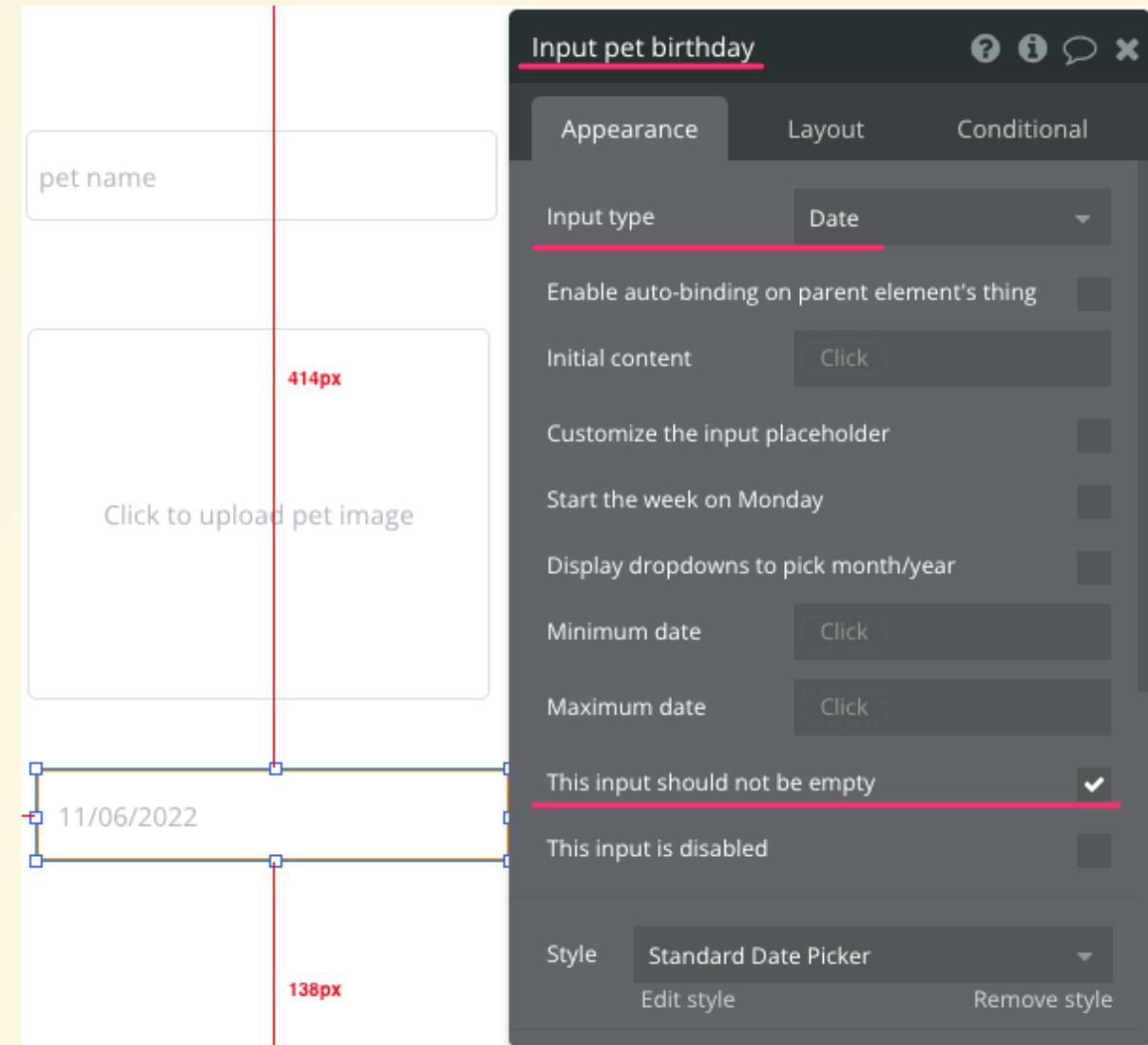
The screenshot shows the UI Builder interface with the following details:

- Page:** pet_register
- UI Builder:** Responsive
- Design:** Elements tree
- Workflow:** Search for new elements...
- Data:** Visual elements, Containers, Input forms (expanded), Input, Multiline Input, Checkbox, Dropdown, Search Box, Radio Buttons, Slider Input, Date/Time Picker (highlighted with a red underline), Picture Uploader
- Styles:** (Icon)
- Plugins:** (Icon)
- Settings:** (Icon)
- Logs:** (Icon)

- Double-click on the element to set the element details
 - Select **Date** for the **Input type** to allow only date input.
 - Select **Date & Time** to input date and time.
 - Select **Date & Time** here to allow date and time input.

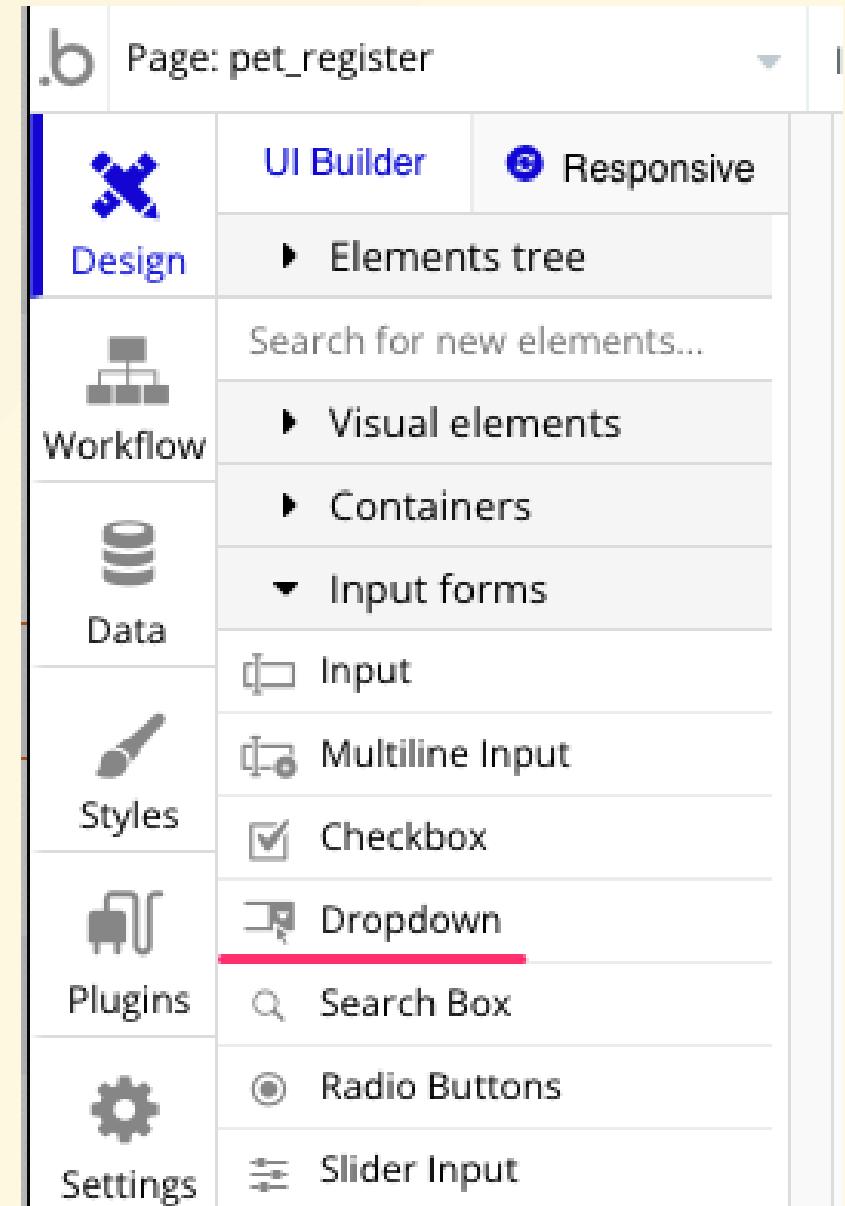


- Check This input should not be empty to make it mandatory.
- Specify Input pet birthday as the name of the element.

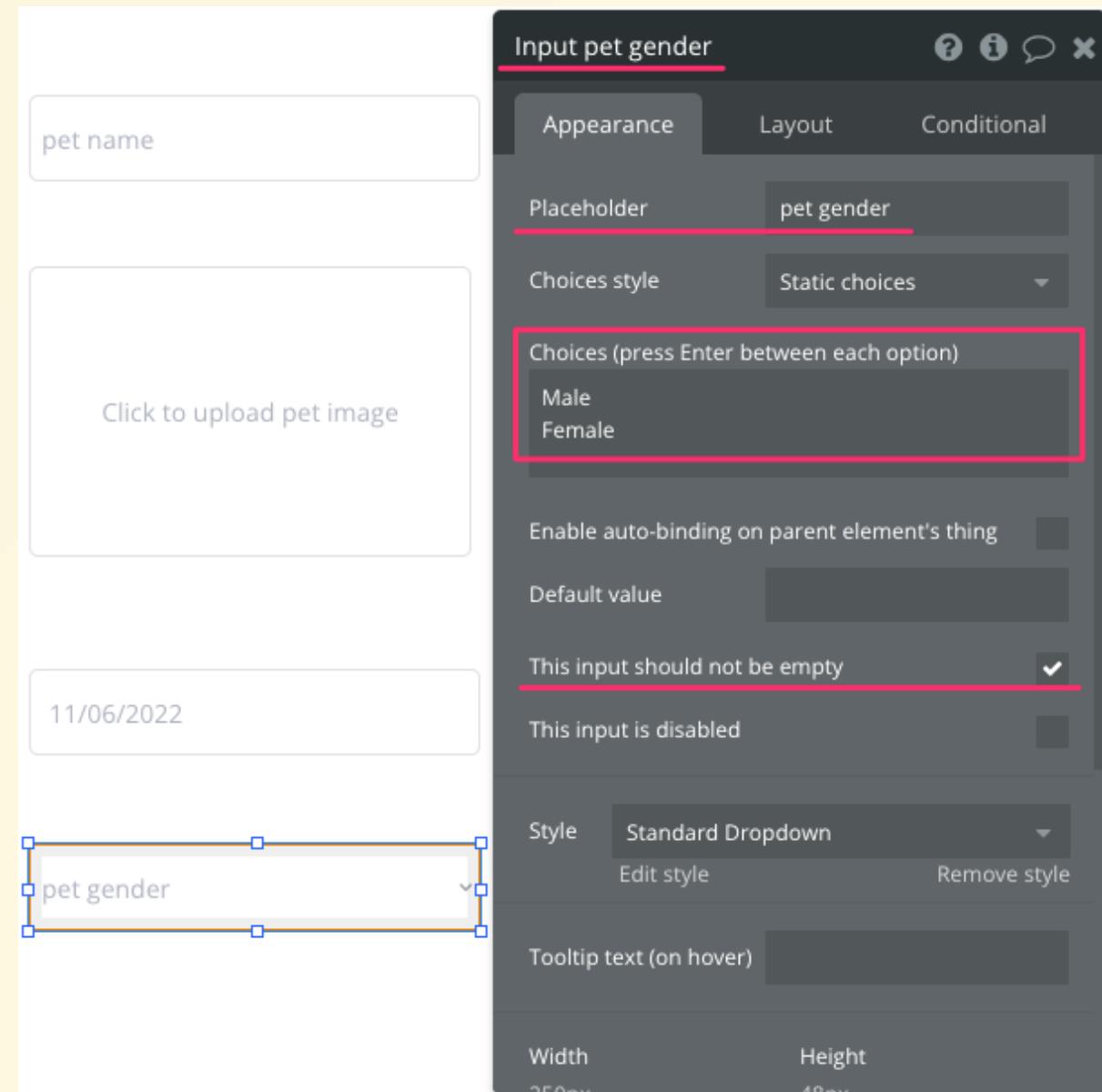


Pet Gender

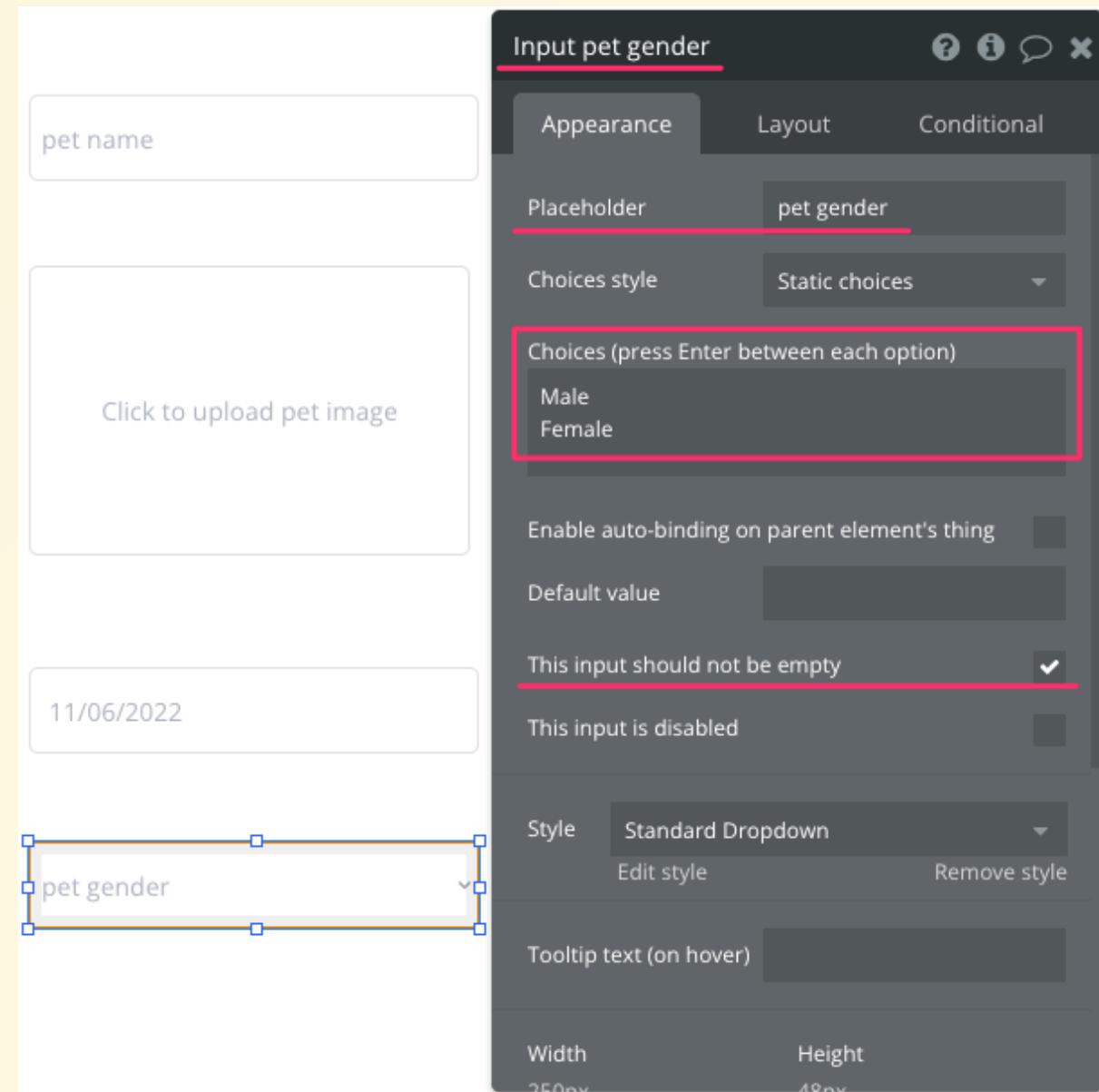
- Finally, we place an element to select the gender of the pet
- Select **Dropdown** from the left panel and click on the right panel to place the element



- Double-click on the element to set the element details
 - Type "pet gender" in the Placeholder field.
 - In the Choices field, type "Male", press Enter to start a new line, and type "Female" on the next line.

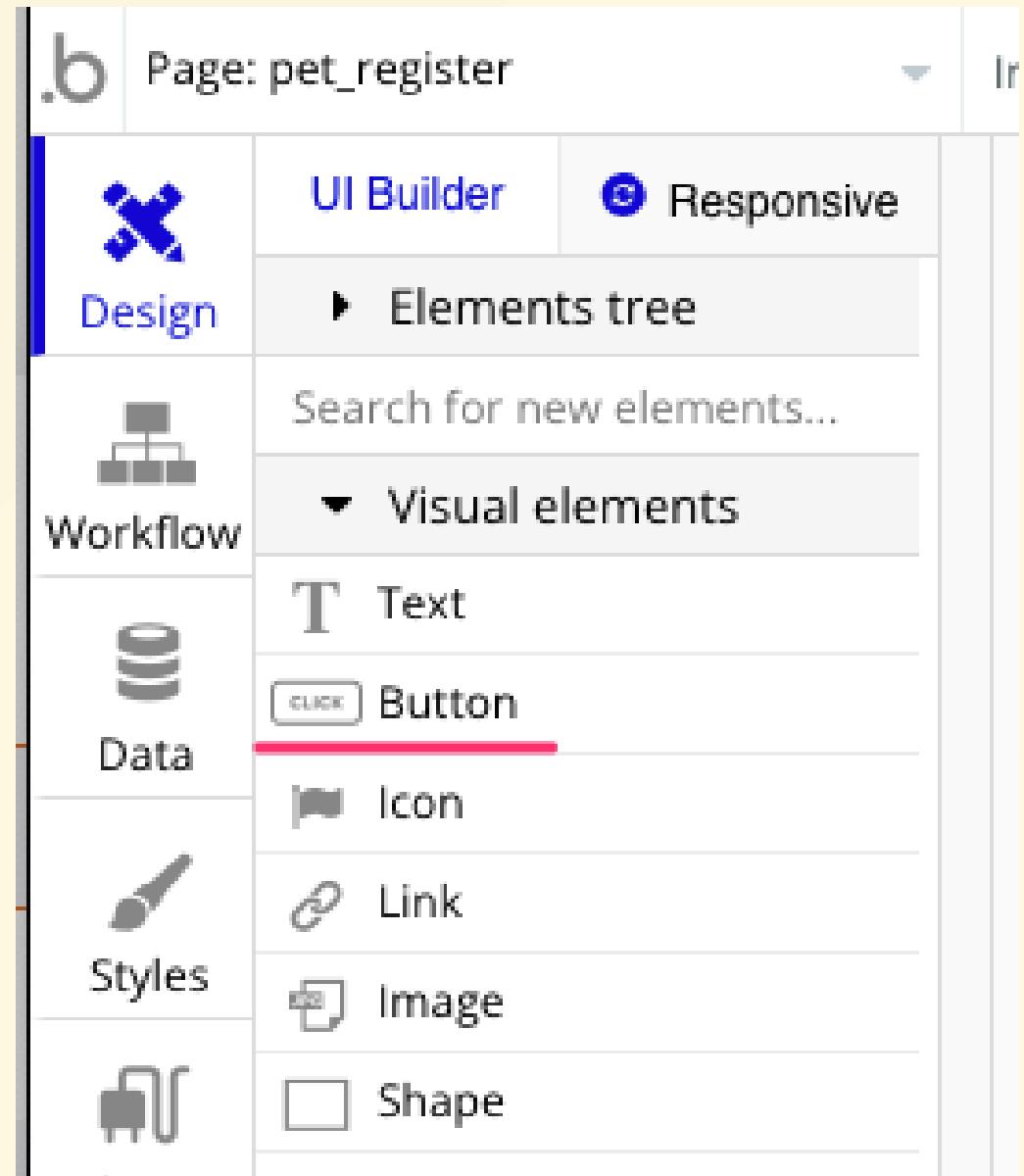


- Check This input should not be empty to make it mandatory.
- Specify Input pet gender as the name of the element.

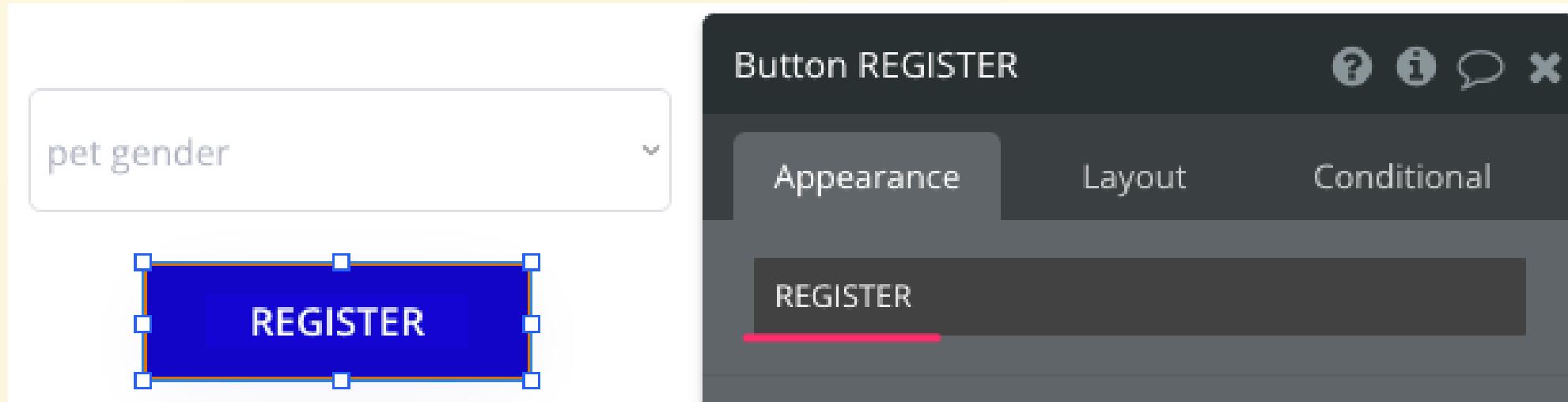


Registration Button

- Now that you have entered your pet's information, place the registration button
- Select the **Button** in the **Visual elements** in the left panel and click on it in the right panel to place the element.

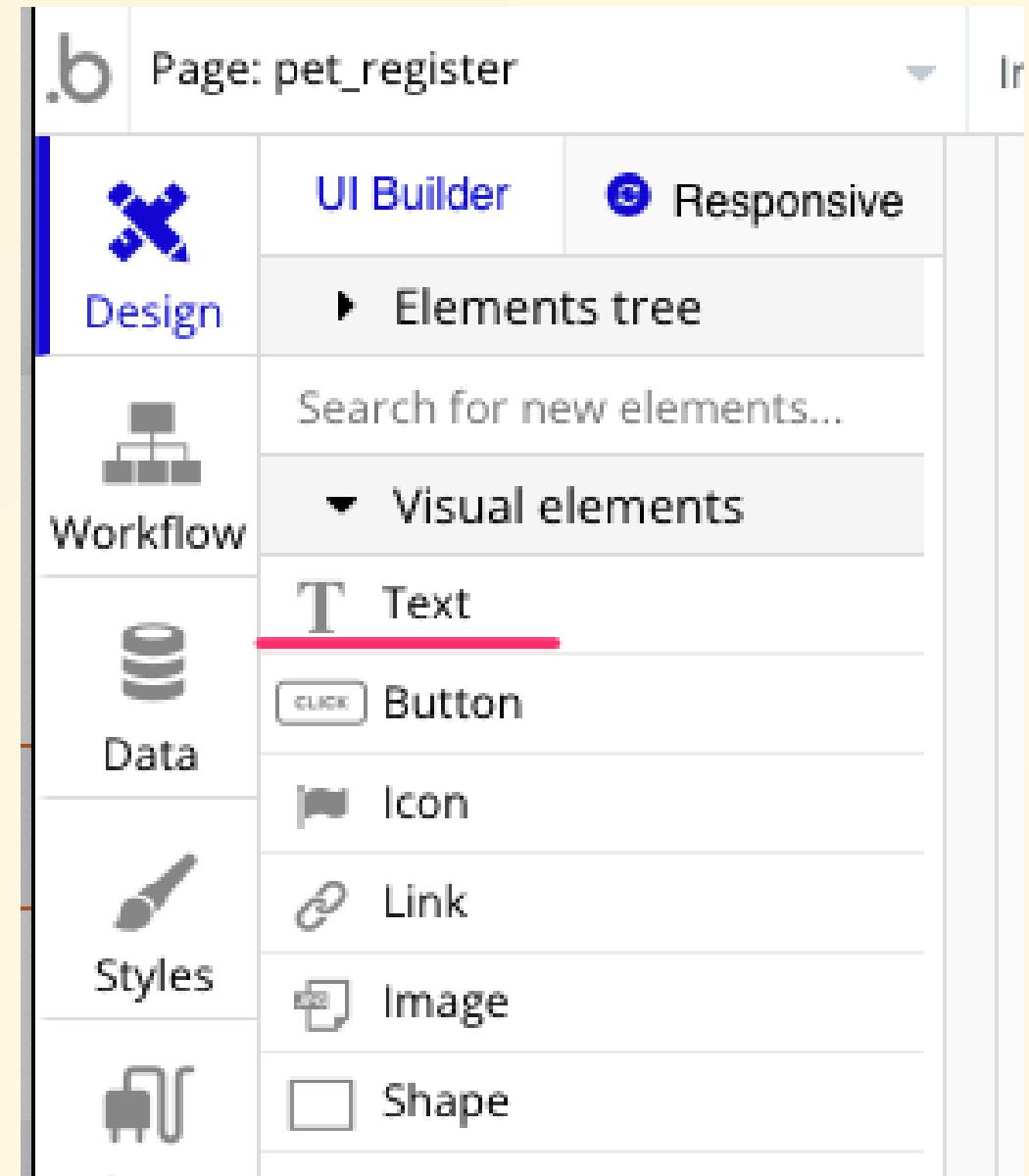


- Double-click on the element to set the details of the element
 - At the top of the Appearance tab, click "...edit me..." at the top of the Appearance tab. at the top of the Appearance tab and type "REGISTER"

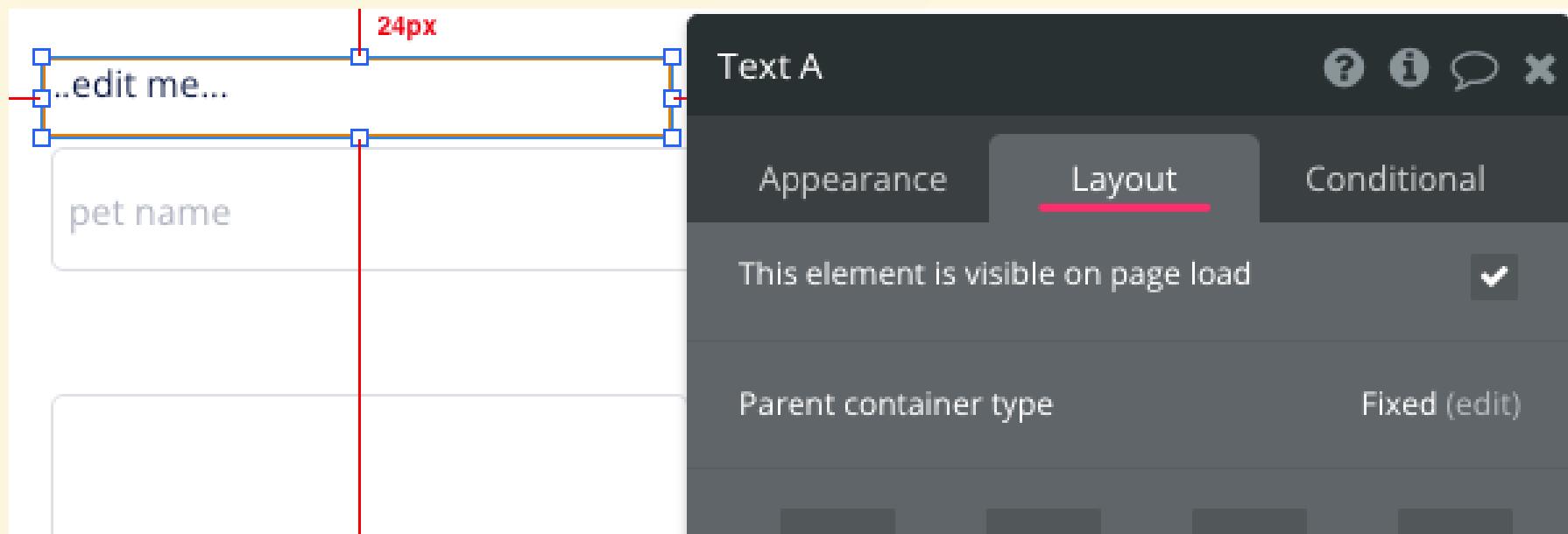


Label each input element

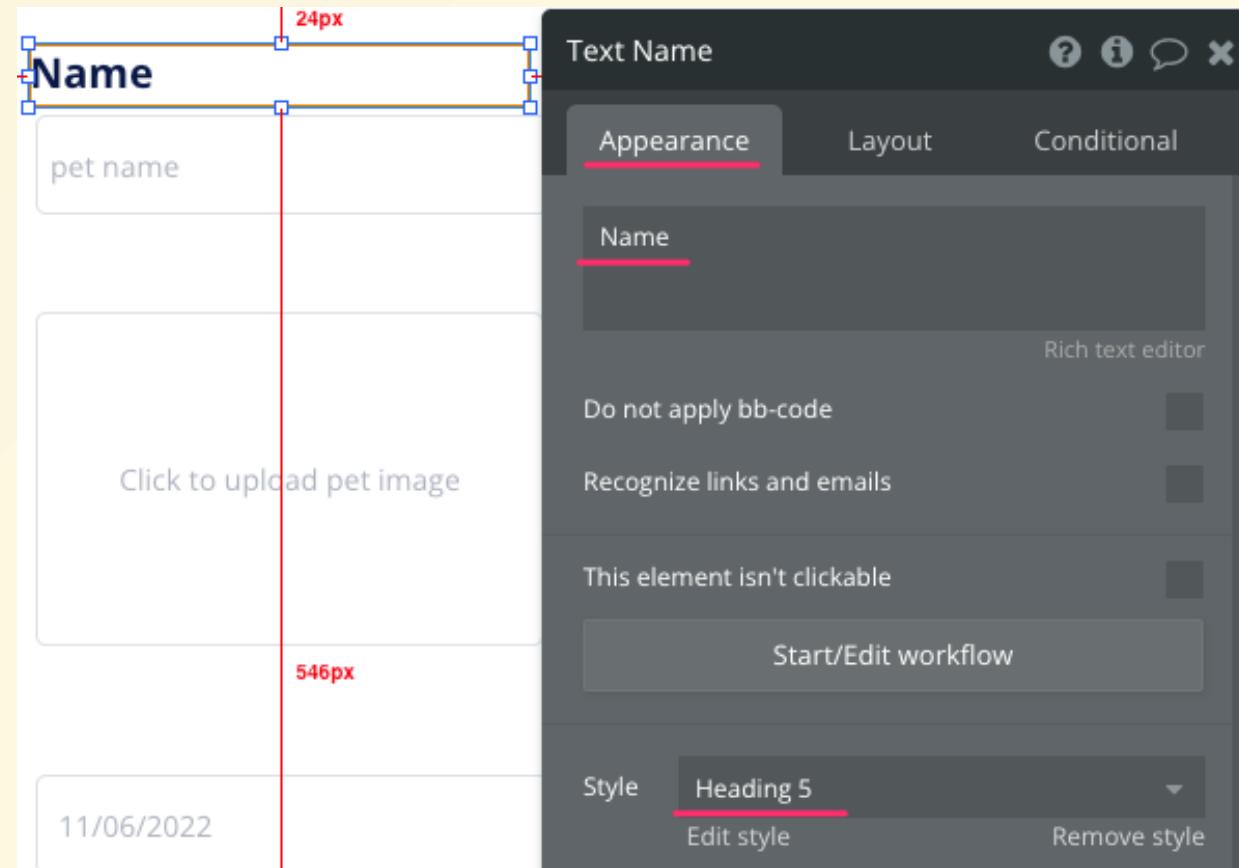
- Let's put a label in the upper left corner of each element, as shown in the Adalo screen
- Select **Text** in **Visual elements** in the left panel and drag the element in the right panel.



- When placing, make sure that the height of the element is at least 30px!
 - Otherwise, the contents of the element will not be displayed correctly!
 - You can also edit the **H** value from the Layout tab of the Advanced dialog after placing the element!



- This is also where you set the details of the element
 - At the top of the Appearance tab, click "....edit me..." at the top of the Appearance tab. at the top of the Appearance tab and type `Name
 - Select **Heading 5** for **Style** in the middle of the dialog.



- Let's add other labels in the same way

- Name
- Image
- Birthday
- Gender

The image shows a user interface for registering a pet. It consists of four input fields arranged vertically, each with a label above it. The first field is labeled "Name" and contains the placeholder "pet name". The second field is labeled "Image" and has a placeholder "Click to upload pet image". The third field is labeled "Birthday" and contains the date "11/06/2022". The fourth field is labeled "Gender" and contains the placeholder "pet gender". At the bottom right is a large blue button with the word "REGISTER" in white capital letters.

Name
pet name

Image
Click to upload pet image

Birthday
11/06/2022

Gender
pet gender

REGISTER

- The same procedure can be used, but you can save time by copying and using the label for the Name you just created
 - Copy (Ctrl + C) and paste (Ctrl + P) and rewrite only the label contents.
 - On Mac, copy (Command + C) and paste (Command + P)

The image shows a user interface for registering a pet. It consists of several input fields and a register button. The fields are labeled with underlined text: "Name", "Image", "Birthday", and "Gender".

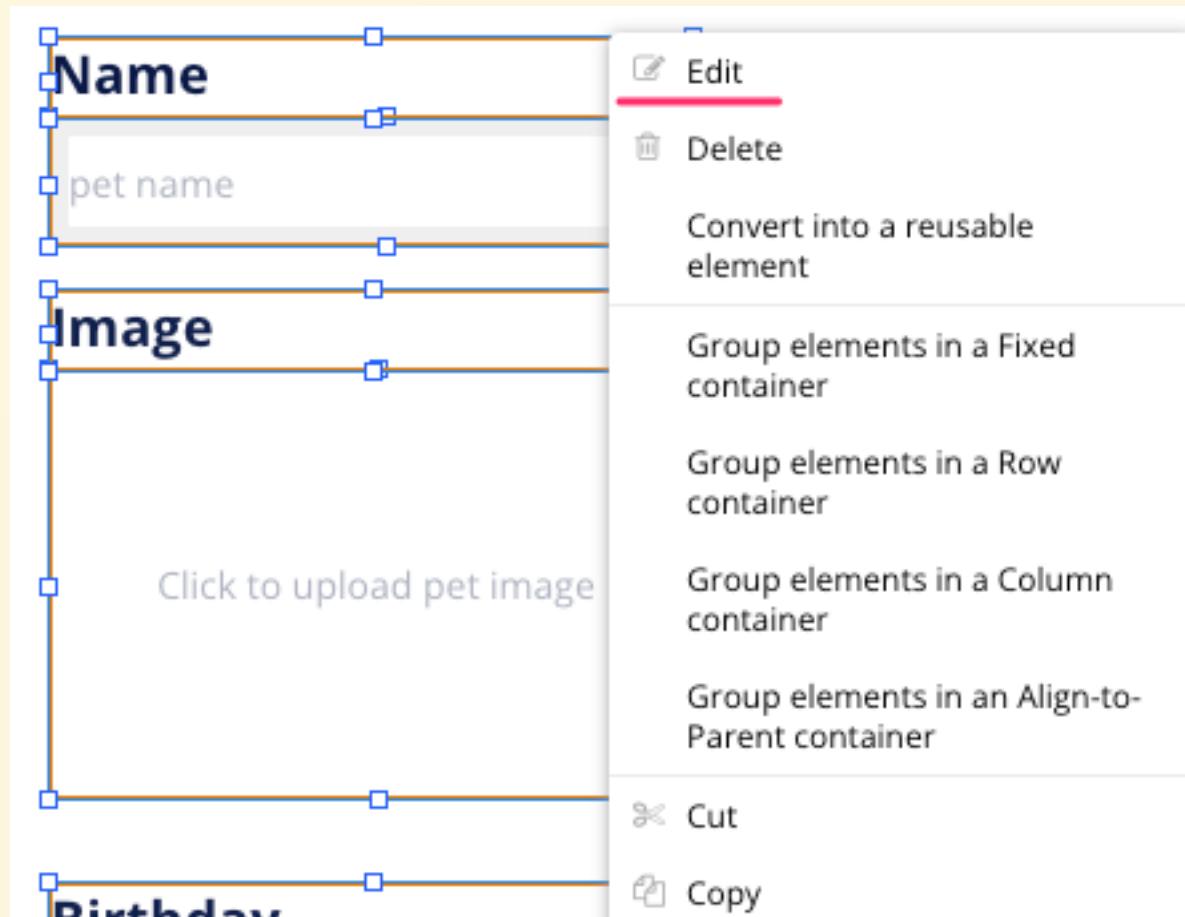
- Name:** A text input field containing the placeholder "pet name".
- Image:** A large rectangular area with a placeholder "Click to upload pet image".
- Birthday:** A text input field containing the date "11/06/2022".
- Gender:** A dropdown menu with the placeholder "pet gender".
- REGISTER:** A large blue button at the bottom right.

- Finally, unify the width of all elements to 220px
 - Double-click on each element and set the **W** (Width) value to 220 in the dialog.
 - This is to make it look better when displayed on the screen.

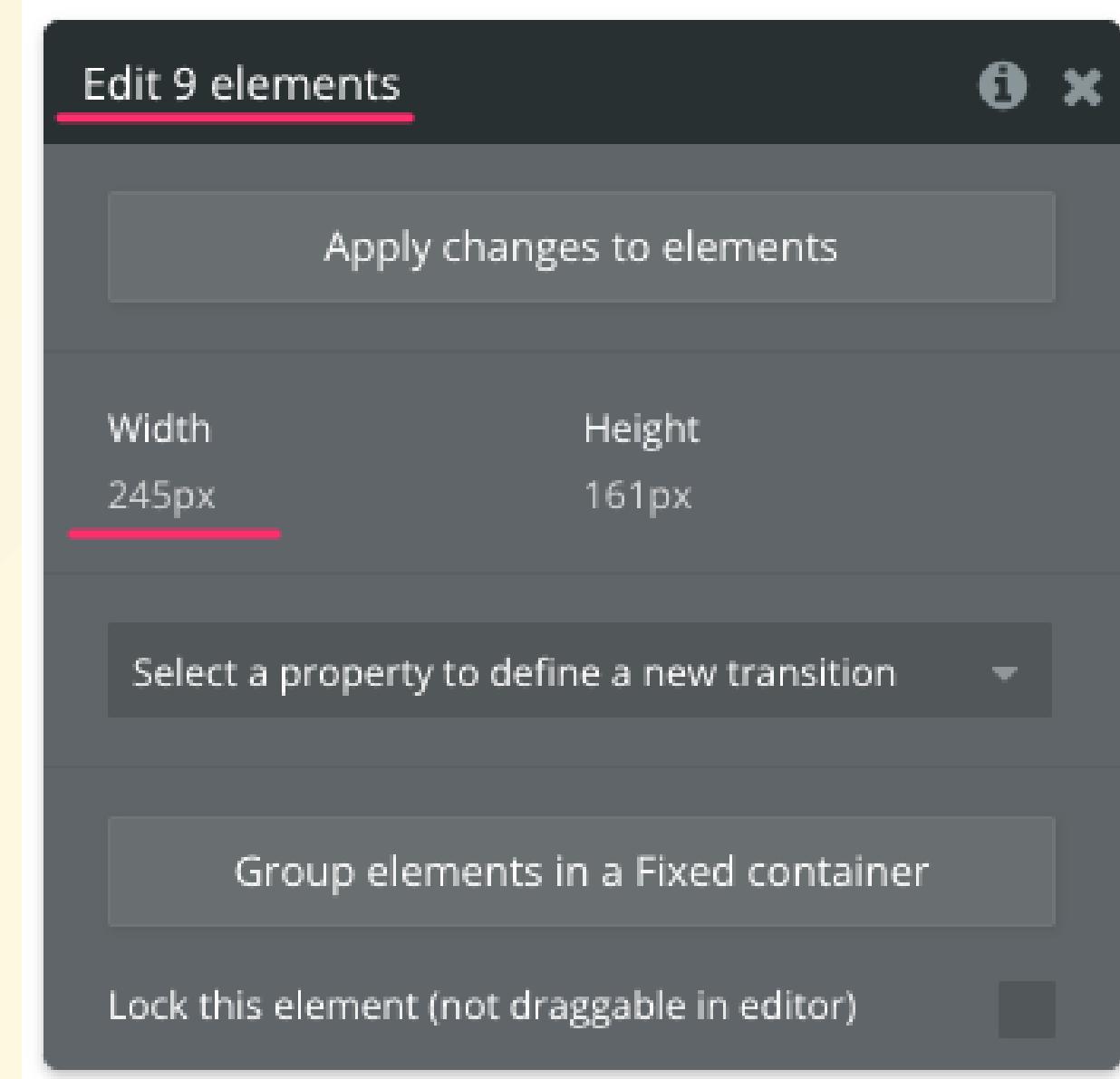
- You can change them one by one, but since there are so many elements, let's change them all at once!
- With all elements selected, right-click and select Edit
 - Select all elements by dragging

The image shows a user interface for registering a pet. It consists of several input fields and a large blue button. At the top left is a field labeled "Name" with the placeholder "pet name". To its right is a larger field labeled "Image" with the placeholder "Click to upload pet image". Below these is a field labeled "Birthday" with the placeholder "11/06/2022". Further down is a field labeled "Gender" with the placeholder "pet gender". At the bottom center is a large, solid blue rectangular button with the word "REGISTER" in white capital letters.

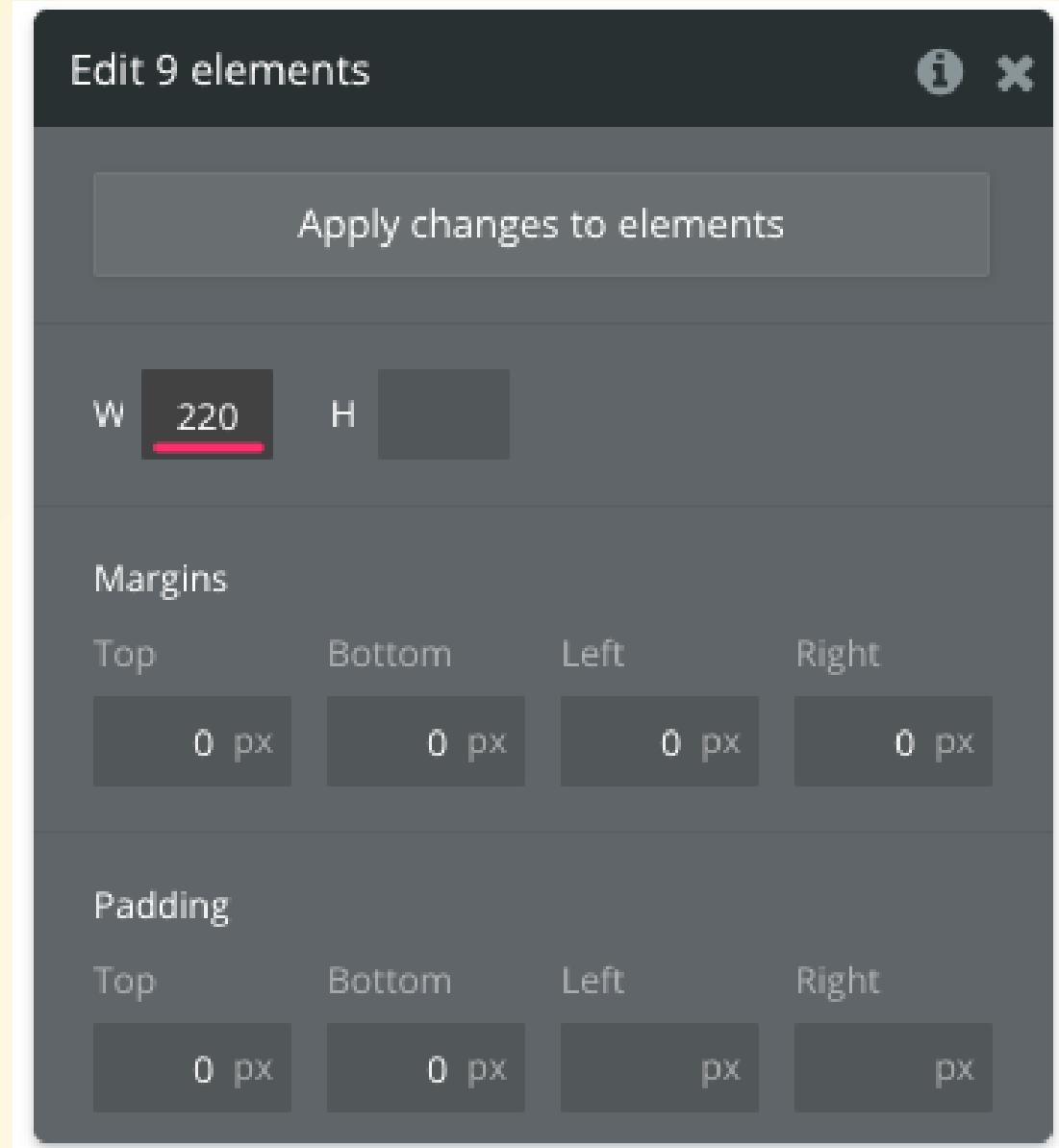
- With all of the elements selected, right-click on any one of them.
- Select Edit from the submenu that appears.



- Then a dialog box will appear, confirming that the number of elements in the selection is "9"
- If all is well, click on **Width** in the dialog box.



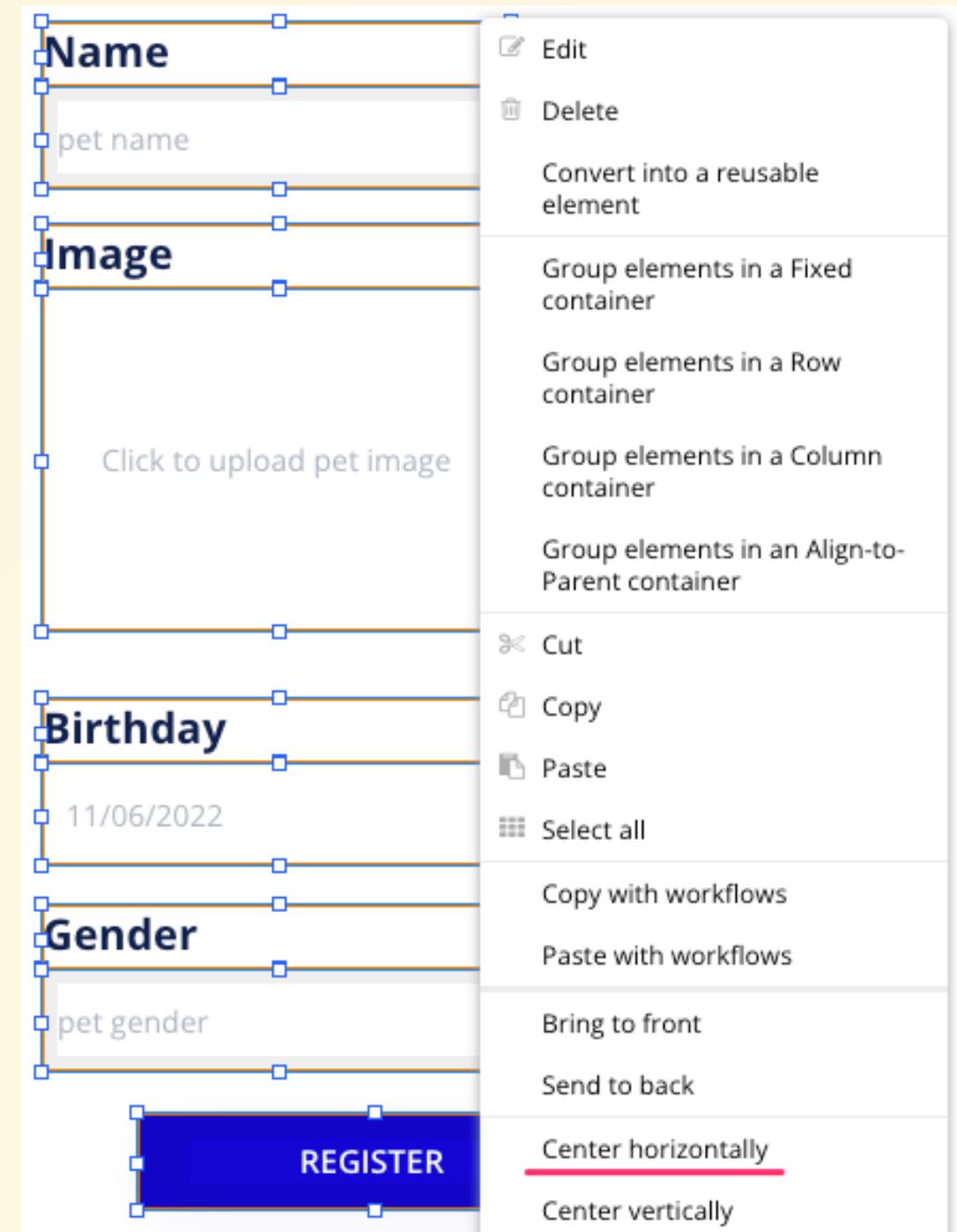
- Then you can set the width (W) and height uniformly for the elements in the wash, so enter 220 for W.
- Click on "Apply changes to elements".



Finally, organize and preview the elements

- As before, select all elements by dragging, right-click and select

Center horizontally to center the input form horizontally.



- When you're done, click **Preview** in the upper right corner of the screen to preview it and see how it works!

- Are all elements visible?
- Can values be entered in all input forms?
- If an image is specified, is it displayed?

Name

Image

Click to upload pet image

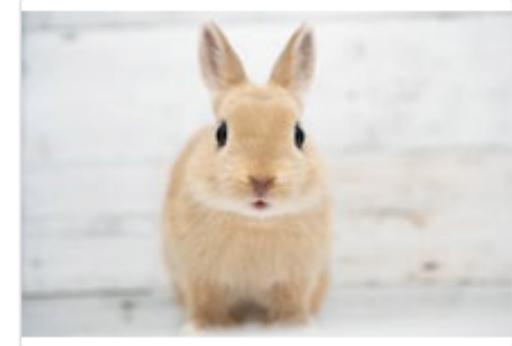
Birthday

Gender

REGISTER

Name

Image



Birthday

Gender

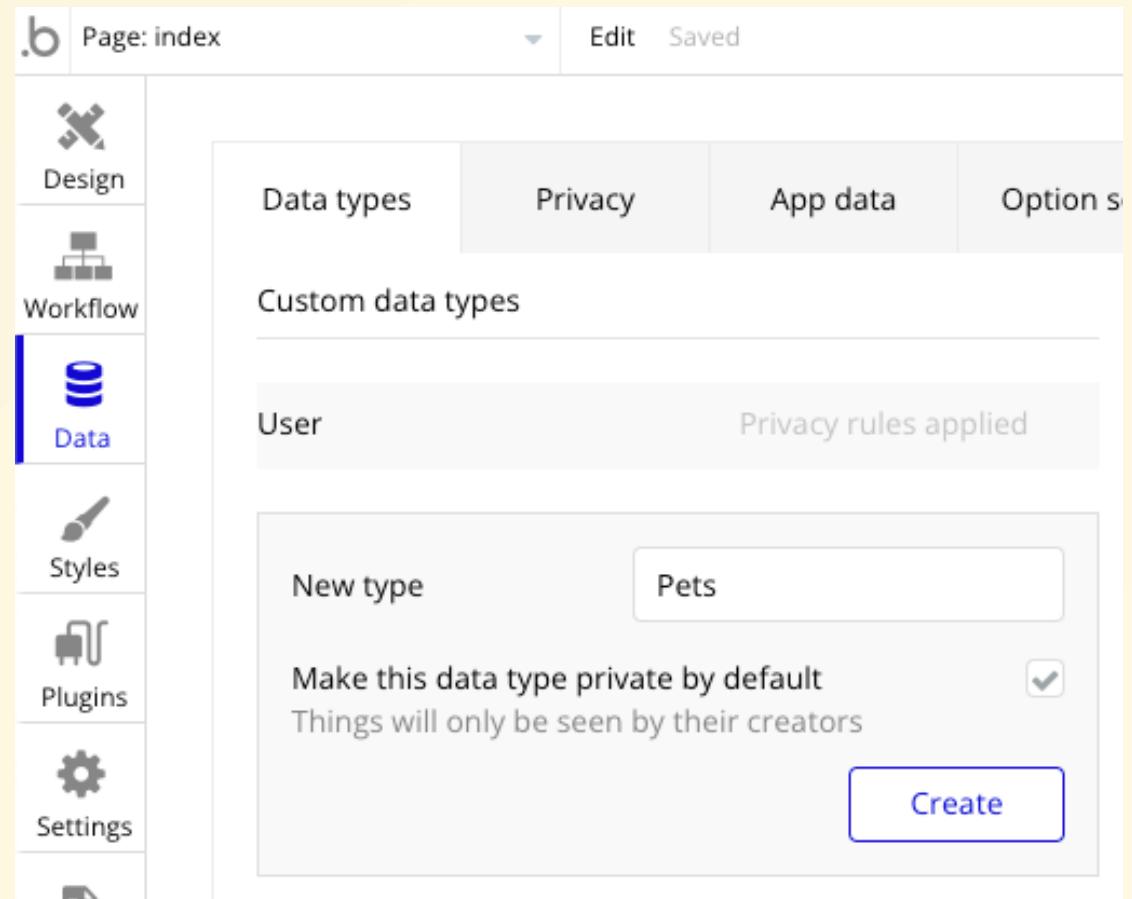
REGISTER

Let's connect to the database

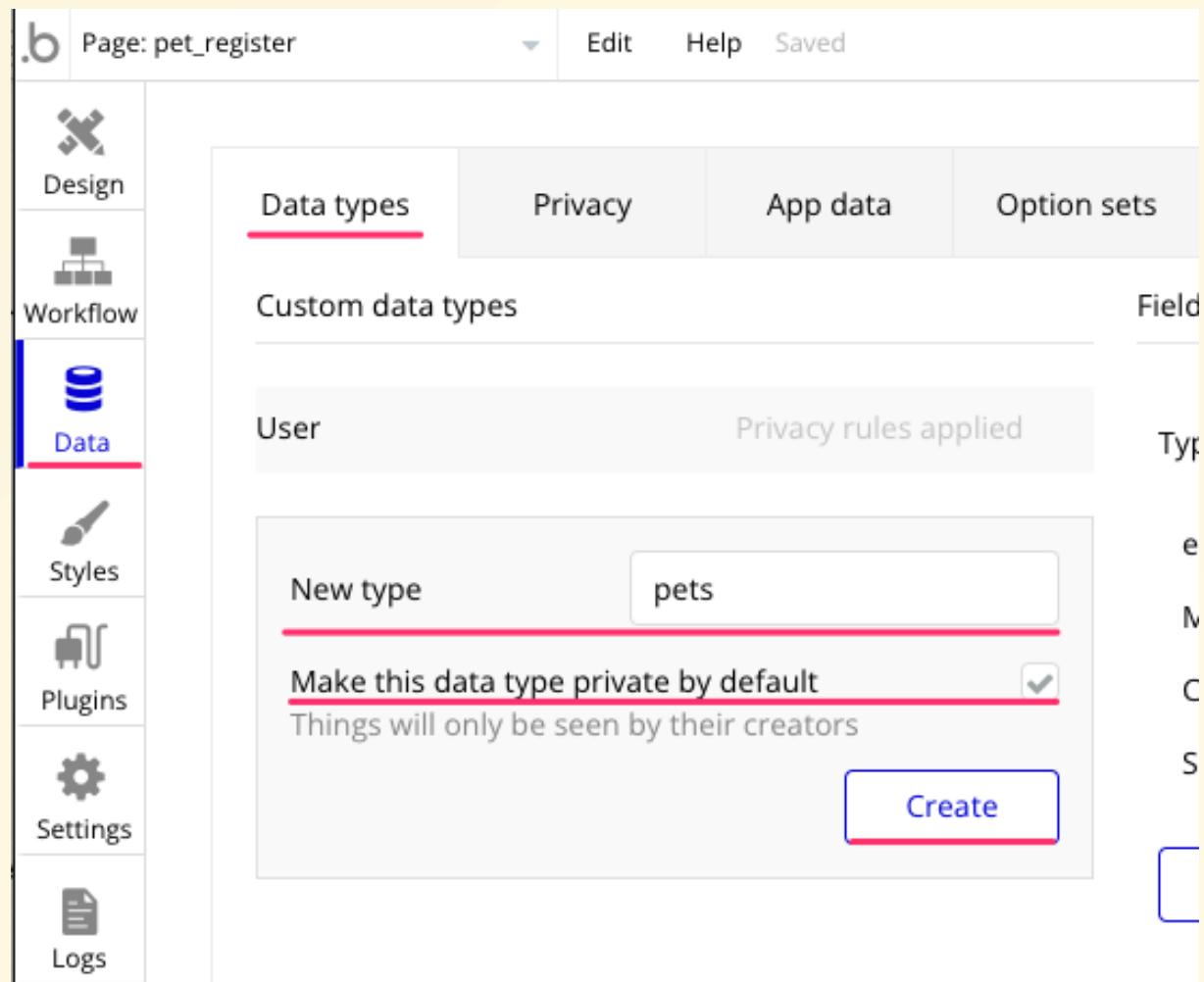
- Now that the screen layout is ready, it's time to connect to the database!

First, prepare a box to store your pet's information

- Select Data from the tabs in the left menu.
- Then the Data types tab will be activated, and under Custom data types, type "Pets" in the text box marked **New type**.



- Then check
Make this data type
private by default
and click the **Create**
button.



By the way...

- Adalo and Bubble have some differences in the way they refer to the various database elements
- Type: Defines the type of data (box).
 - Collection in Adalo
- Field: Element to represent the data type
 - Property in Adalo

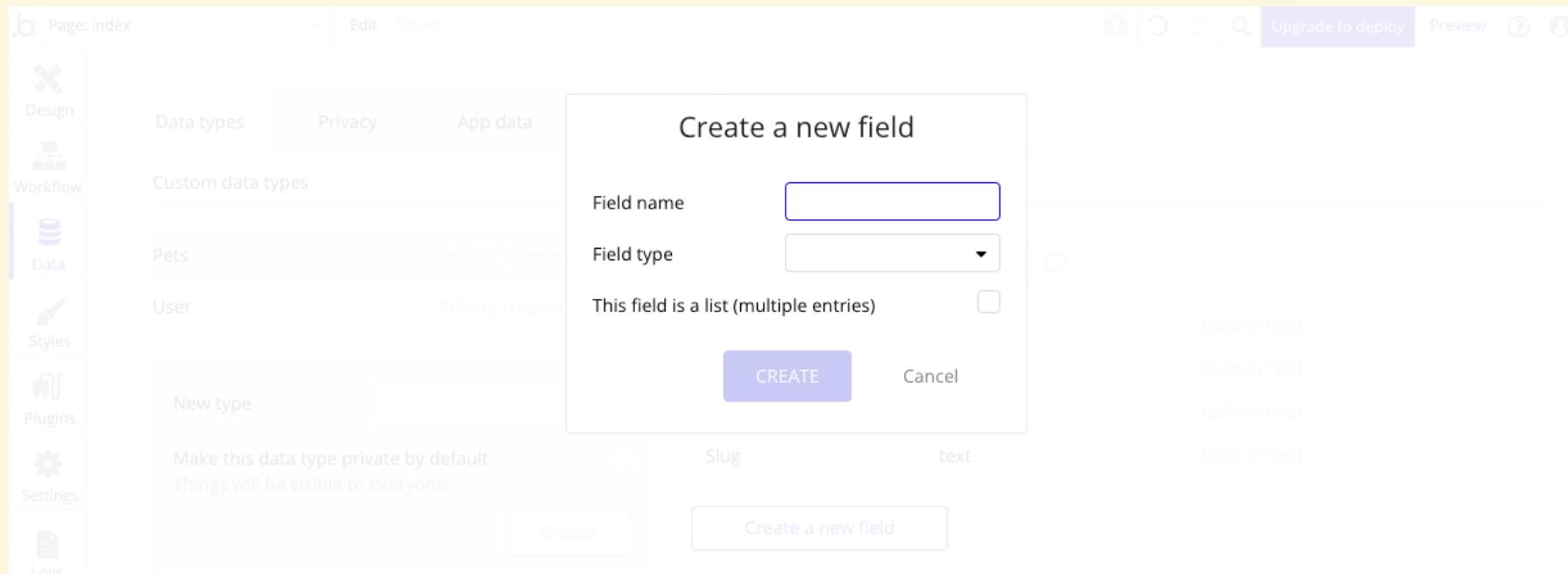
- Add elements (fields) to the created Pets type
- Select **Pets** from **Custome data types**.
- Click **Create a new field** under **Fields for type Pets** on the right side

The screenshot shows the Backstage application interface. On the left, there is a vertical sidebar with icons for Design, Workflow, Data (which is selected), Styles, Plugins, Settings, and Logs. The main area has a header with 'Page: index', 'Edit', 'Saved', and various navigation icons. Below the header, there are tabs for 'Data types', 'Privacy', 'App data', 'Option sets', and 'File manager'. A sub-header 'Custom data types' is followed by a list item 'Pets' which is highlighted with a red underline. To the right of the list items, there is a section titled 'Fields for type Pets'. It shows a table with four rows:

Type name	Pets	
Creator	User	Built-in field
Modified Date	date	Built-in field
Created Date	date	Built-in field

Below the table, there is a section for 'New type' with a text input field and a 'Create' button. At the bottom of this section, there is a note: 'Make this data type private by default' with a checkbox and the text 'Things will be visible to everyone'. A red box highlights the 'Create a new field' button at the bottom of the 'Fields for type Pets' section.

- A pop-up will then appear and you will enter the required information.



- Enter the name of the element in the **Field name** field.

Create a new field

Field name

Field type

This field is a list (multiple entries)

CREATE Cancel

- Select the element type (text, number, date, etc.) for the **Field type** !

Create a new field

Field name

Field type

This field is a list (multiple entries)

CREATE Cancel

- In Adalo, you could specify a relationship between types from Relationship
- In Bubble, registered types are also displayed in the **Field type** field.

Create a new field

Field name

Field type

This field is a list (multiple entries)

CREATE Cancel

Now let's add the necessary fields to the Pets type

- Name: text
- Image: image
- Birthday: date
- Gender: text

Fields for type Pets

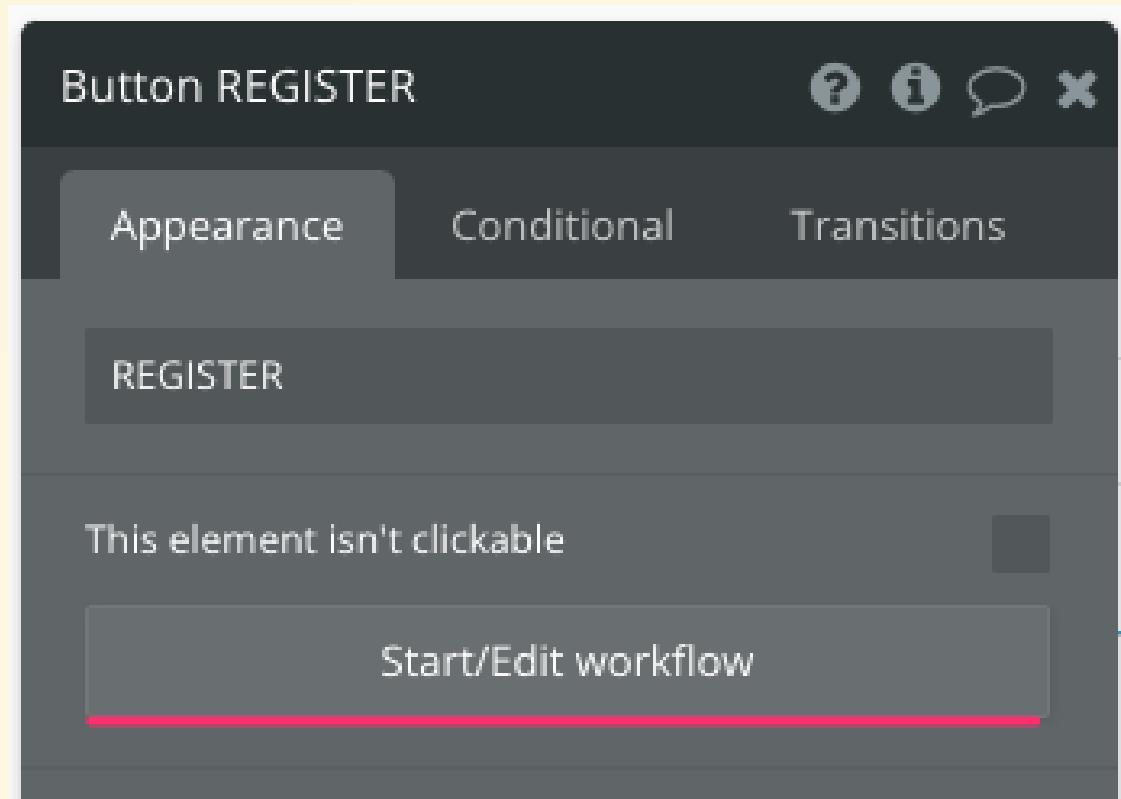
Type name	Pets	
Birthday	date	default <input type="text"/> trash comment
Gender	text	default <input type="text"/> trash comment
Image	image	default <input type="button" value="Upload"/> comment trash
Name	text	default <input type="text"/> comment trash
Creator	User	Built-in field
Modified Date	date	Built-in field
Created Date	date	Built-in field
Slug	text	Built-in field

Create a new field

Enabling the registration of pets when the registration button is clicked

- Now that the box for storing pet information is ready, it is time to add a movement to save the information to the database when the registration button is clicked.
- In Bubble, we will use the Workflow tab to set up all the movements.
- You can also start from the Workflow tab, but this time, we will start from the button that will be the source of the movement.

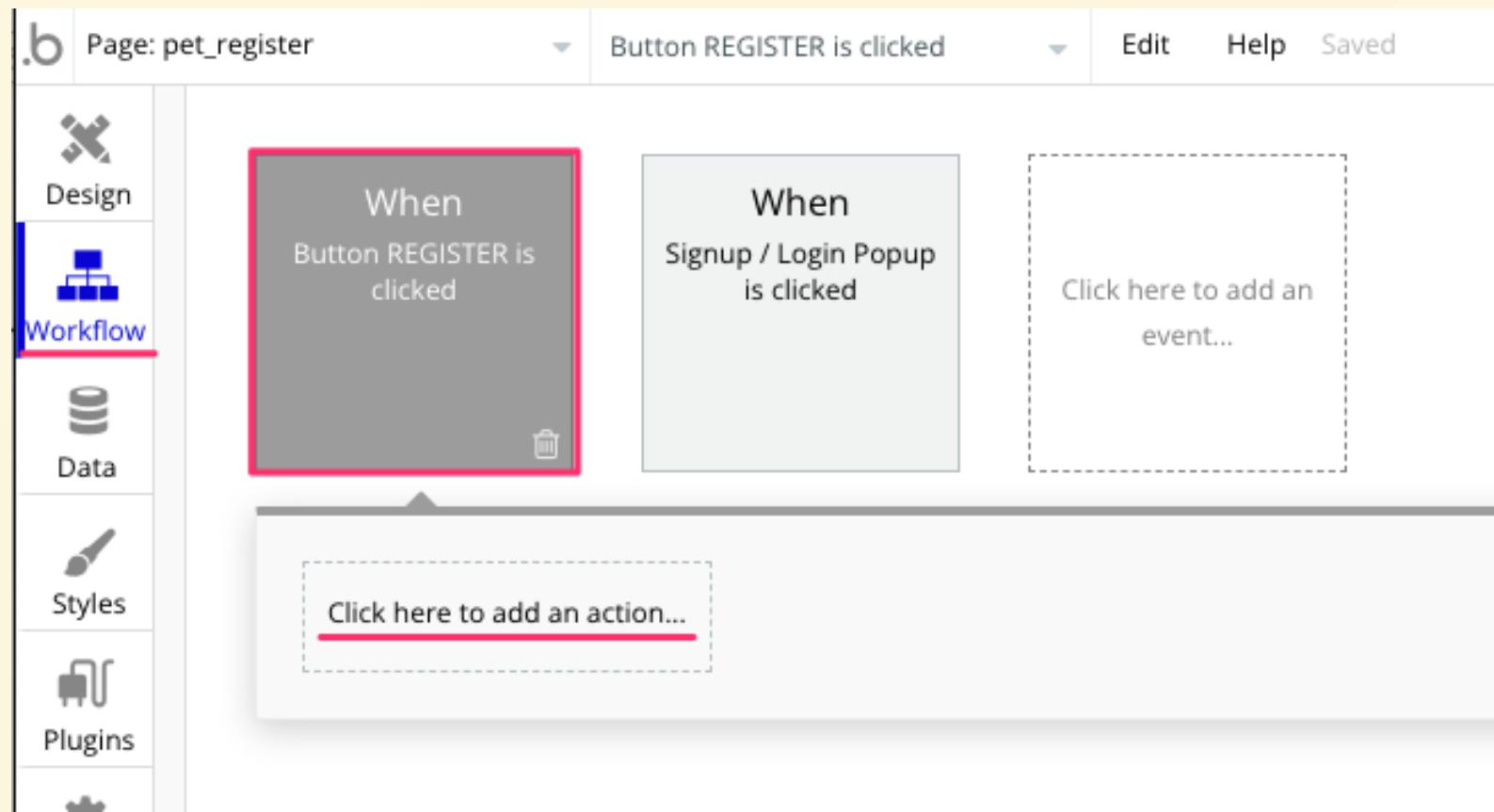
- Select the Design tab from the left menu
- Select the "pet_register" screen from the upper left panel
- Double-click the "REGISTER" button on the "pet_register" screen from the right panel
- Click Start/Edit workflow in the "Appearance" tab.



- You will then switch to the Workflow tab, where you will see **Button REGISTER is clicked** in the When section.
 - This is literally the workflow definition when the "REGISTER" button is clicked.

The screenshot shows a workflow editor interface. At the top, it says "Page: pet_register" and "Button REGISTER is clicked". The top navigation bar includes "Edit" and "Help". On the left, there's a sidebar with icons for "Design", "Workflow" (which is selected), "Data", "Styles", and "Plugins". The main workspace contains three boxes. The first box, "When Button REGISTER is clicked", has a red border around it. The second box, "When Signup / Login Popup is clicked", is a standard grey box. Below these is a dashed box with the text "Click here to add an event...". At the bottom, there's another dashed box with the text "Click here to add an action...".

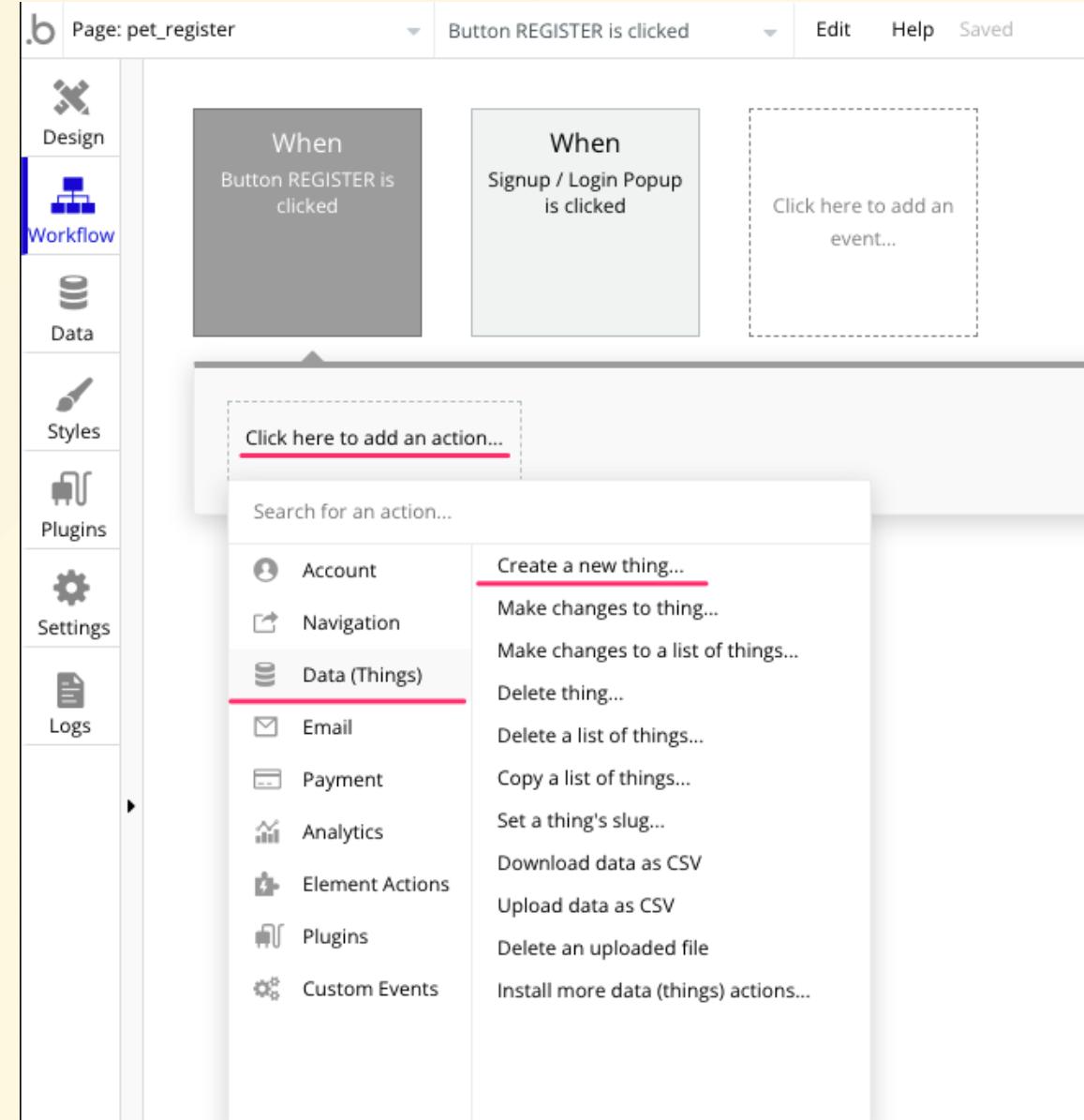
- So, select **Click here to add an action...** to set the behavior when the button is pressed



- Click here to add an action...

Click here to add an action...` to display a popup window where you can specify various actions.

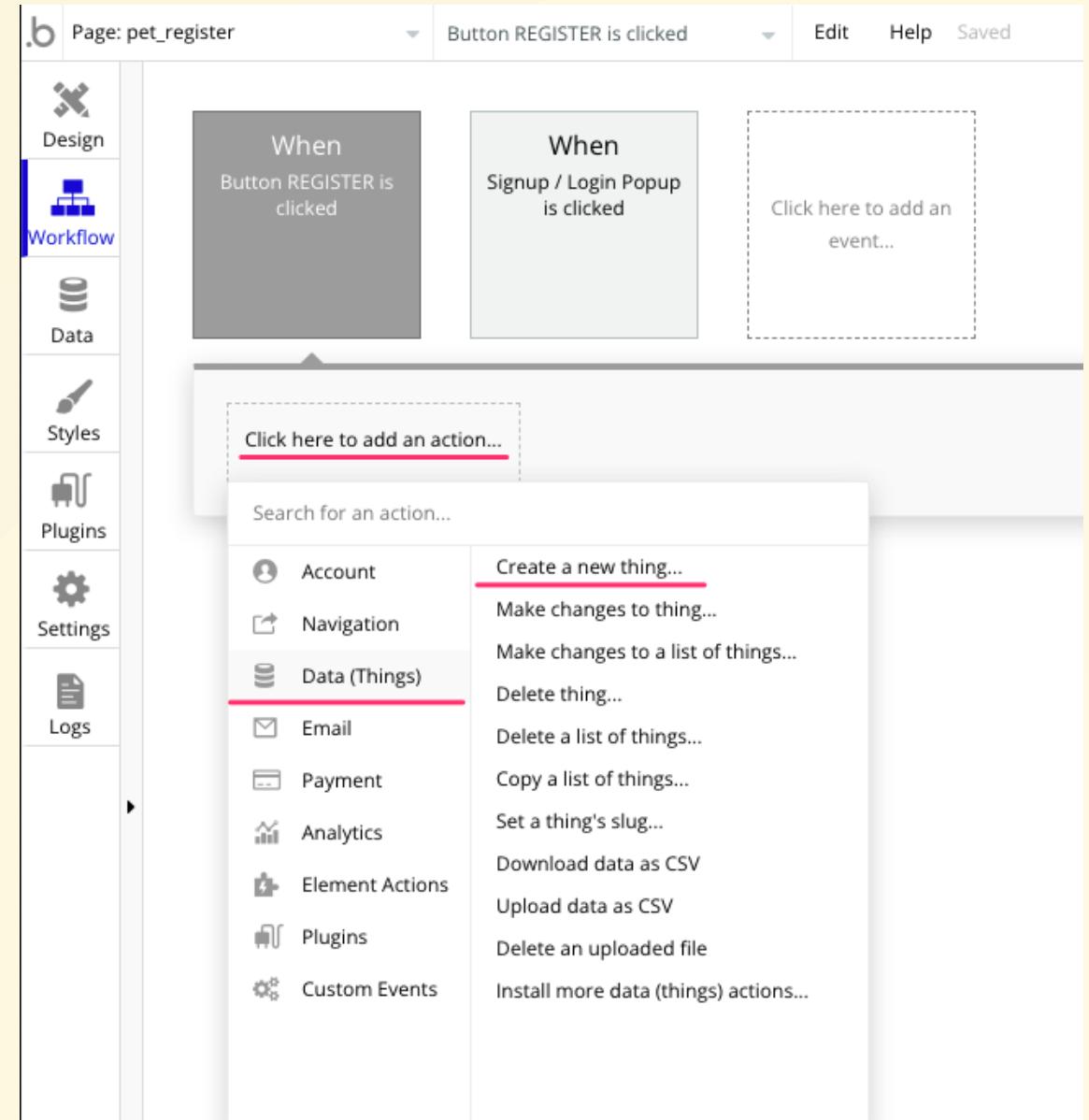
- Select **Data(Things)** for database operations (actions) like this one.



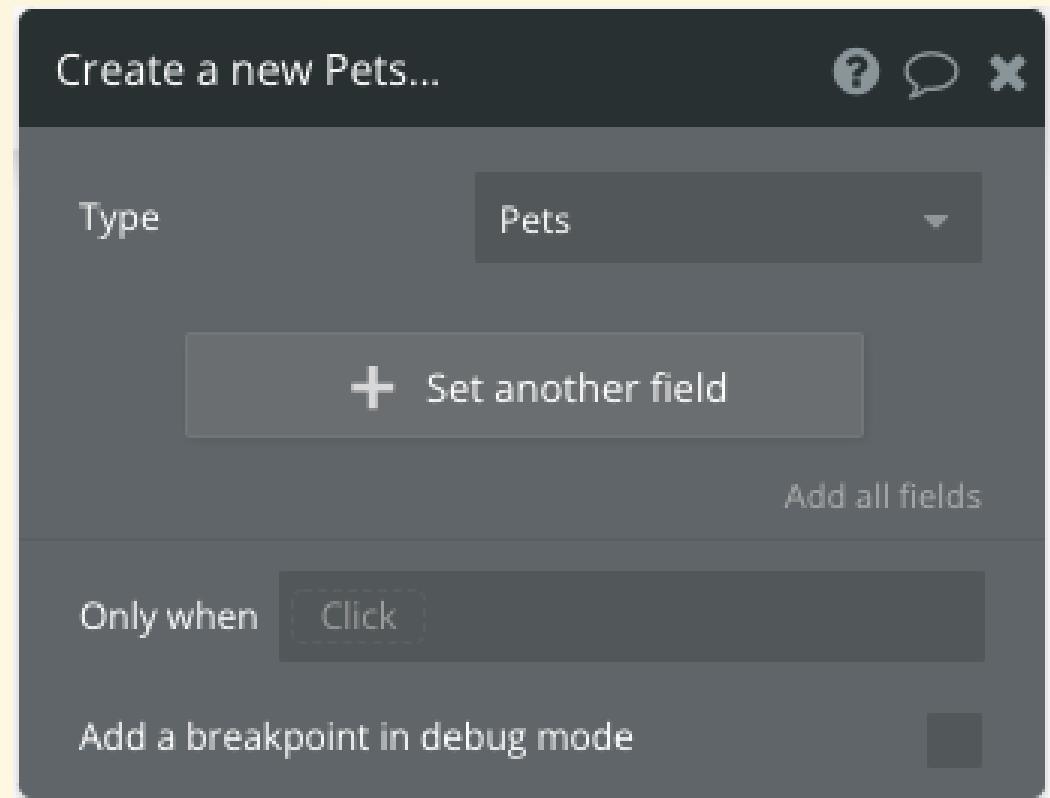
- Then you will see more sub-elements, select

Create a new thing...
Select

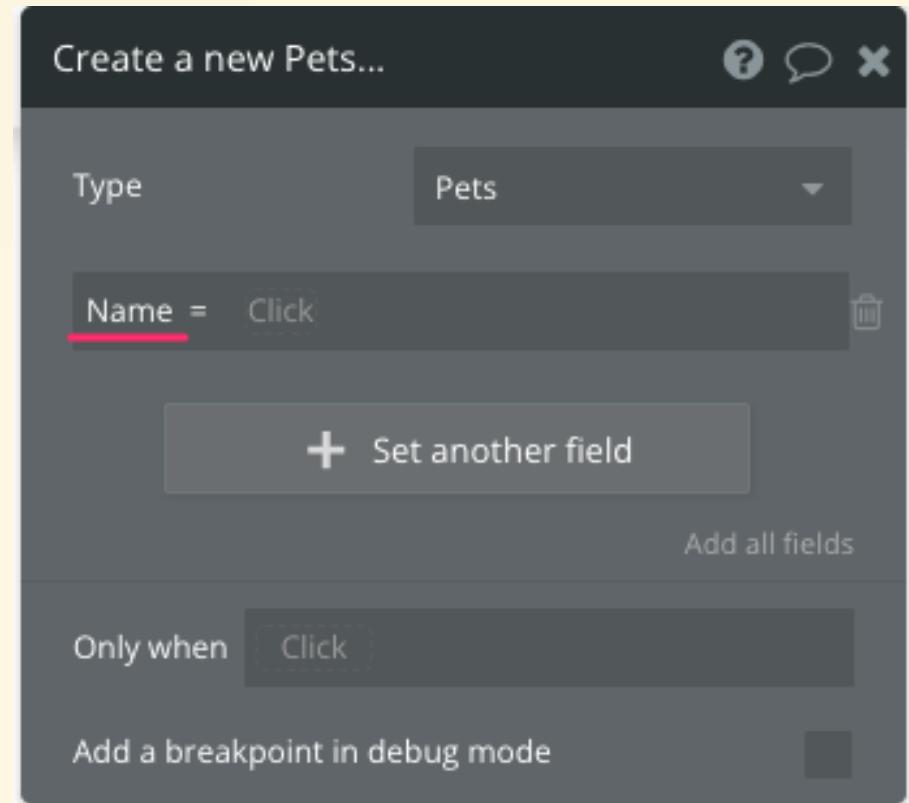
Create a new thing...` from
the list.



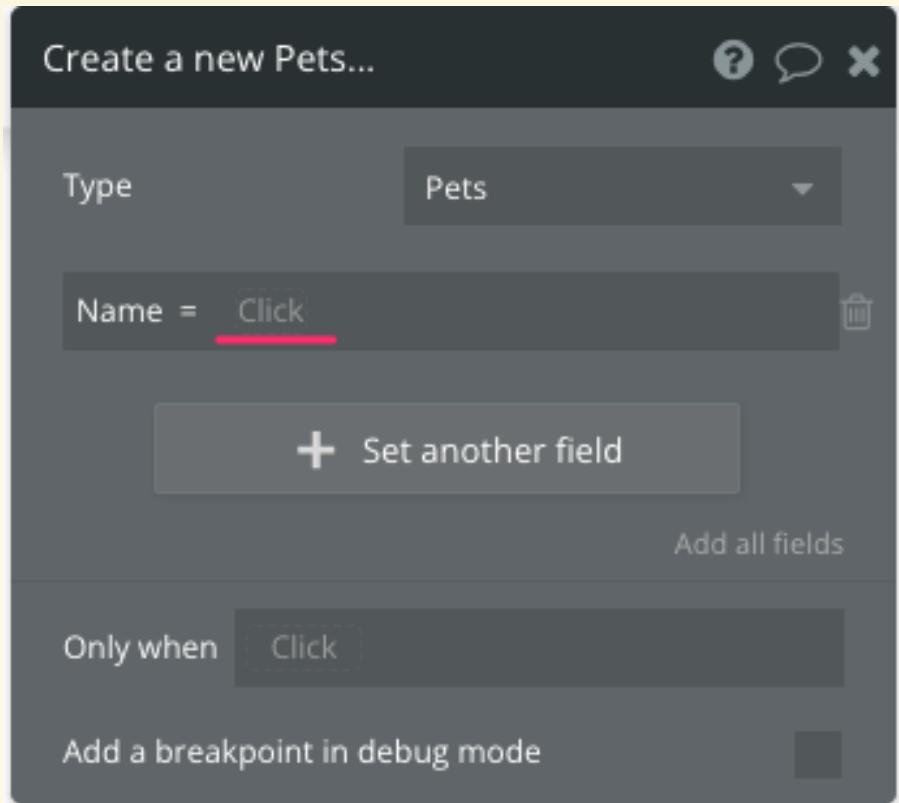
- The **Create a new thing...** dialog box will appear and you need to specify the type of registration you want to make.
 - In this case, it is Pets.
- Then, a button **Set another field** will appear, and you will set the field to the Pets "field" as shown in the title of the dialog.



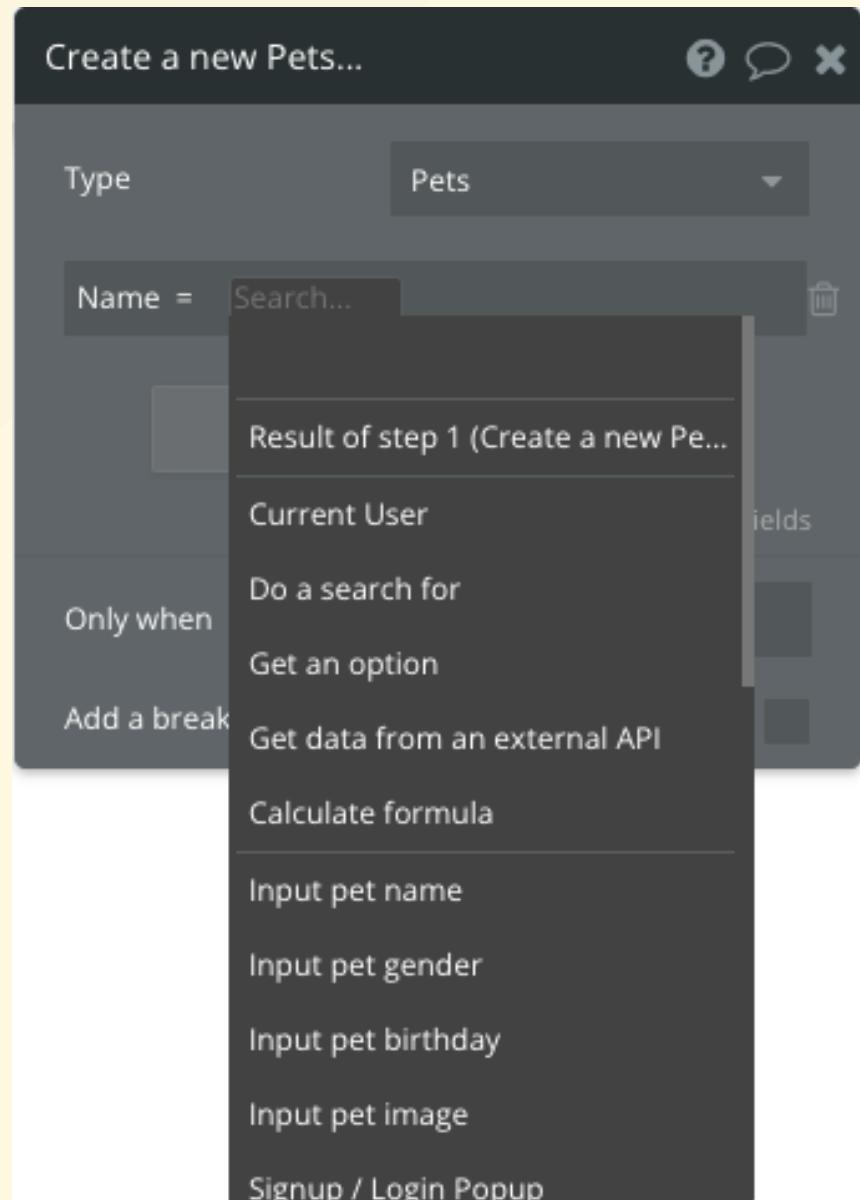
- Here we specify the "field" name of the item we want to store in Pets on the left-hand side, and the value we actually want to store in that "field" on the right-hand side
 - First, select Name for the left-hand side



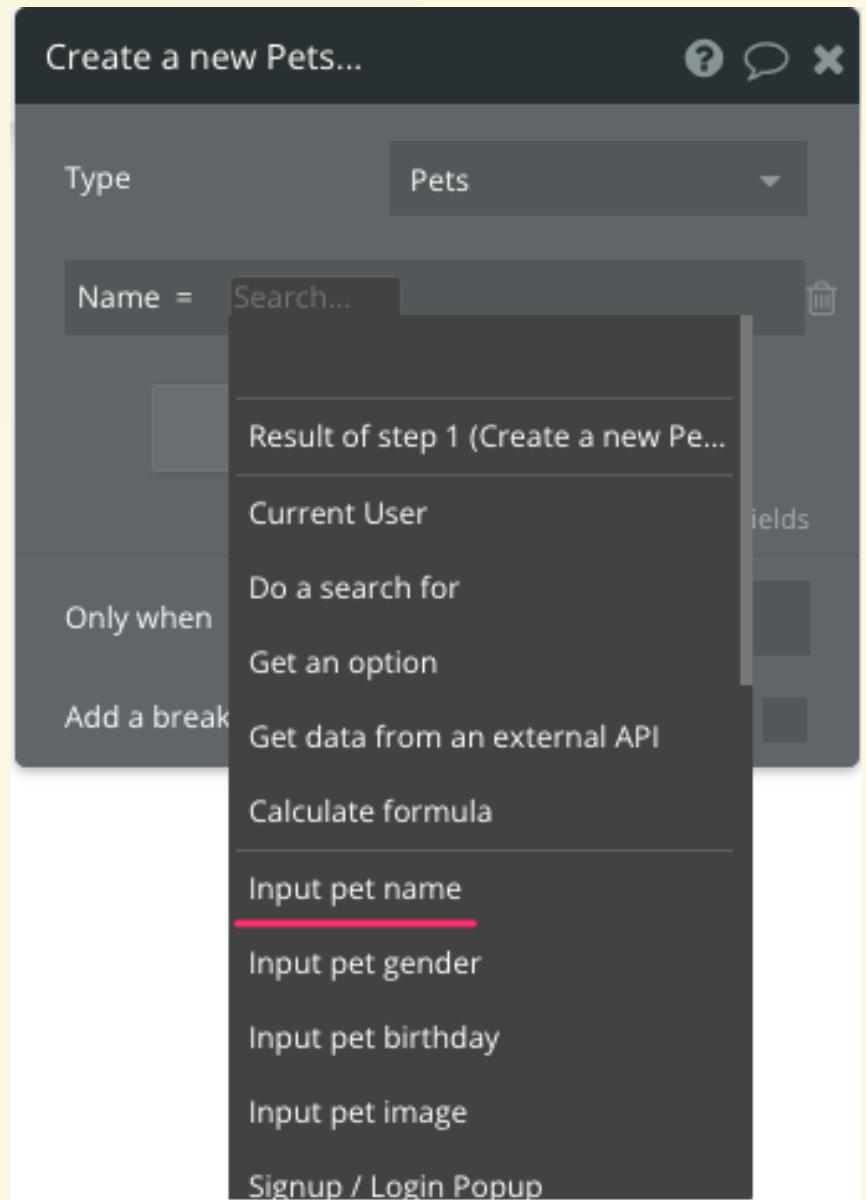
- If you specify the left side,
you will see **=** and now
Click on the right side.



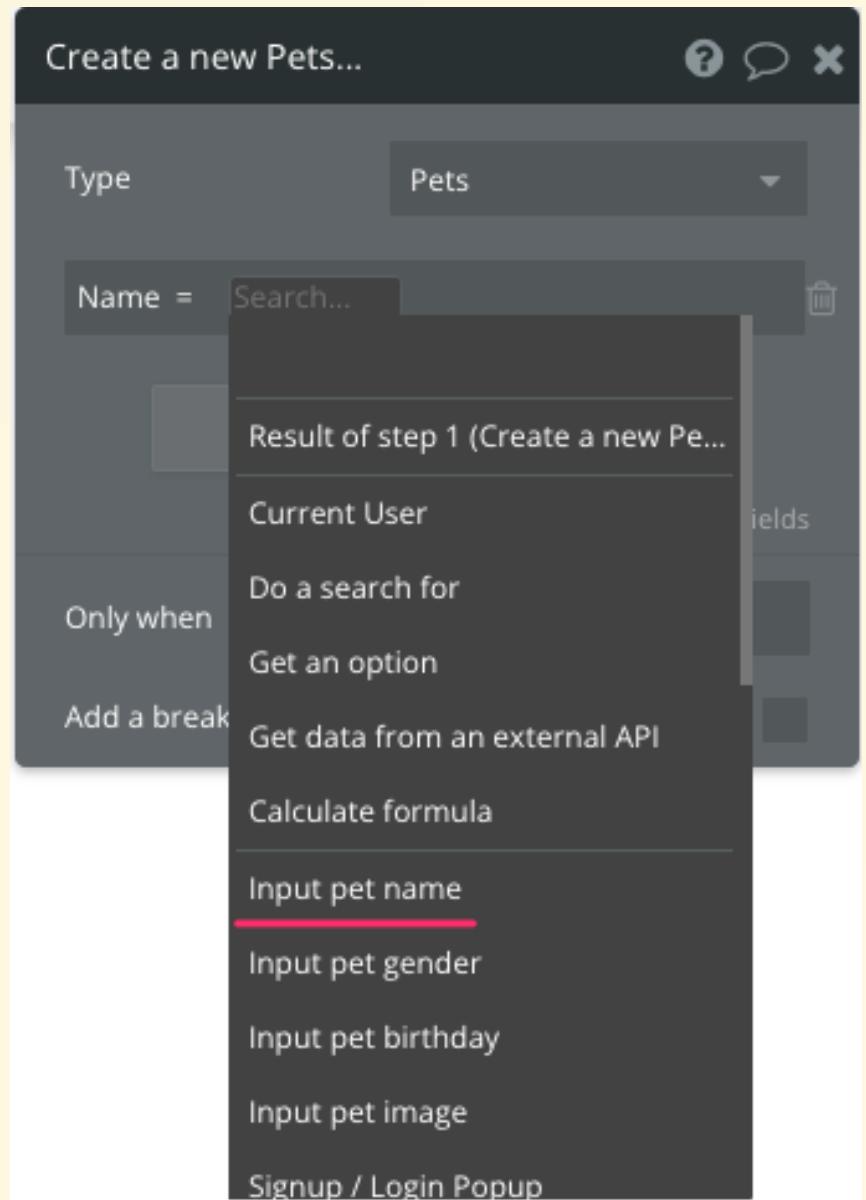
- Click here and you'll see a variety of options.
- I won't bore you with the details here, but you can specify what you want to do with the fields selected on the left side.



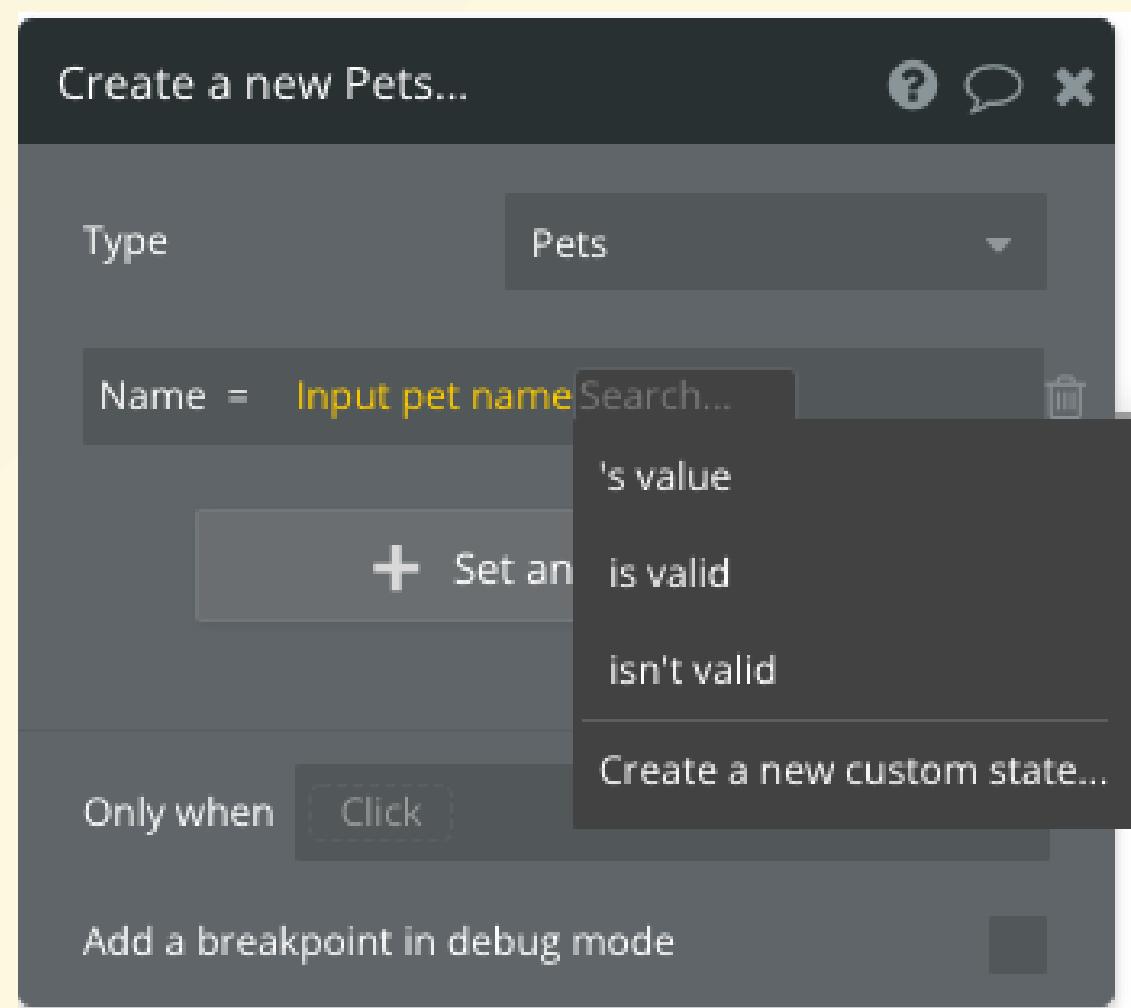
- Select the **Input pet name** field, which will be the name of the pet entered on the screen this time.



- The `Input pet name` is the name you gave to the input element when you created the screen.
- This makes it easier to uniquely identify input elements, for example, when multiple input elements exist on a single screen.



- Selecting **Input pet name** will display more options
- In this case, we want to use the **value** of the pet name entered on the screen, so we select **'s value**.
 - The other options, **is valid** and **isn't valid**, are used when you want to check if the entered value is valid or not.



- When all are specified, it looks like this

Create a new Pets...

Type Pets

Name = Input pet name's value

Image = Input pet image's value

Birthday = Input pet birthday's value

Gender = Input pet gender's value

+ Set another field

Only when Click

Add a breakpoint in debug mode

Let's make it move!

- Now that you've set up your pet, let's see if it actually registers in the preview!
 - By the way, you can preview by clicking "Ctrl + P" (shortcut key)
- Fill in all the input elements and click the "REGISTER" button!
- Nothing special happens on the screen, but let's see if the data is actually registered!

- Close the preview, open the Data tab, select App data, and make sure that the information you just entered from the screen is saved when you select **All Pets** in the left panel.

The screenshot shows the Bubble app interface with the following details:

- Page:** pet_register
- Tab:** App data (selected)
- Database views:** Application data - All Pets - Development version
- Buttons:** New view, Primary fields, Search, 1 entries (displaying 1), New entry, Delete (0), Upload, Modify, Export, Bulk
- Search bar:** Search view names or data types...
- Table:** Shows a single entry for a pet named "チロ".

	Birthday	Gender	Image	Name	Created Date	Modified Date	Slug
<input type="checkbox"/> <input type="text" value="Nov 11, 2022 12:00 am"/>	Nov 11, 2022 12:00 am	Female		チロ	Nov 12, 2022 1:17 pm	Nov 12, 2022 1:17 pm	
- Left sidebar:** Design, Workflow, **Data** (selected), Styles, Plugins, Settings
- Top right:** Upgrade to deploy, Preview, Help, User icon

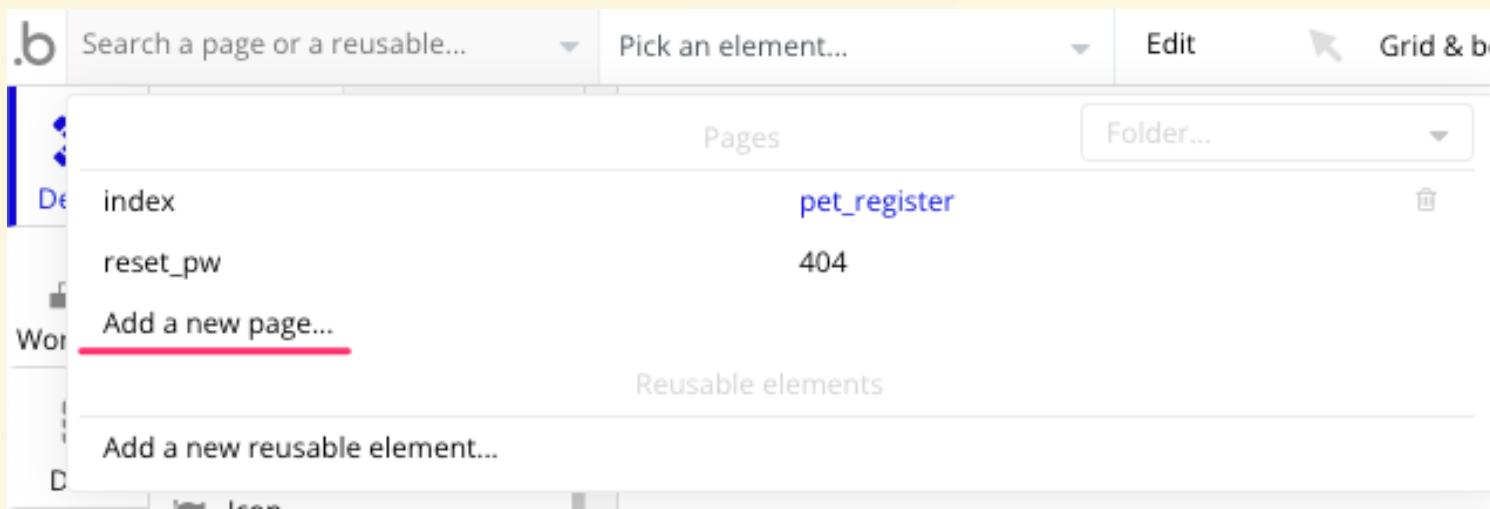
- Now I'm down to saving what I type on the screen to the database!



- Next, let's try to create the screen transitions!

Prepare only the frame of the pet list page

- First of all, prepare only the frame of the pet list page.
- Click **Page: index** in the upper left corner of the screen to display a popup window.
- Click **Add a new page...** in the pop-up window. ` in the pop-up window.



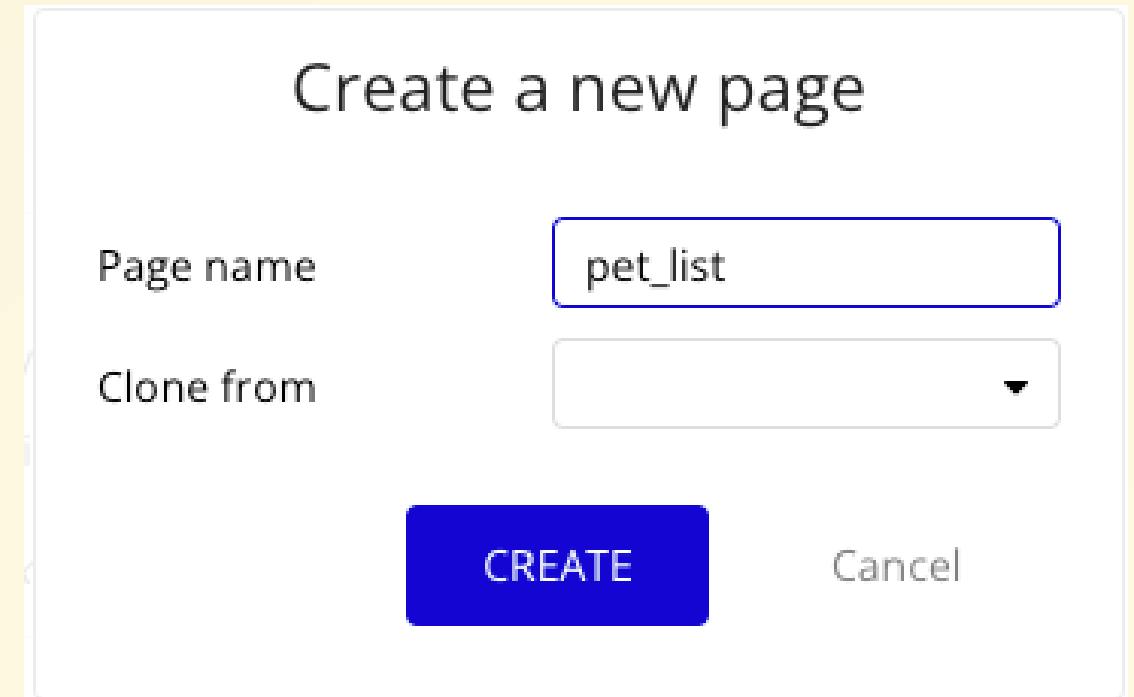
- Add a new page... to pop up a new page
- Page name should be pet_list.
- Clone form is left blank this time
- Click the "CREATE" button.

Create a new page

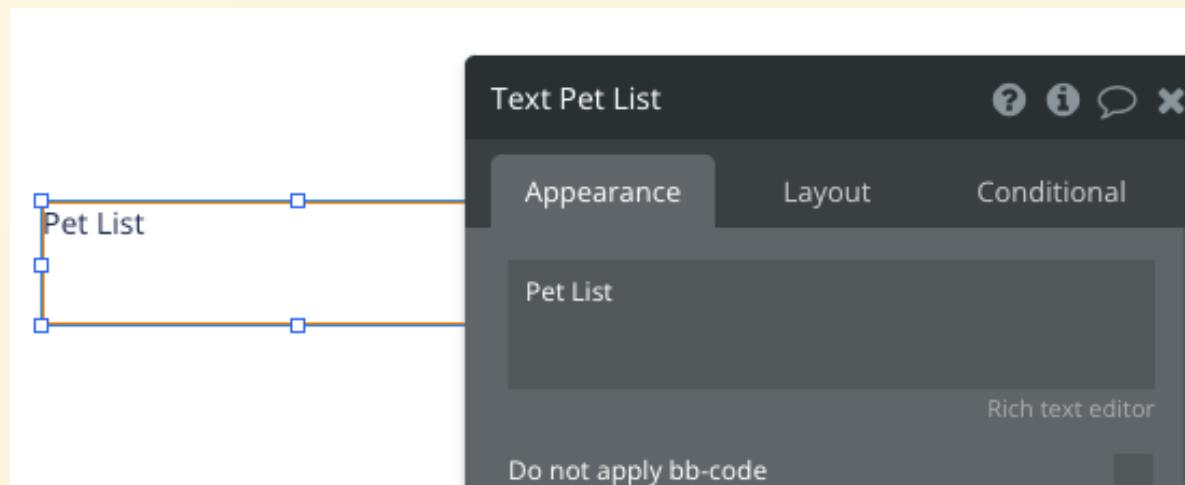
Page name

Clone from

CREATE Cancel

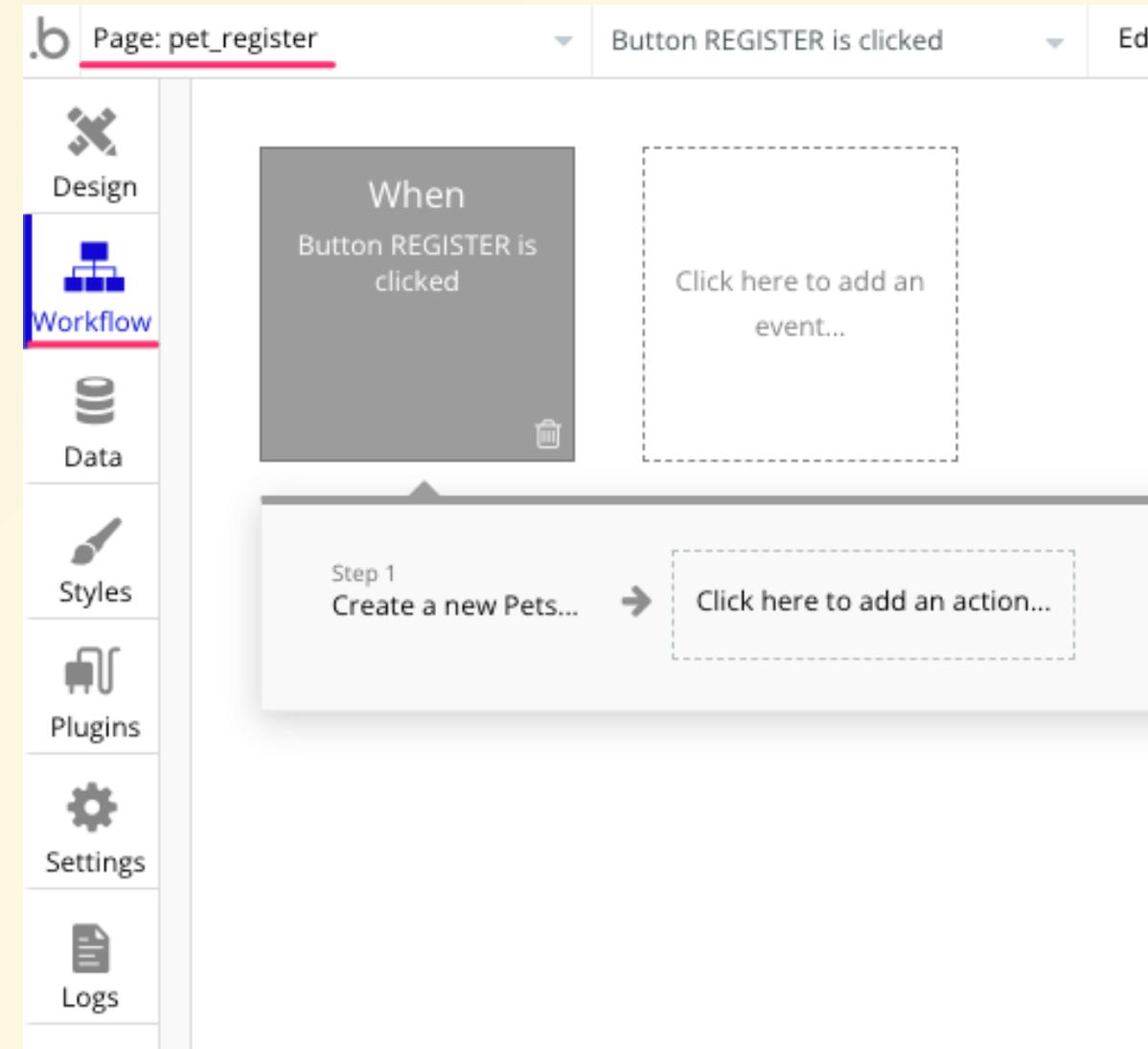


- Since it is difficult to tell which screen is which if the screen is blank, let's place only the text label from the Visual elements and call it "Pets List".
- Now only the frame of the Pets List screen has been created.

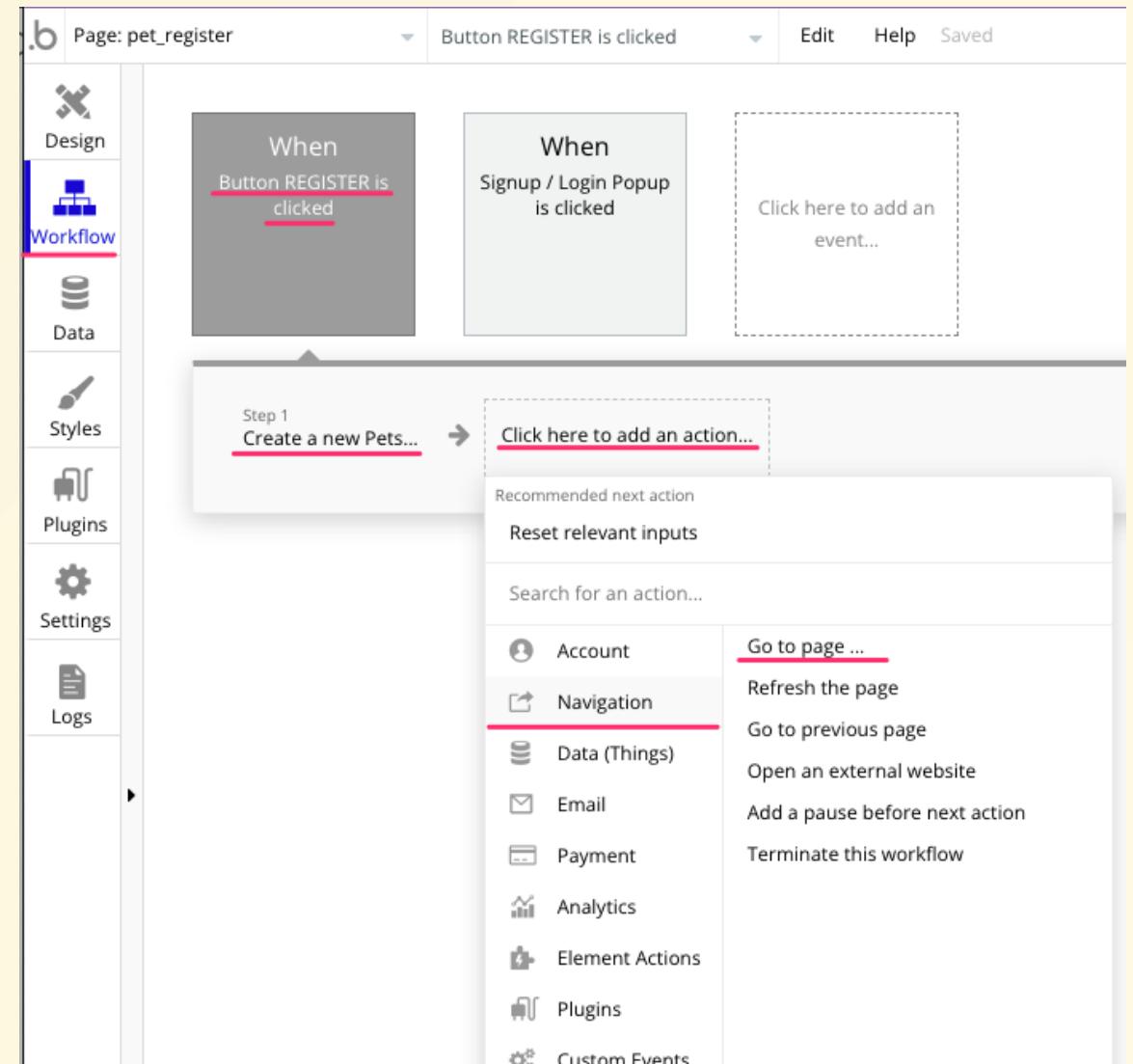


Let's make a screen transition to the pet list screen

- Select the pet_reigster page from the top left
- Click the Workflow tab from the left menu
- You should see the behavior when you click the "REGISTER" button that you just set.

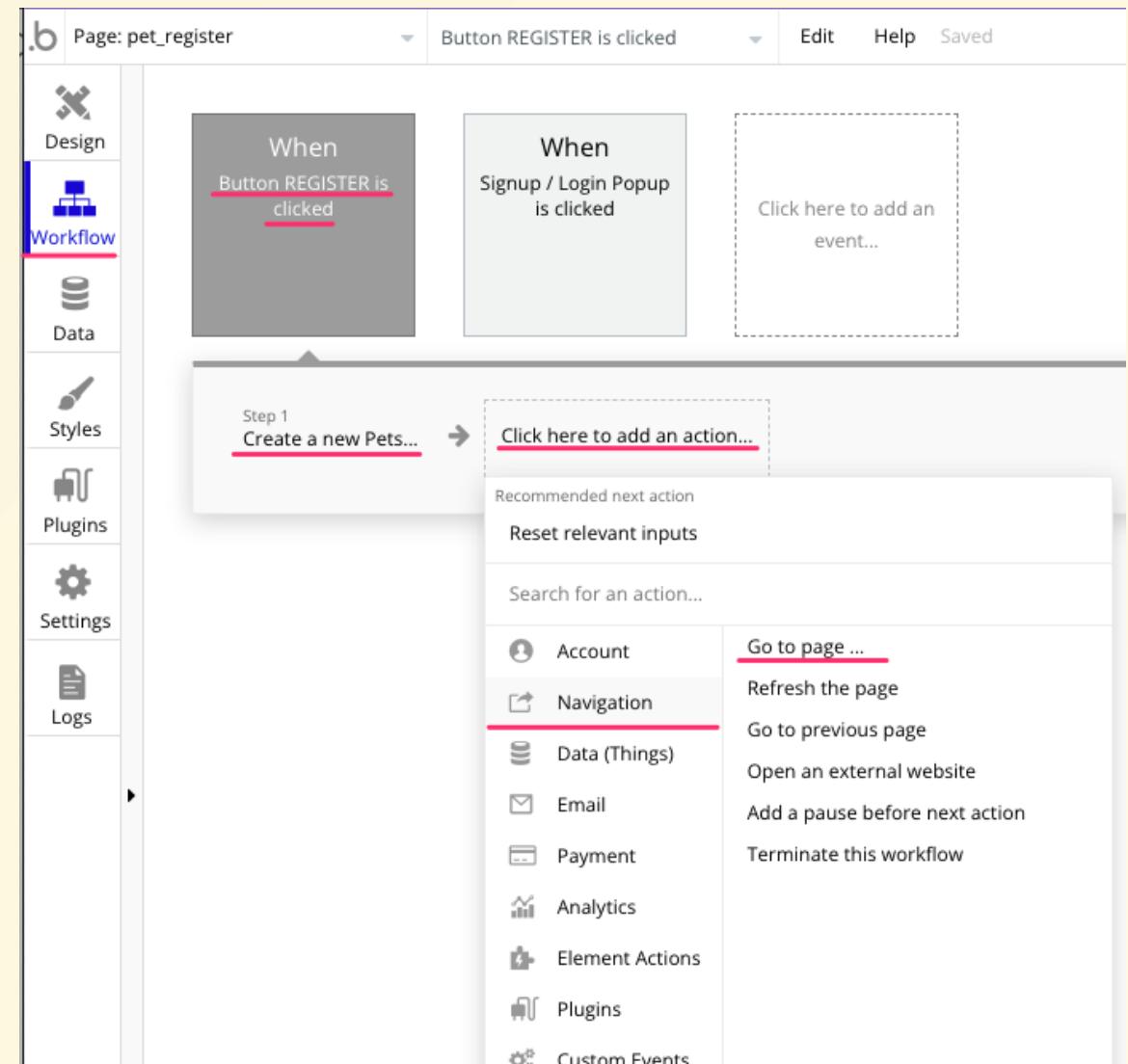


- Here, to the right of "Create a new Pets..." Click here to add an action...
Click here to add an action...

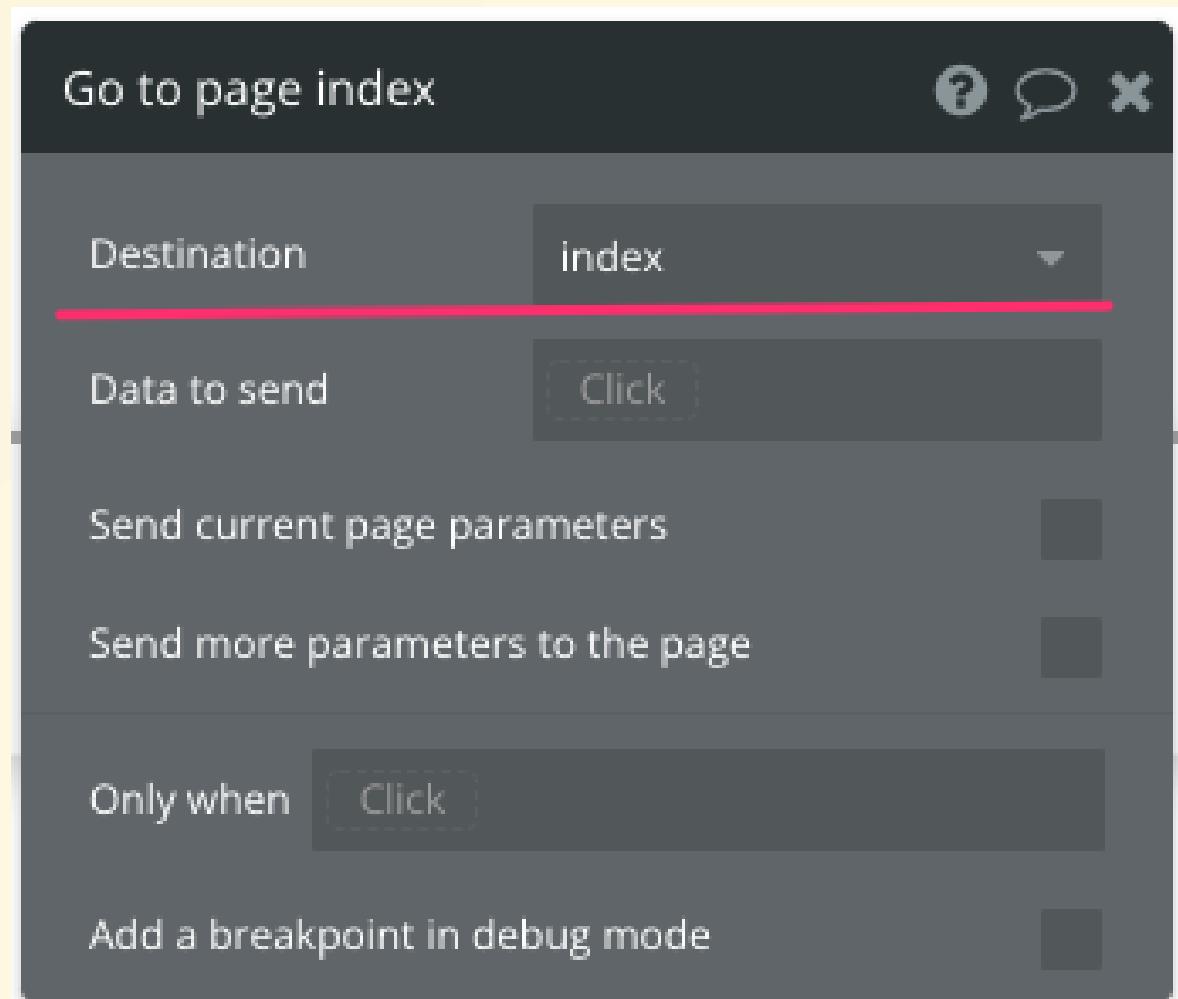


- Now select **Navigation** and as a sub-element

Go to page... as a subelement.



- Then a dialog box will appear, and in the "Destination" field, select the screen to which you want to move.
- In this case, select `pet_list` to move to the pet list screen.



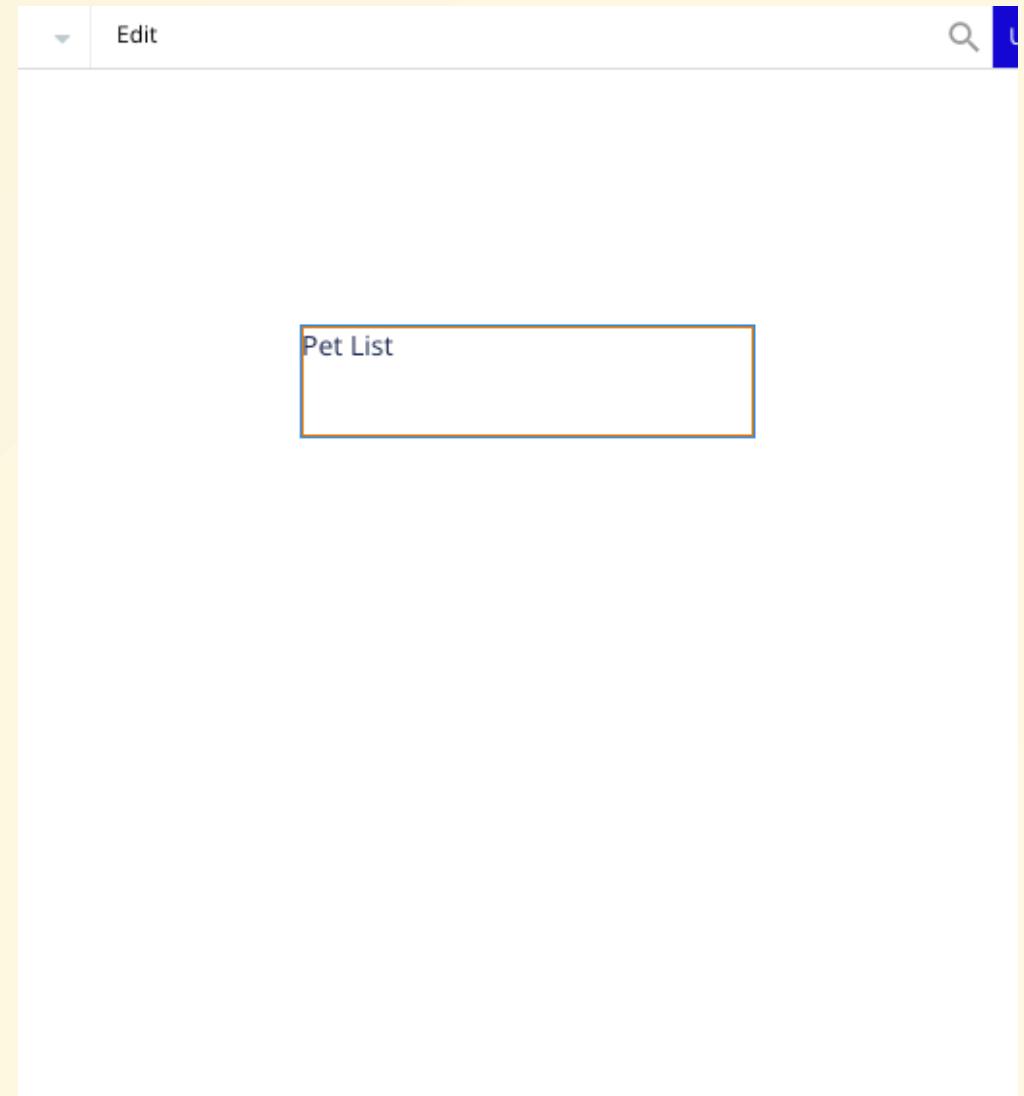
- Now let's preview it.
- Enter the pet information as before and click the "REGISTER" button.
- Then, you will see a transition to the Pet List screen.
- Now you have mastered the screen transition! 

Listing Pets

- Next, let's build the pet list screen!

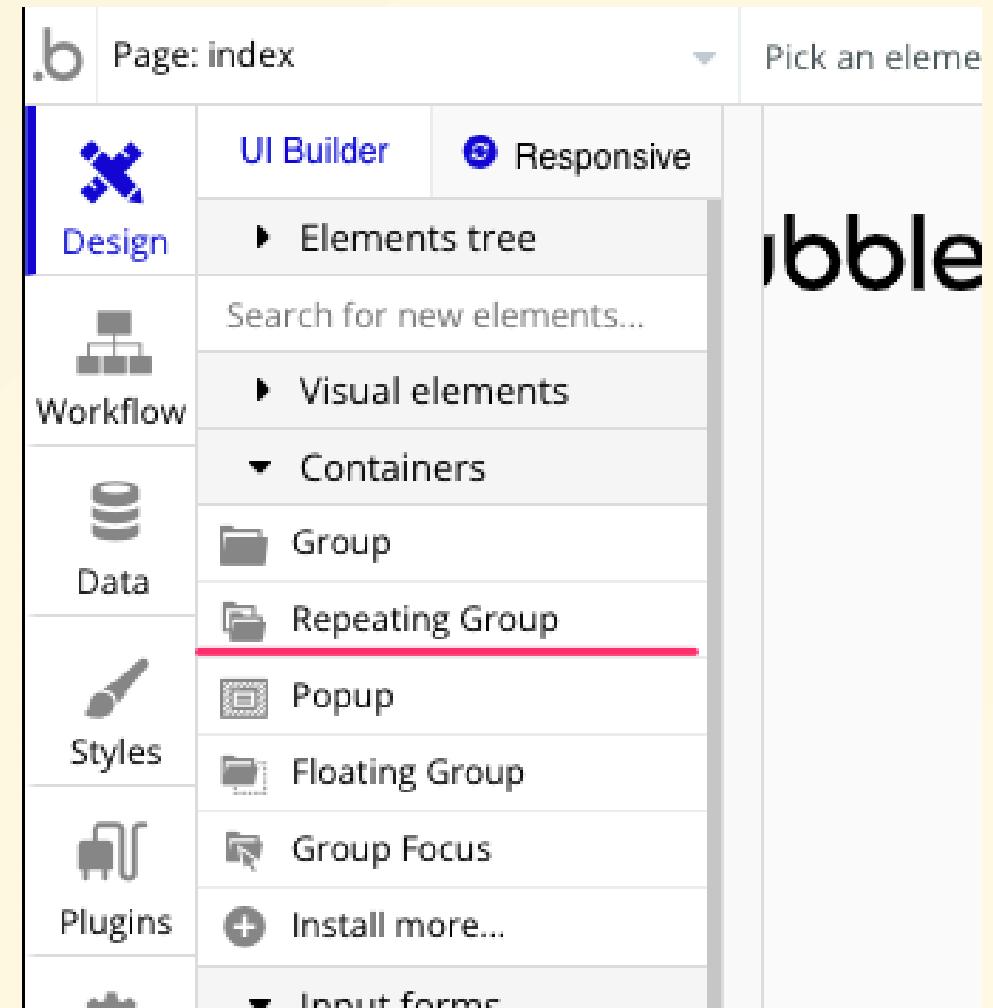


- Select `pet_list` from the list of pages in the upper left corner.
- Delete the text of the pet list you have just prepared.
 - Select the element and press Backspace to OK.

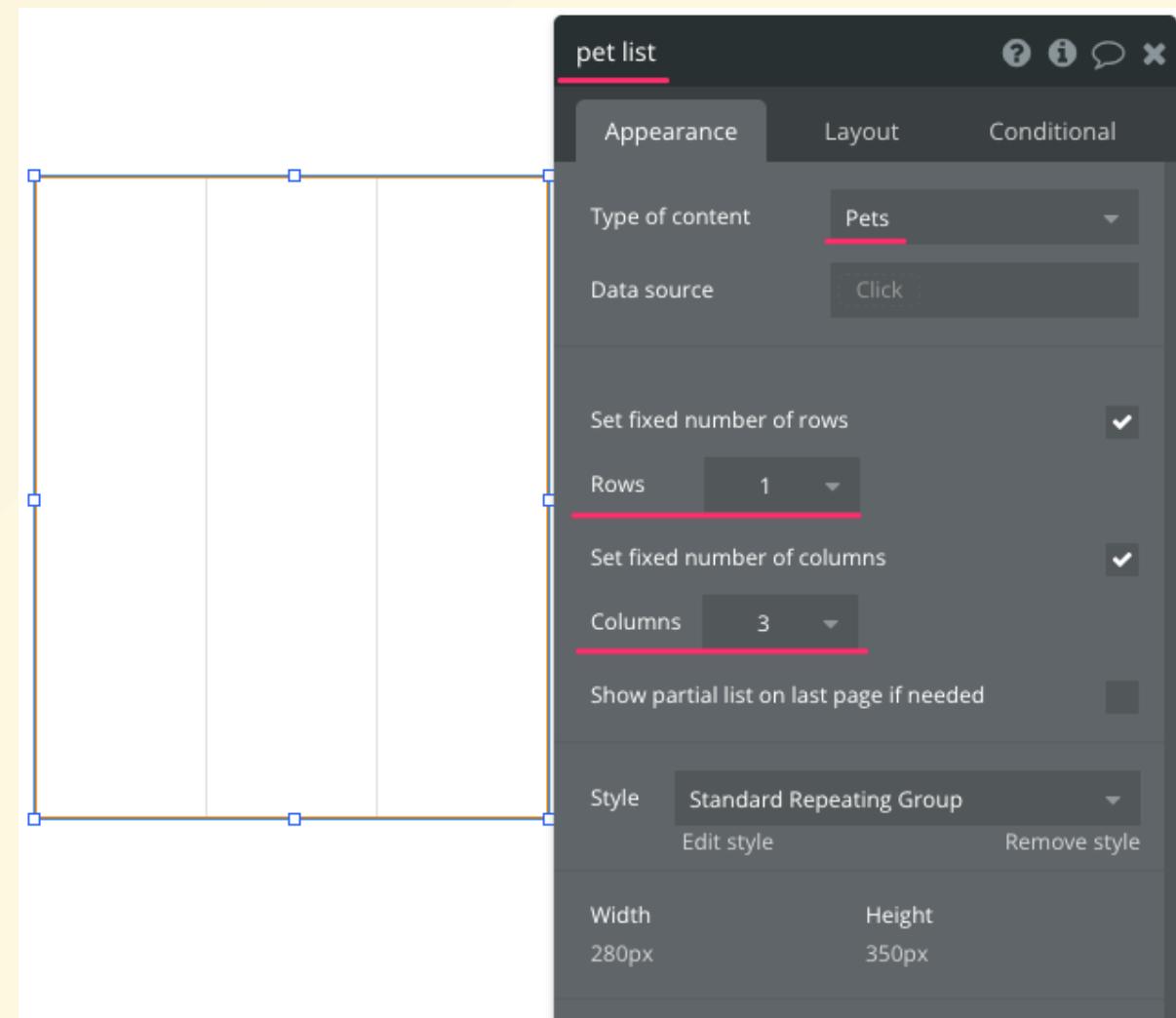


We will create a pet list screen.

- To display the same element repeatedly, as in this case, select **Repeating Group** from **Containers**.
- Drag it to the canvas in the right panel.



- The element detail settings popup will appear, so let's set it up
- Enter **pet list** for the top element name.
- In the **Type of content** field, specify the type of data to be repeated.
 - In this case, it is **Pets**.

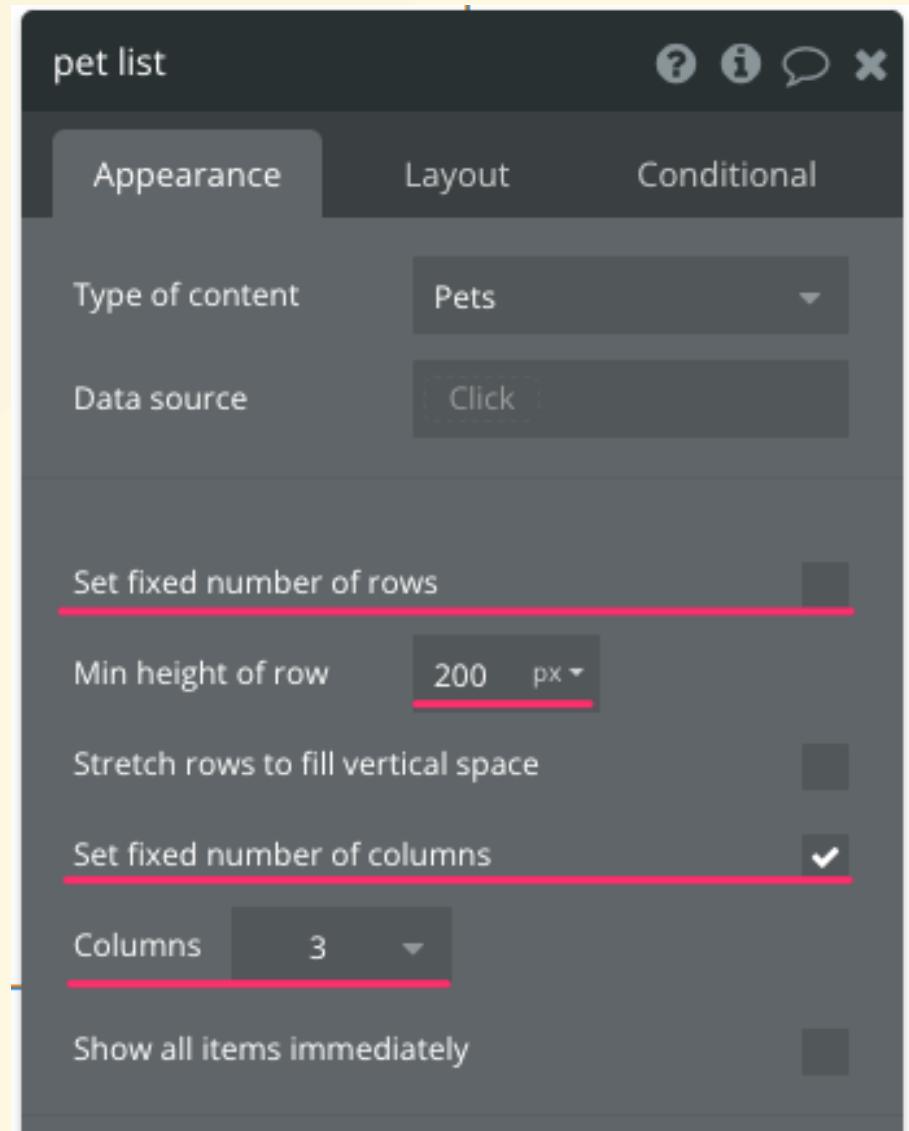


- Specify the number of rows and columns in the list

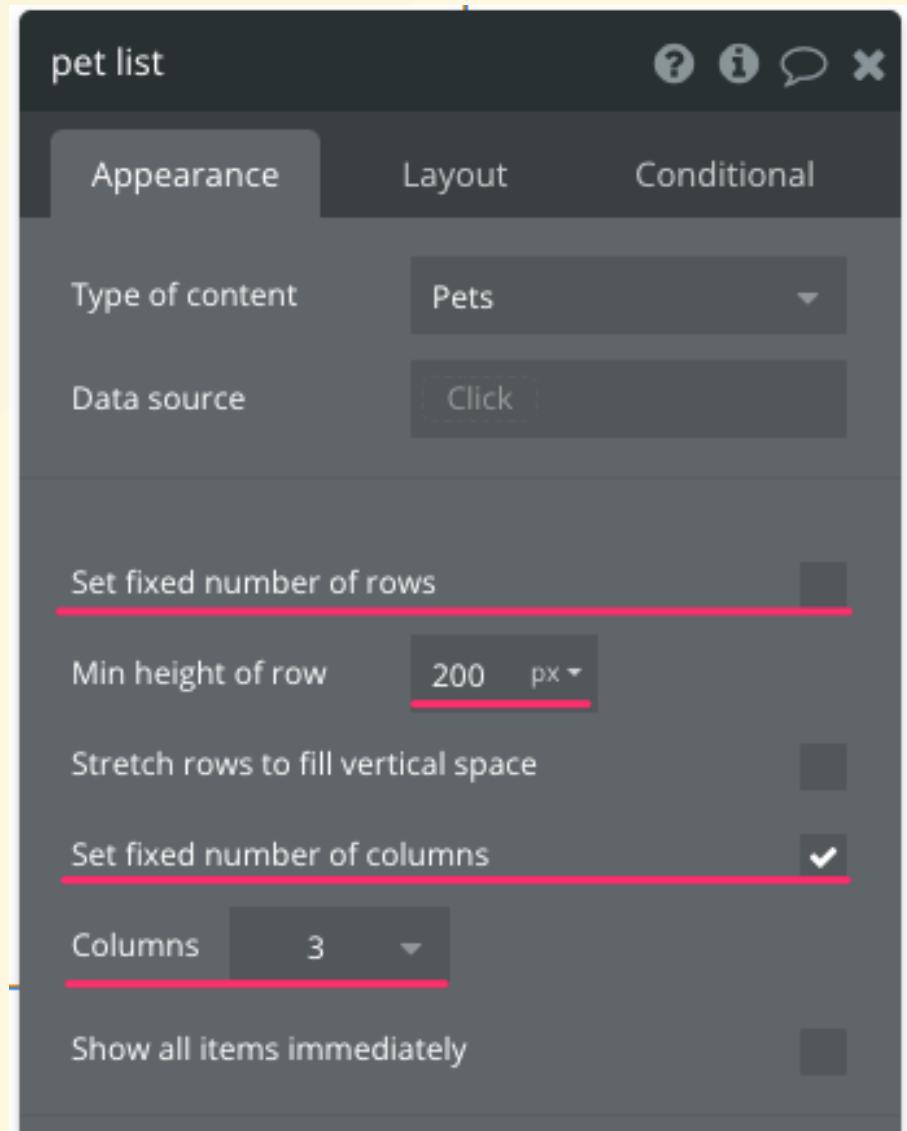
- Uncheck

Set fixed number of rows , since we want the number of rows to change dynamically.

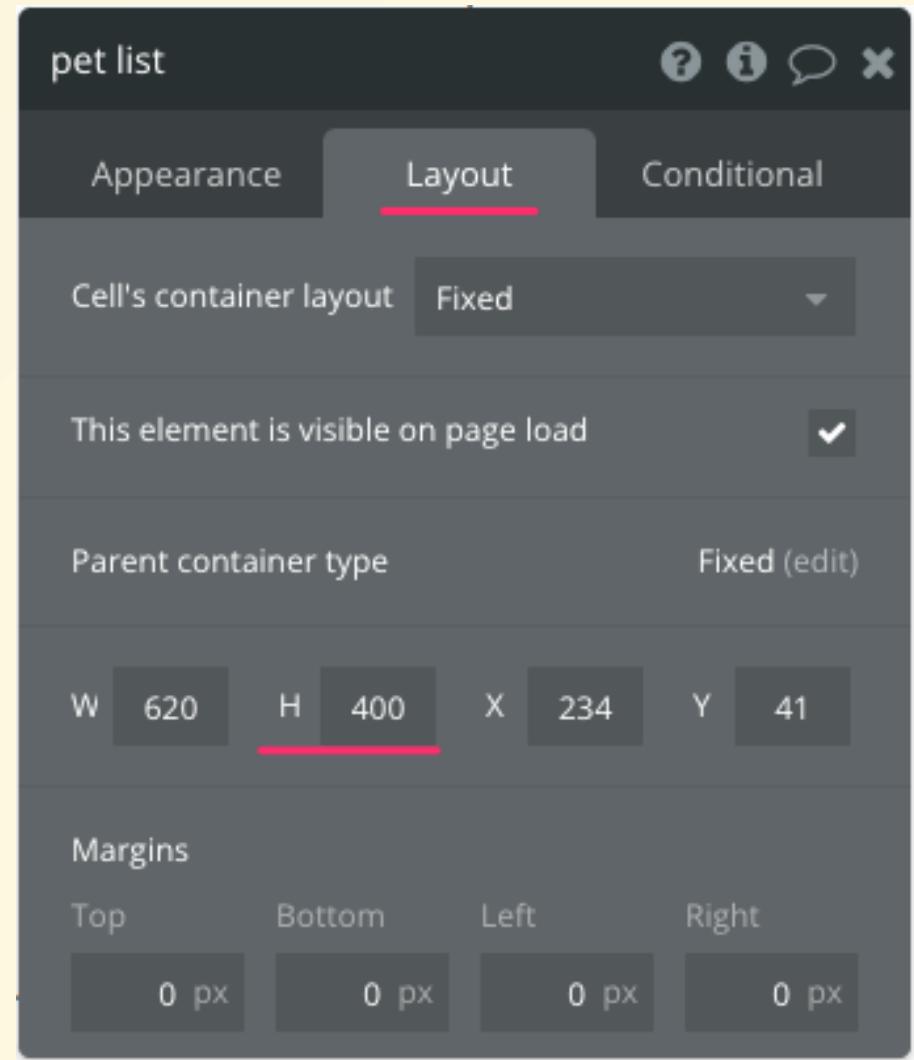
- Instead, specify the minimum height of a row
 - In this case, set it to "200px".



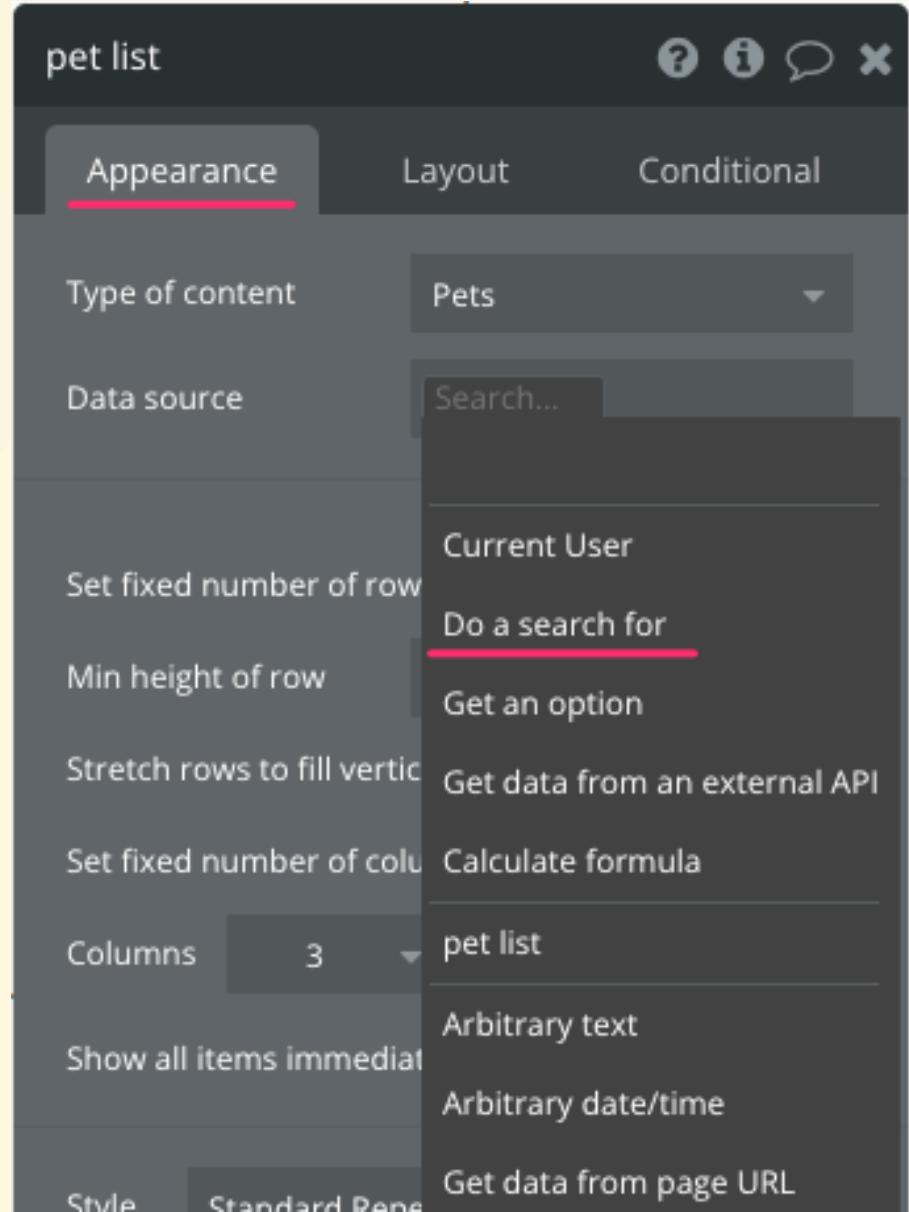
- Leave the Set fixed number of columns checkbox ON since we want to display 3 images per row.
- Set the value of Columns to "3"



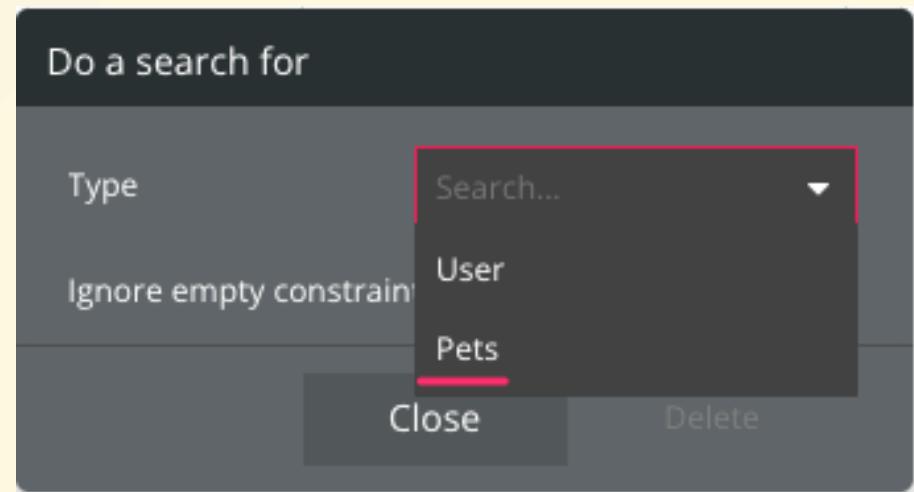
- The height of the entire table should be two rows, so select **Layout** and set the value of **H** to "400".



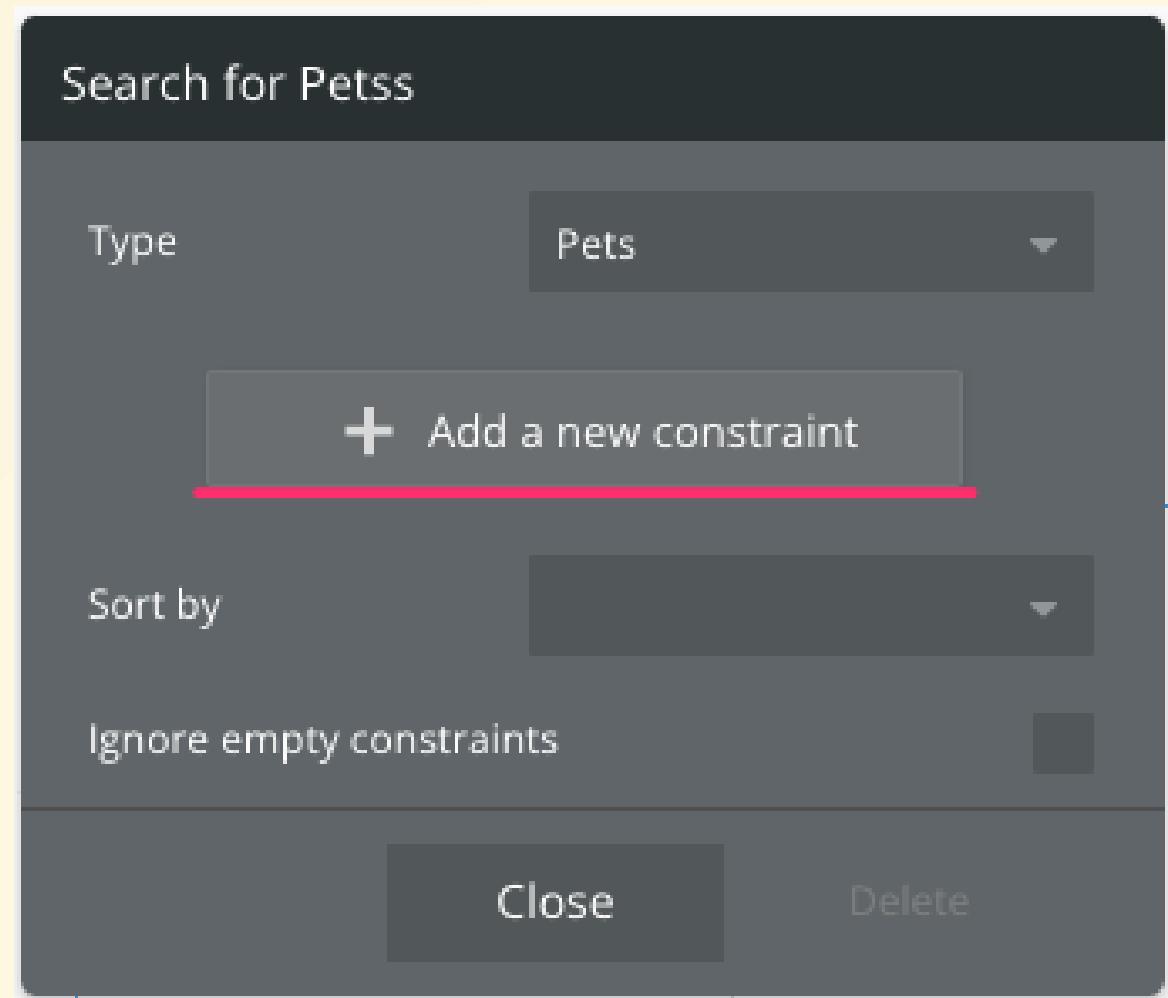
- Next, specify the data to be displayed in the list
- From the **Appearance** tab, click on **Data source**, then
 - Do a search for.
 - This is to specify the data to be listed



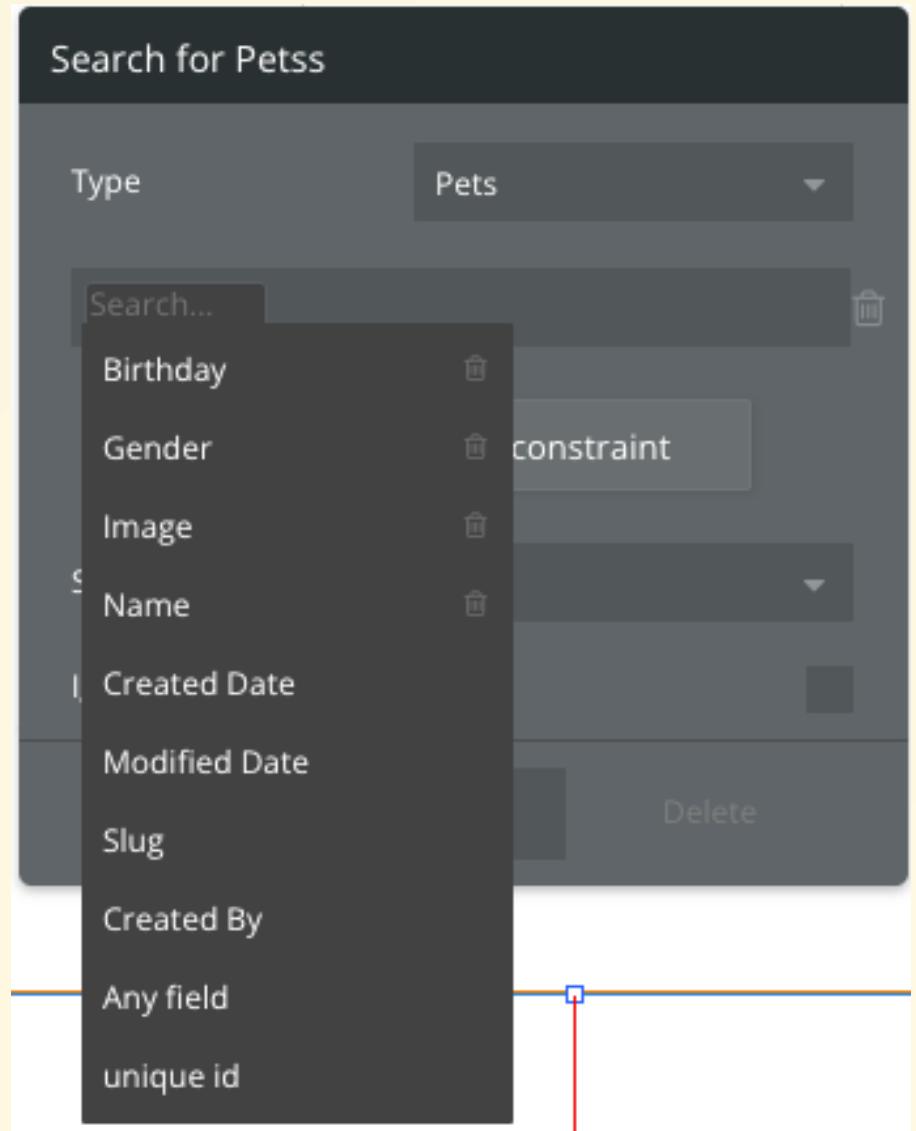
- "Do a search for" popup will appear, specify **Pets** for Type



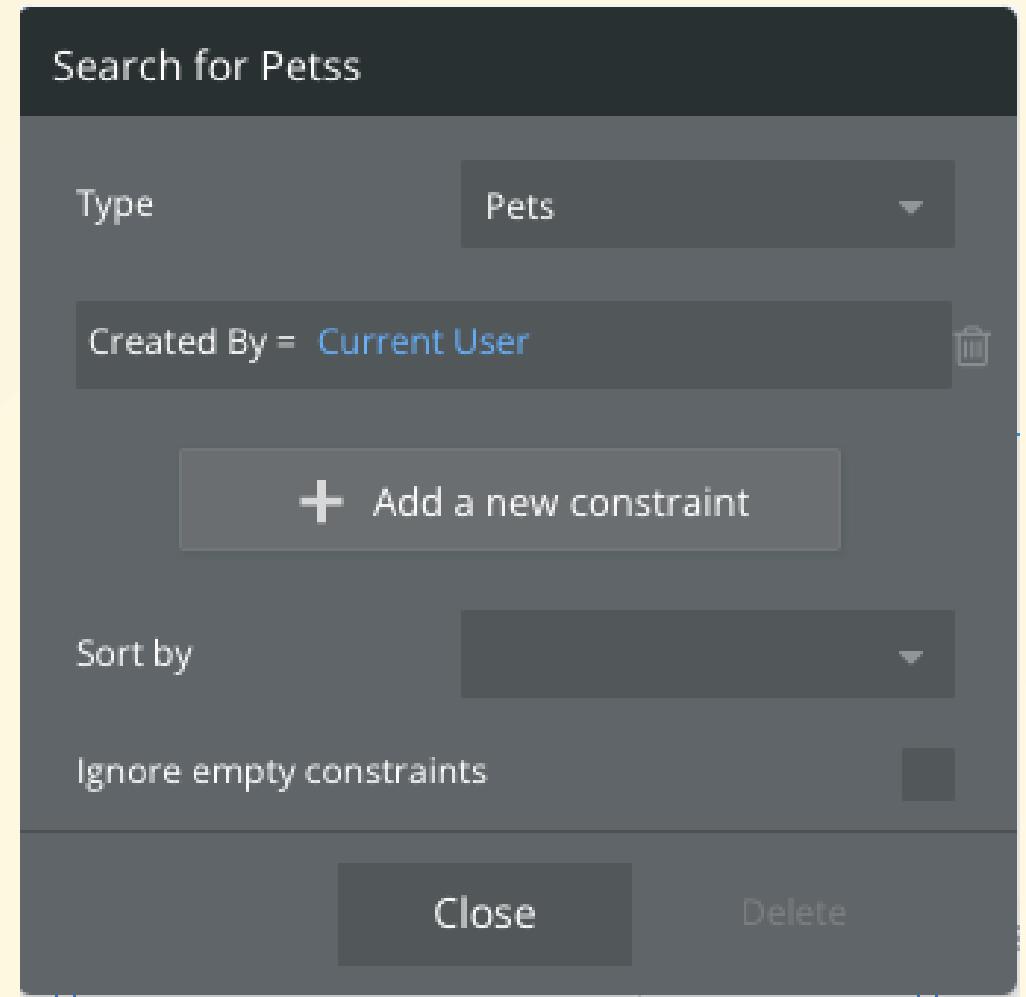
- If this is all, the list of pets can be displayed even if the user is not logged in, so we will add a constraint to display only pets registered by logged in users.
- Click "Add a new constraint" and add more conditions



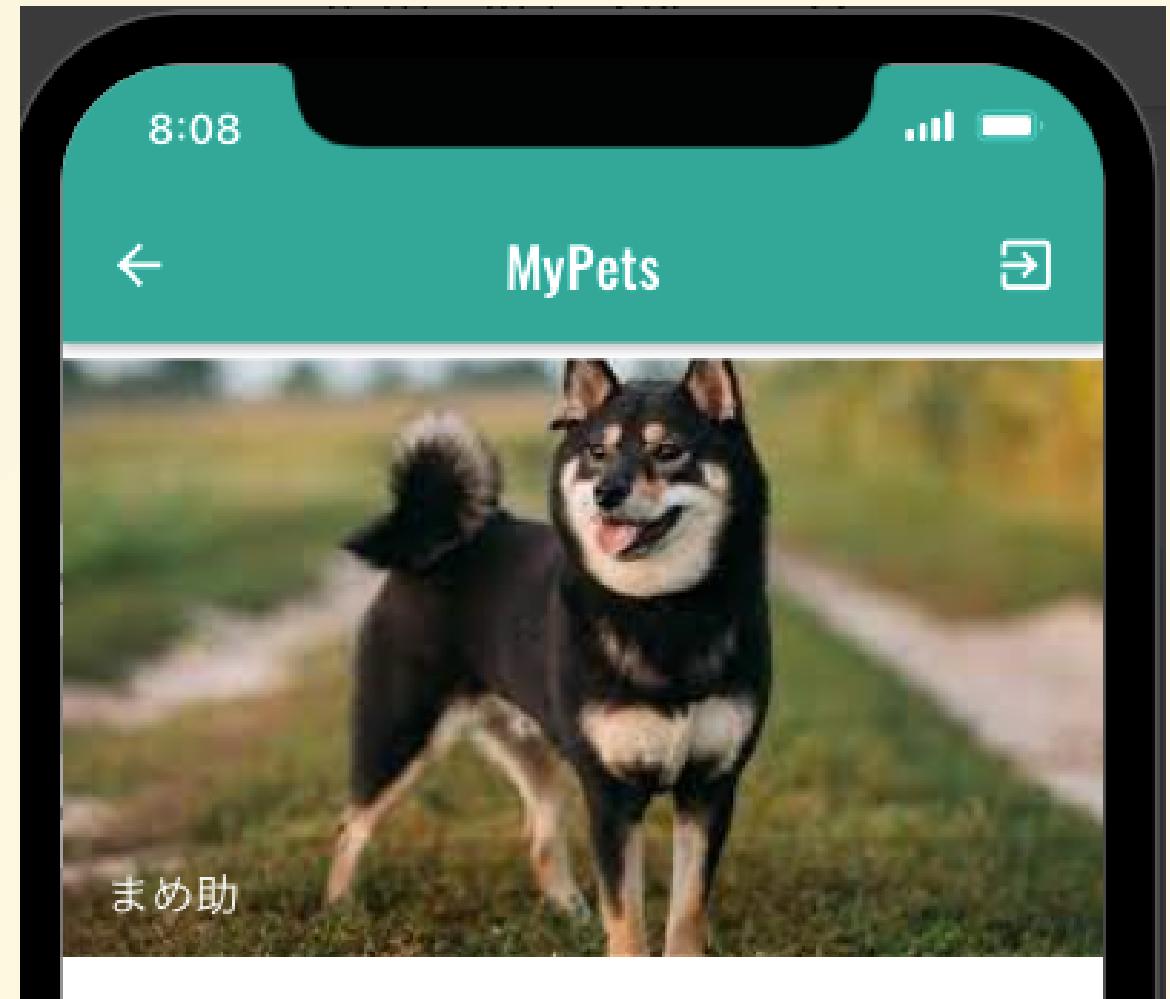
- What are the settings?
- The condition to be set is that the "Created By" of the pet must be the same as the "Current User."
- You can find the answer on the next page.



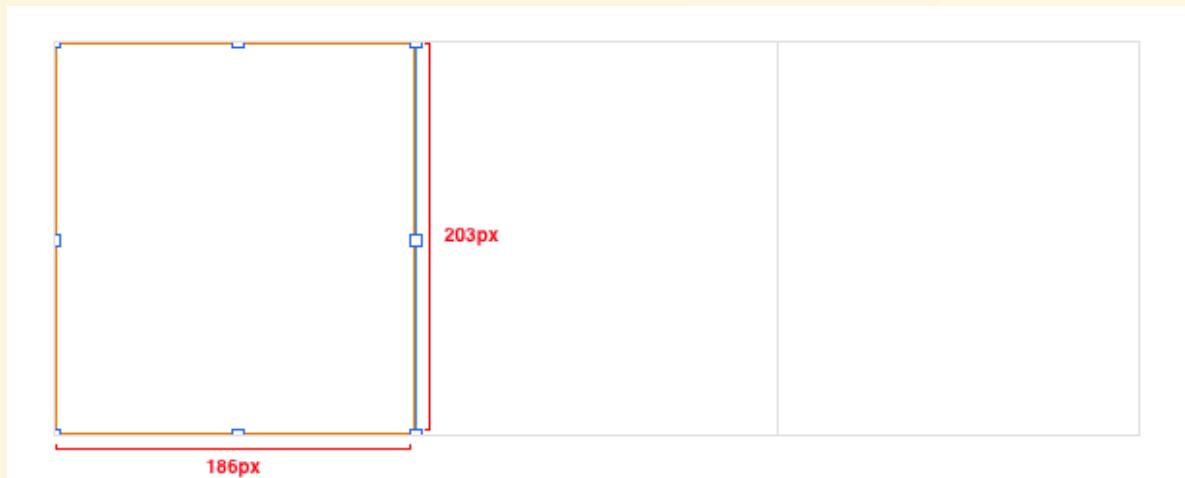
- It looks like this.
- This completes the specification of the data to be displayed in the list.



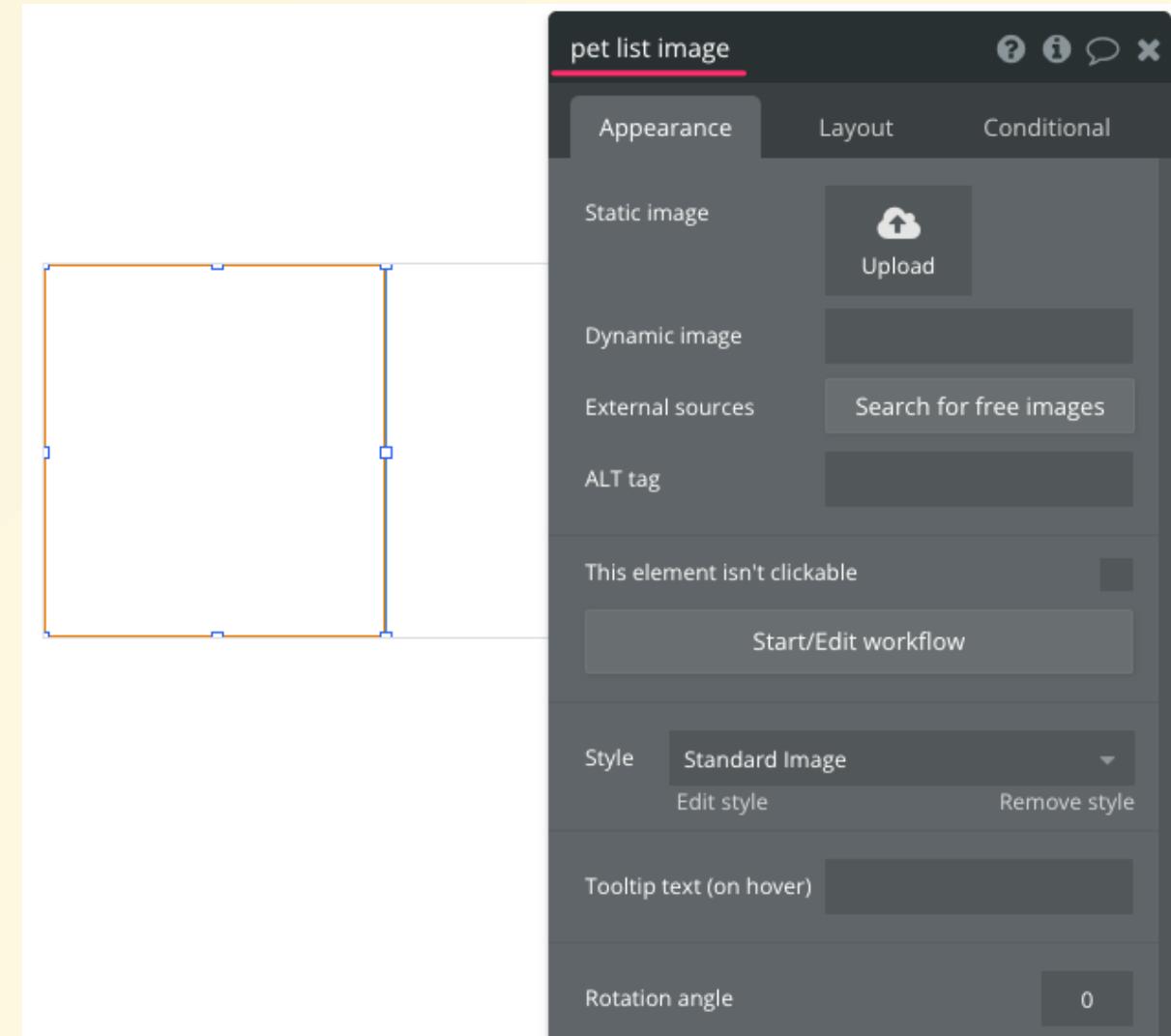
- Next, let's set up the content to be displayed repeatedly
- Let's display the image and name as we did for Adalo



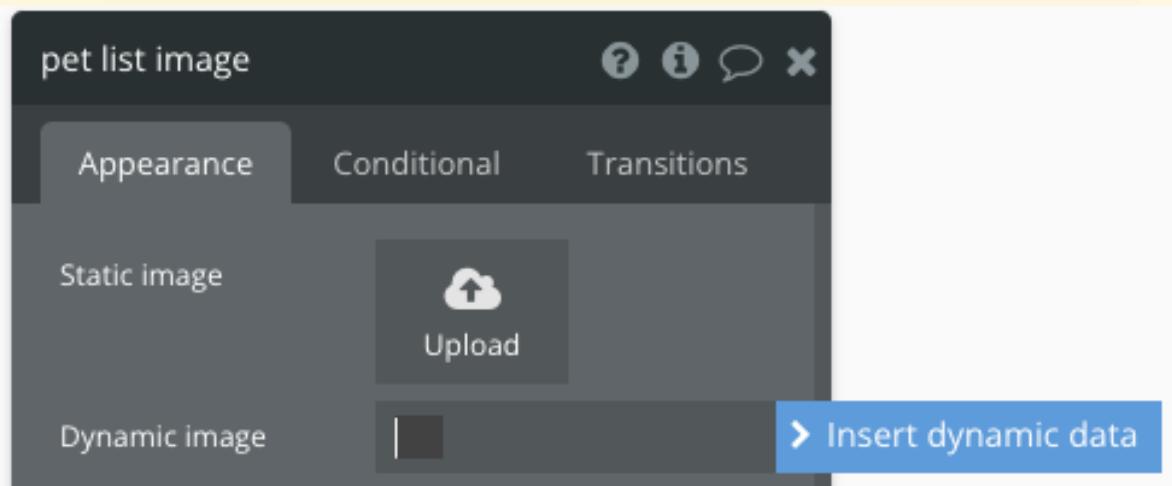
- First, select **Image** from **Visual elements** and drag it to the right panel.
 - If you drag the image into the **Repeating Group**, you will not have to move it later.
 - It is also possible to include it in the Repeating Group afterwards
 - After placing the image, drag it to fill a cell in the Repeating Group.



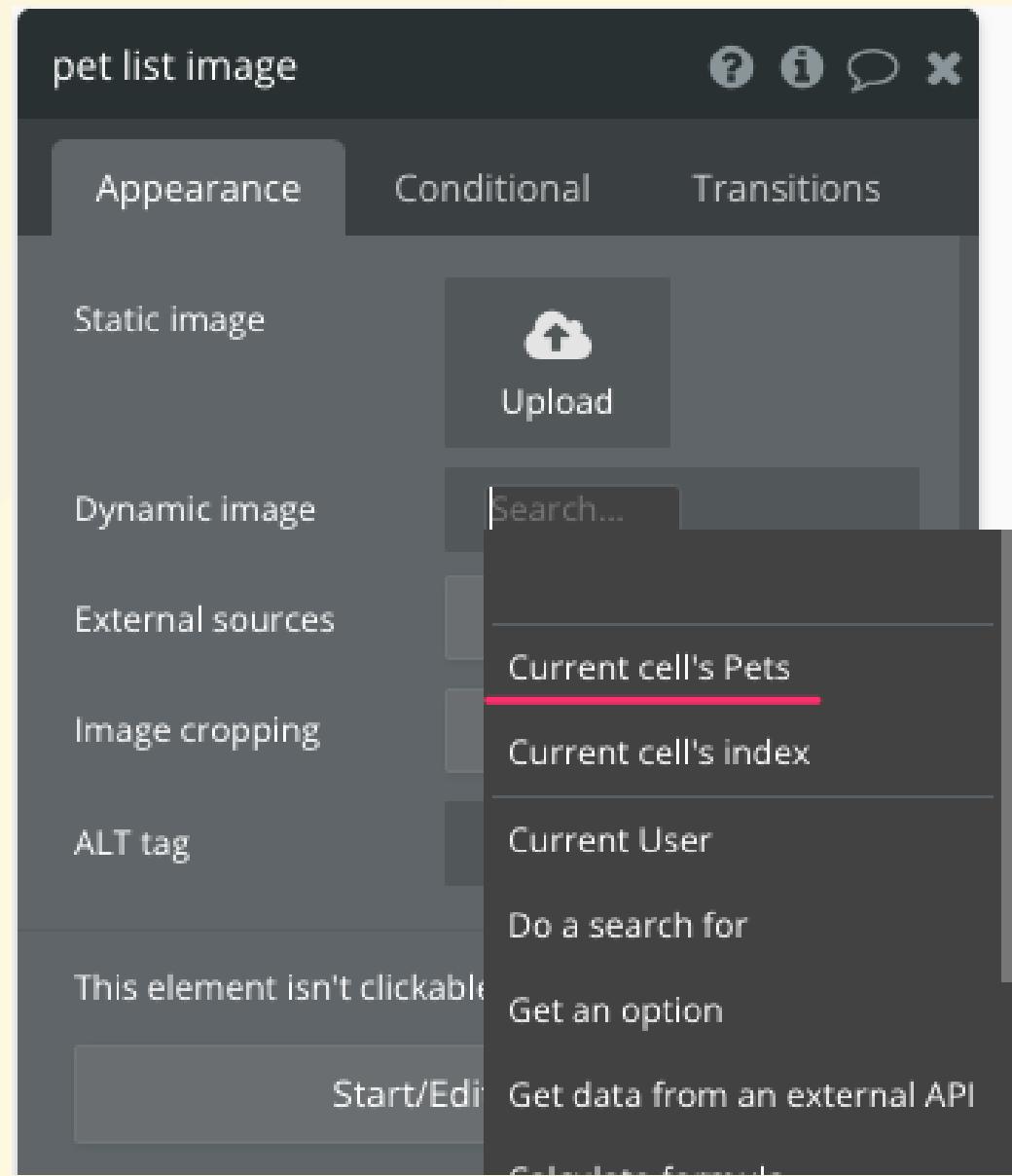
- The element name should be `pet list image`.



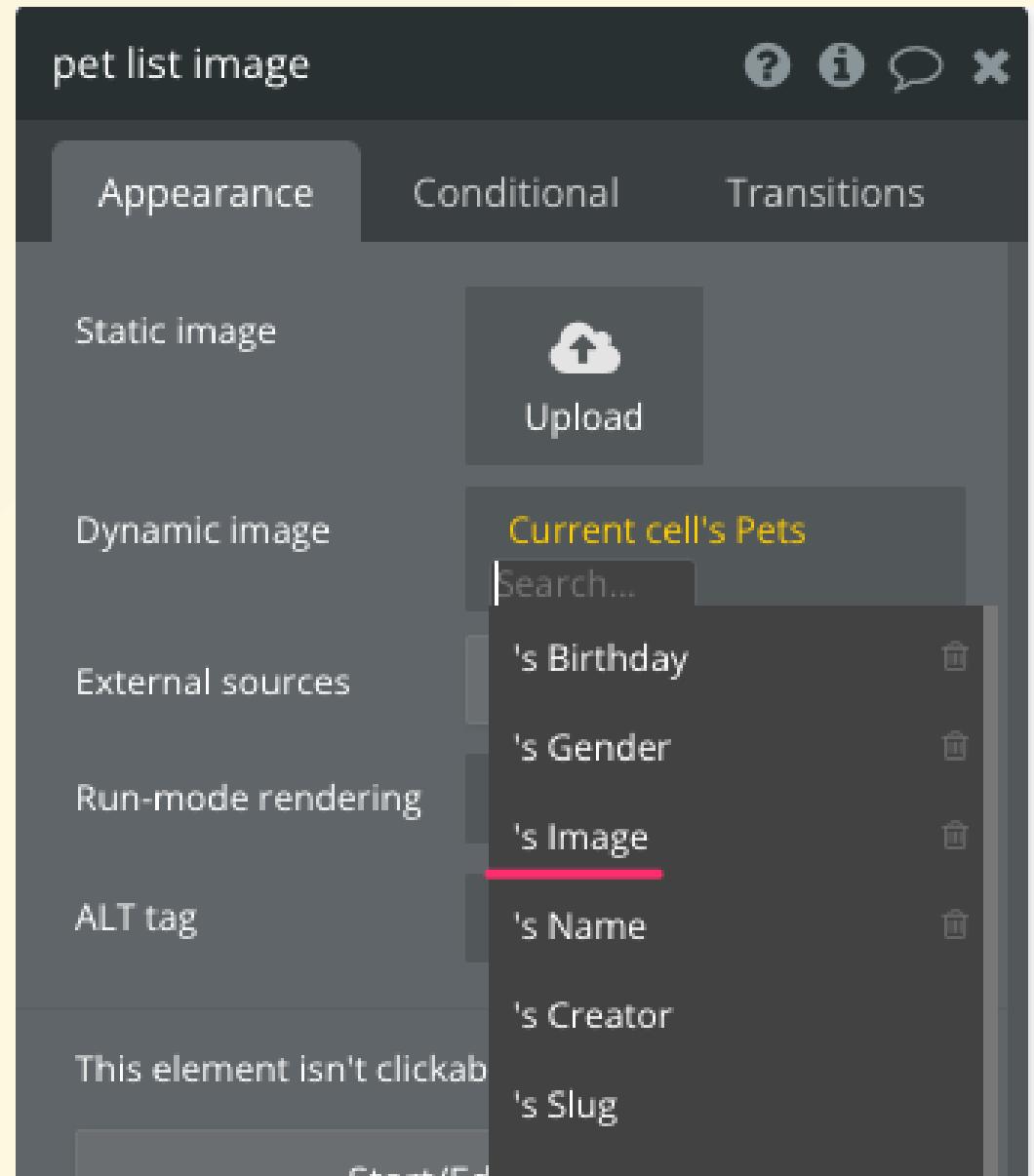
- After placing the element, select "Dynamic image" from the Advanced popup that appears.
- Click on the **Insert dynamic data** button.
 - This function is used when you want to display values or images dynamically, rather than just fixed values or images.



- Click on it and a pull-down will appear, from there click on **Current cell's Pets**.
 - This can be used if you literally want to use the data of the current cell's pets.



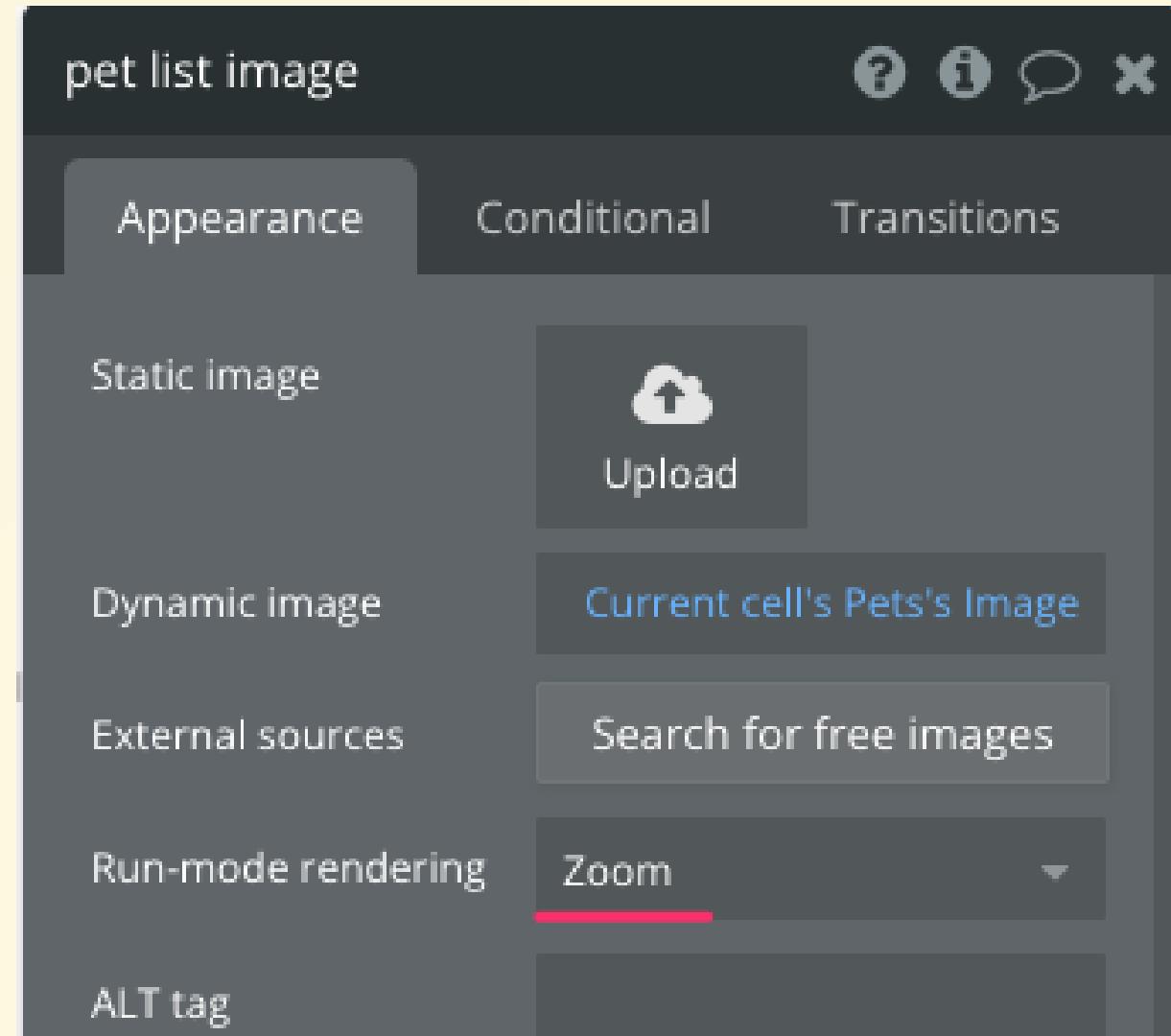
- Select **Current cell's Pets** and a pull-down will appear, displaying the field that the Pets type has, and click ``s Image
 - This will cause the image of the pet in the current cell to be displayed, so that if there are multiple pets, each cell will have an image of the pet.



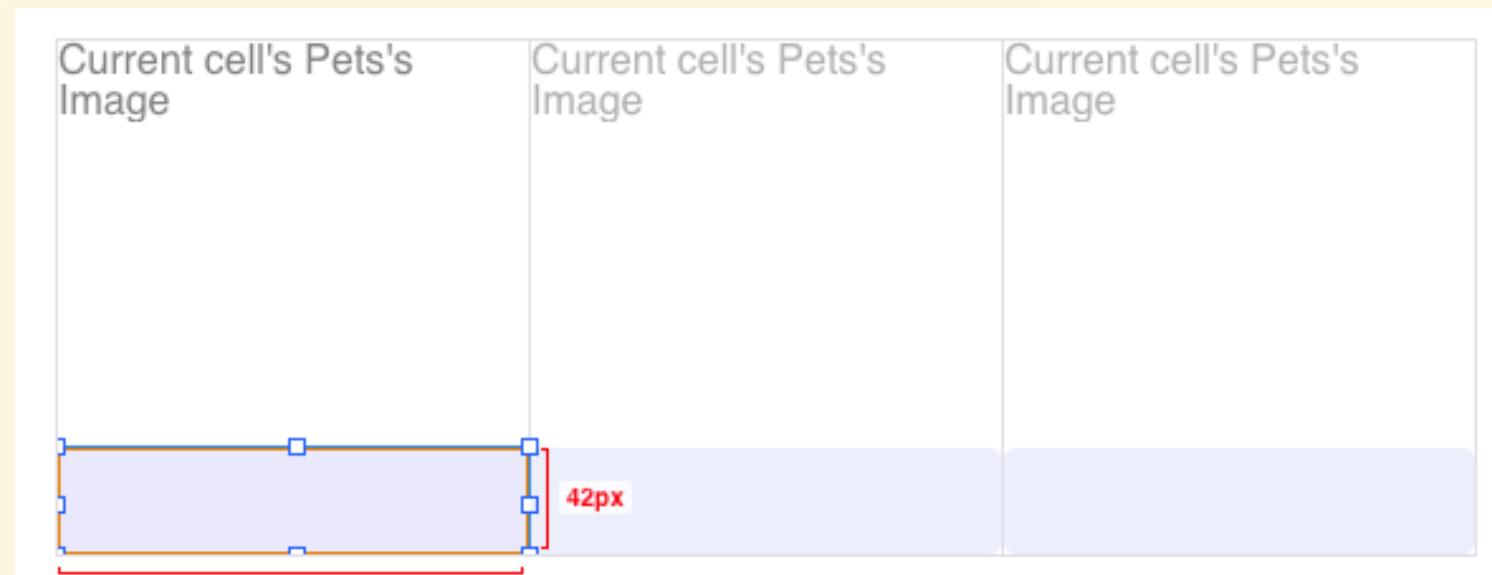
- We have just set the image to fill the entire width of the cell, but we should change the

Run-mode rendering value of the image element from "Stretch" to "Zoom"!

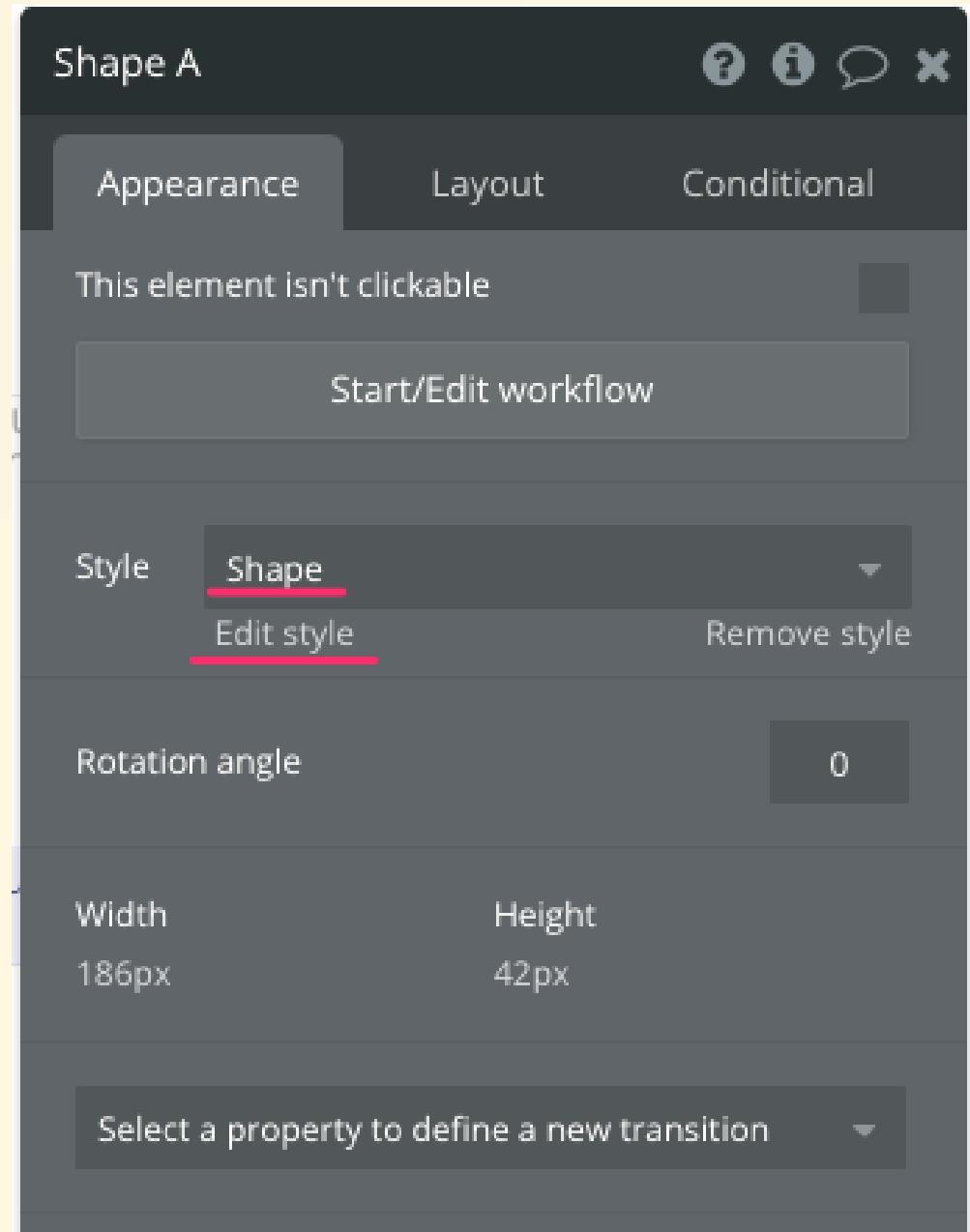
- Otherwise, the image will automatically change its aspect ratio in the cell, so change it to Zoom.



- Next, let's set the background for the name that will appear below the image
- Select **Shape** from **Visual elements** and drag it to the right panel.
 - Drag it into the "Repeating Group" as well.
 - After placing it, move it to the bottom of the Repeating Group and drag it to the full width.



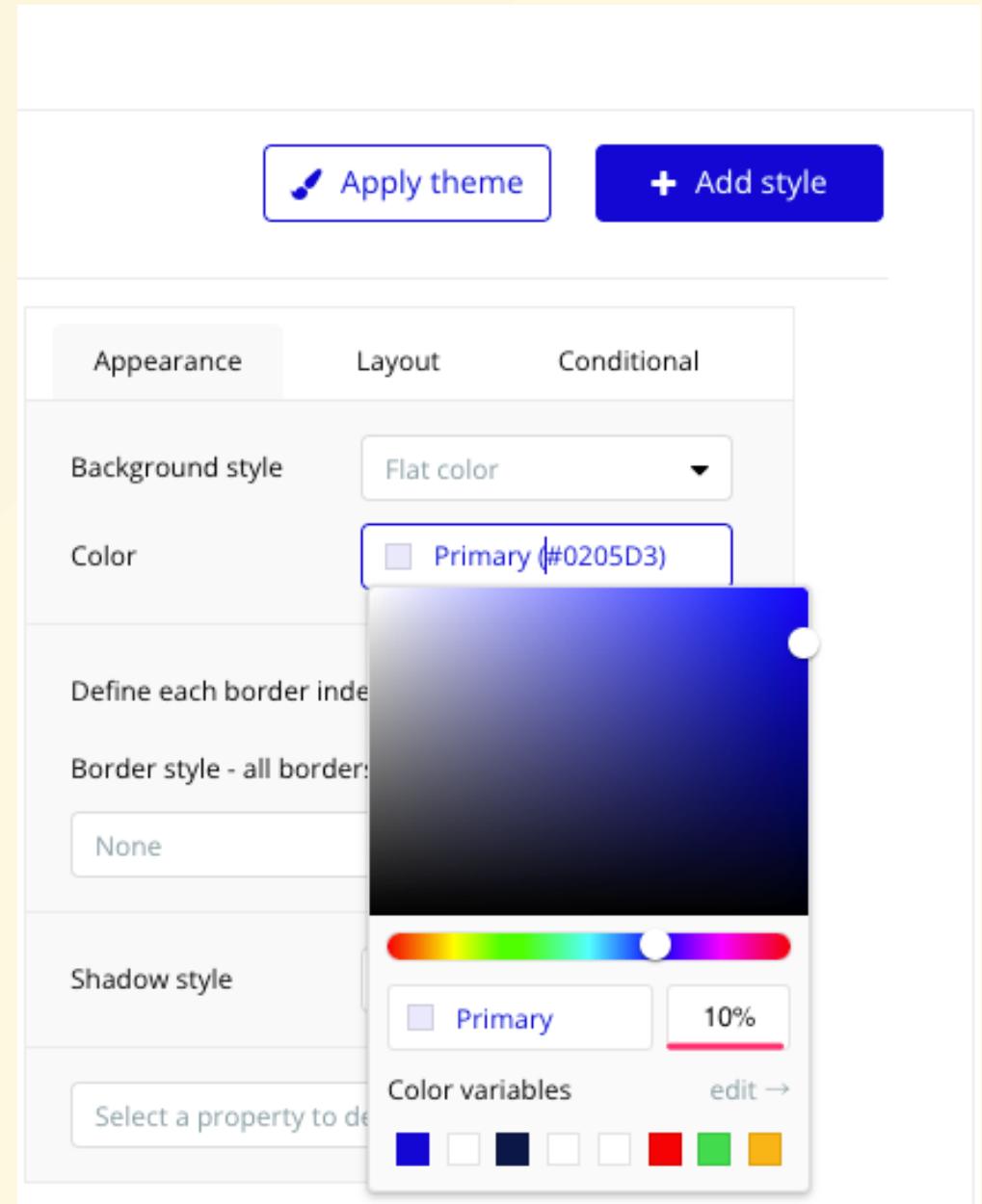
- Let's reduce the transparency of the placed Shape a little
- Double-click on the Shape element to display the usual pop-up window
- Make sure that the Style is set to "Shape", then click on "Edit style" below it



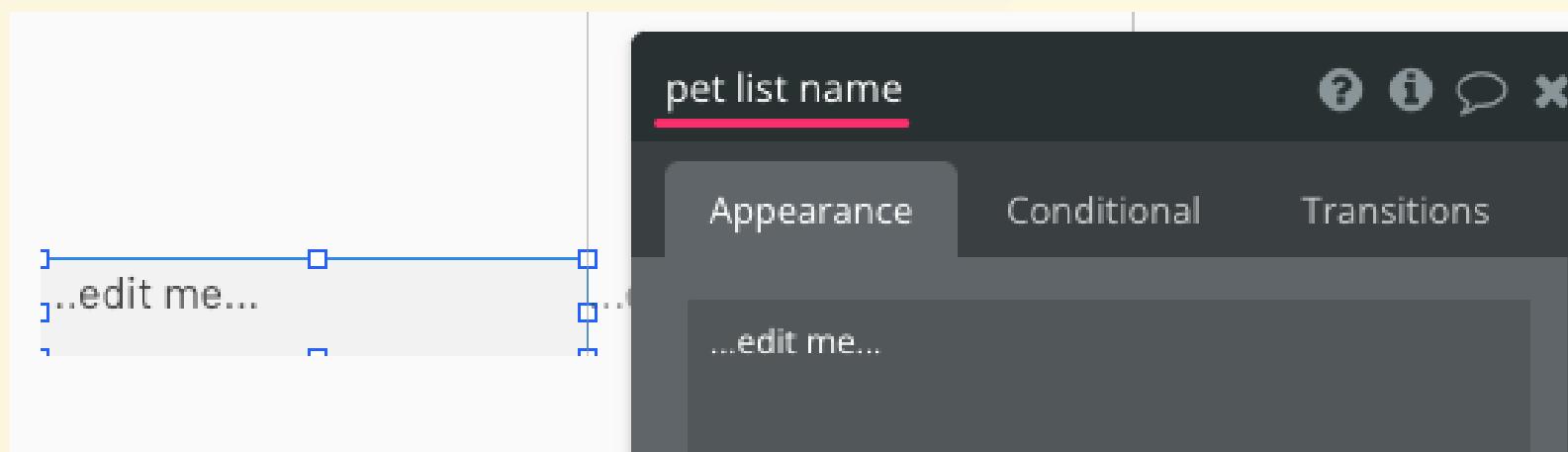
- This will take you to the Styles tab, where you can edit the style of the "Shape" you have just selected

The screenshot shows the Figma Styles tab interface. On the left, a sidebar lists various design elements: Design, Workflow, Data, **Styles**, Plugins, Settings, and Logs. The Styles tab is currently active. The main area is titled 'Element styles' and contains tabs for 'Element styles' and 'Style variables'. A search bar at the top right includes fields for 'Element type' and 'Search by name', along with buttons for 'Apply theme' and '+ Add style'. A list of existing styles is on the left, including 'Alert - Standard Alert (Default)', 'Alert - Warning Alert', 'Button - Flat Button', 'Button - Outline Button', 'Button - Primary Button (Default)', 'Checkbox - Standard Checkbox (Default)', 'Date/Time Picker - Standard Date Picker', and 'Dropdown - Standard Dropdown (Default)'. The 'Shape' style is selected in the center panel, which displays its properties: 'Remove this style as a default for all new Shapes' and a button to 'Find all elements using this style'. To the right, a detailed editor panel shows the 'Appearance' tab selected, featuring a color palette with 'Primary (#0205D3)' highlighted. Other tabs include 'Layout' and 'Conditional'. Below the color palette, there are sections for 'Border style - all borders', 'Shadow style', and 'Color variables'.

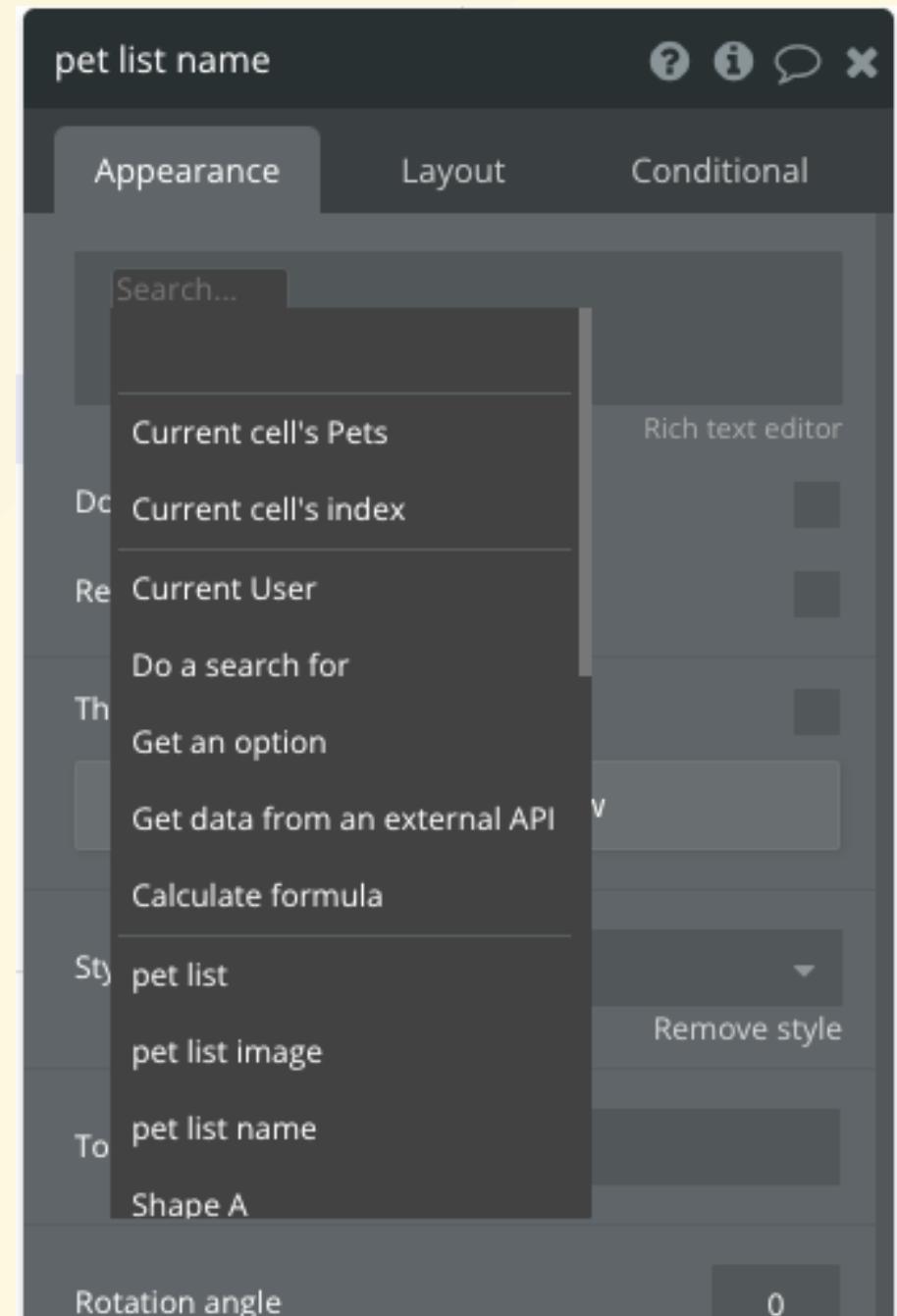
- Selecting the Color option will bring up a pop-up window that allows you to change the color and transparency.
 - Change the number to the right of "Primary" from 30% to 10%. This number is the transparency rate, with 100 being non-transparent and 0 being transparent.



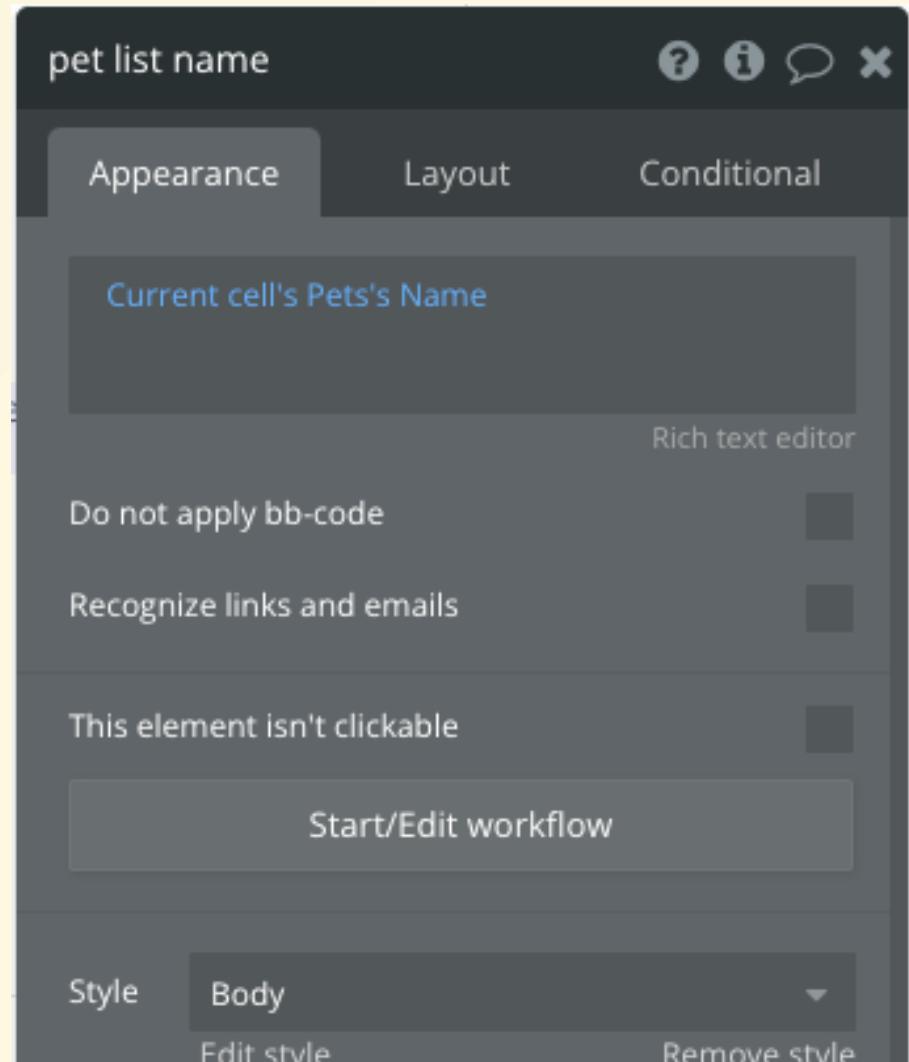
- Finally, we display the pet's name
- Select **Text** from **Visual elements** and drag it over the shape you just placed.
 - Drag it into the Repeating Group as well.
 - Drag the shape to the same size as the shape.
- Name the element **pet list name**.



- Use Dynamic data for pet names as well as pet images
- "...edit me..." and you will see **Insert dynamic data** as you did with the image, so let's set the "pet's name in the current cell" from there.
- You will find the answer on the next page.

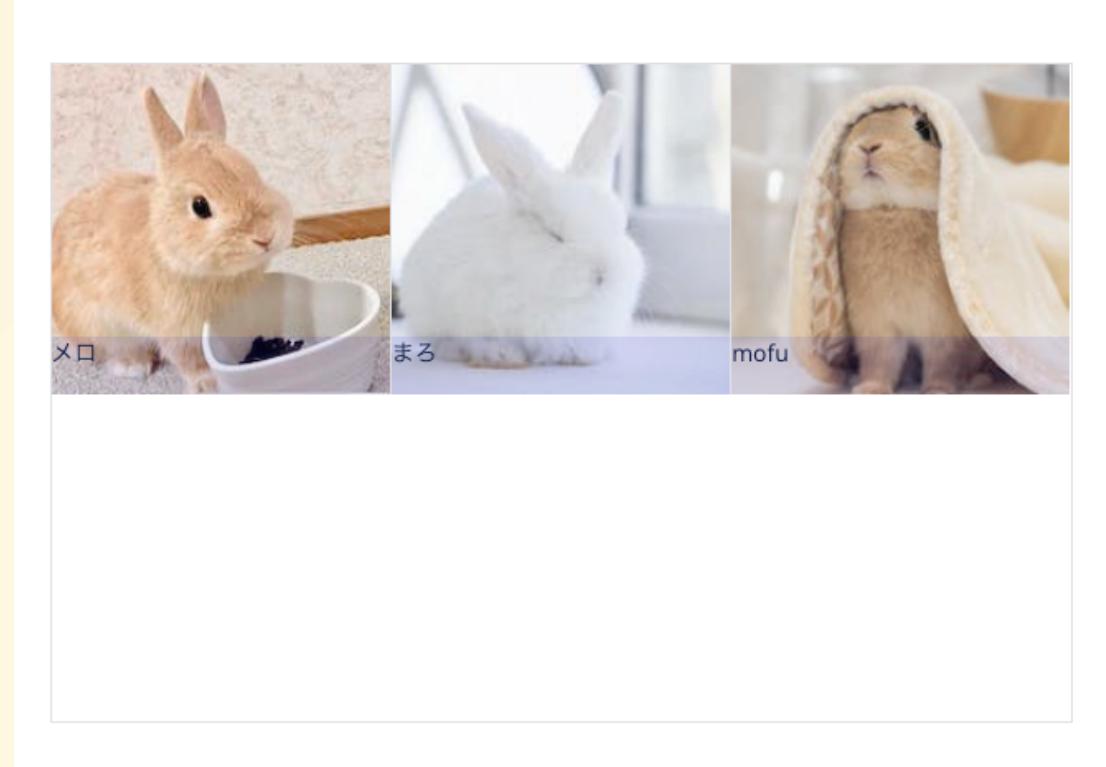


- Current cell's Pets --> 's Name select
- This completes the configuration of the content to be displayed in the listing.



Now that you're done, let's preview it!

- Do you see the pet's image and pet's name as a list of pets?



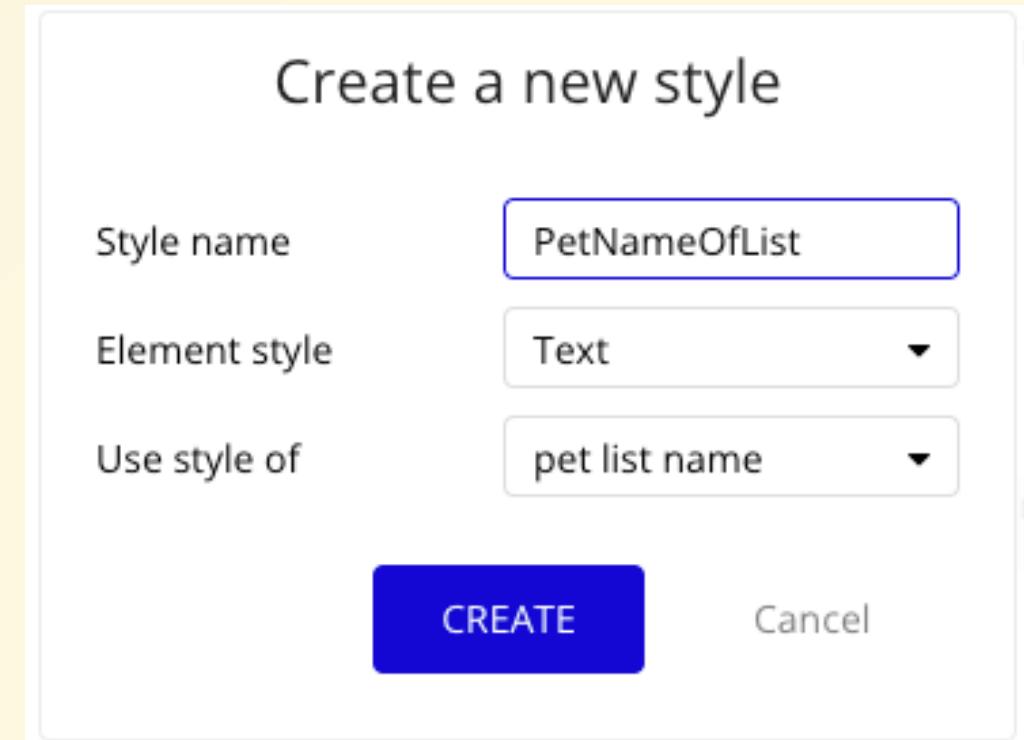
Exercise 1

- The pet's name is aligned in the upper left corner, so let's center it and make the letters a little larger



- Hint 

- Create a new "Style" and set it to "pet list name"



Let's prepare a lead from the pet list to the registration screen.

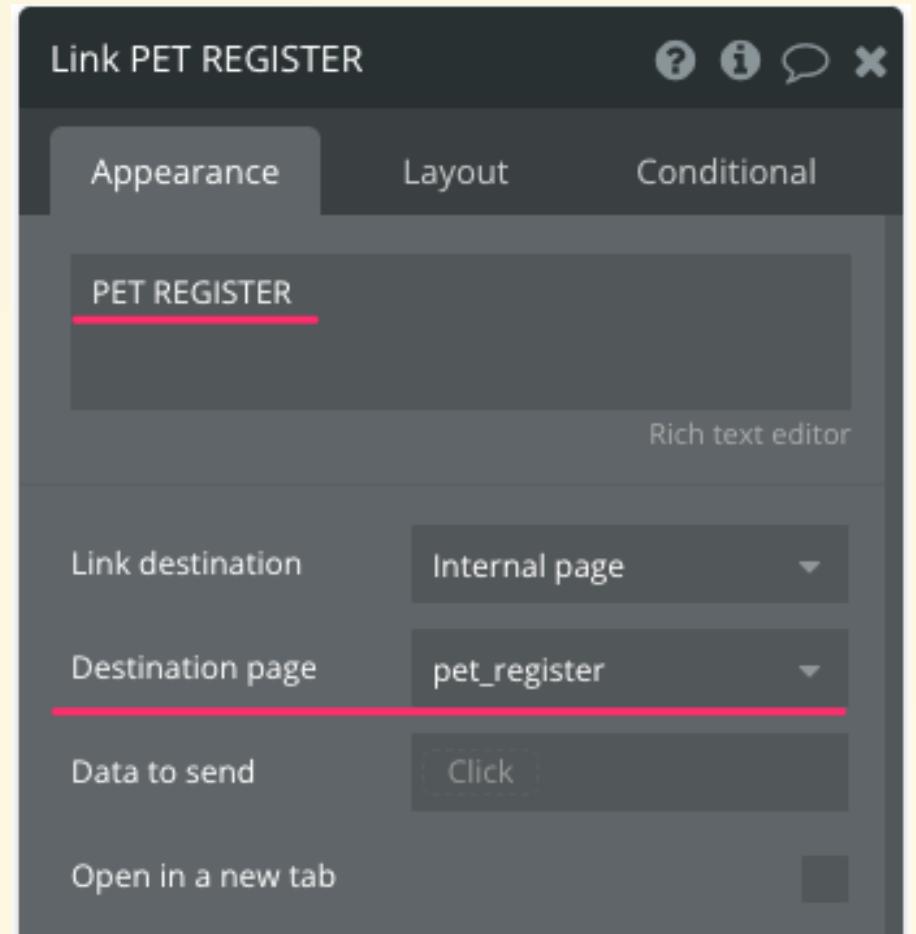
- Now that we have a list of pets, let's prepare a lead from the list screen to the registration screen.
- Let's prepare this line at the top of the Pets List screen.



- Click on the menu in the upper left corner and select `pet_list`.
- Then select `Link` from `Visual elements` and drag it to the top of the screen.



- Name the link "PET REGISTER".
- Select the **Destination page** as the destination for the pet registration page.



Now let's preview it!

- Have you clicked on the "PET REGISTER" link at the top of the pet list to get to the pet registration page?



Let's create a Pette Details screen.

- Next, let's create the Pet Details screen.
- The key point here is the information transfer part for the pets selected in the list screen.

Name

メロ

Image



Birthday

Sep 5, 2022 12:00 am

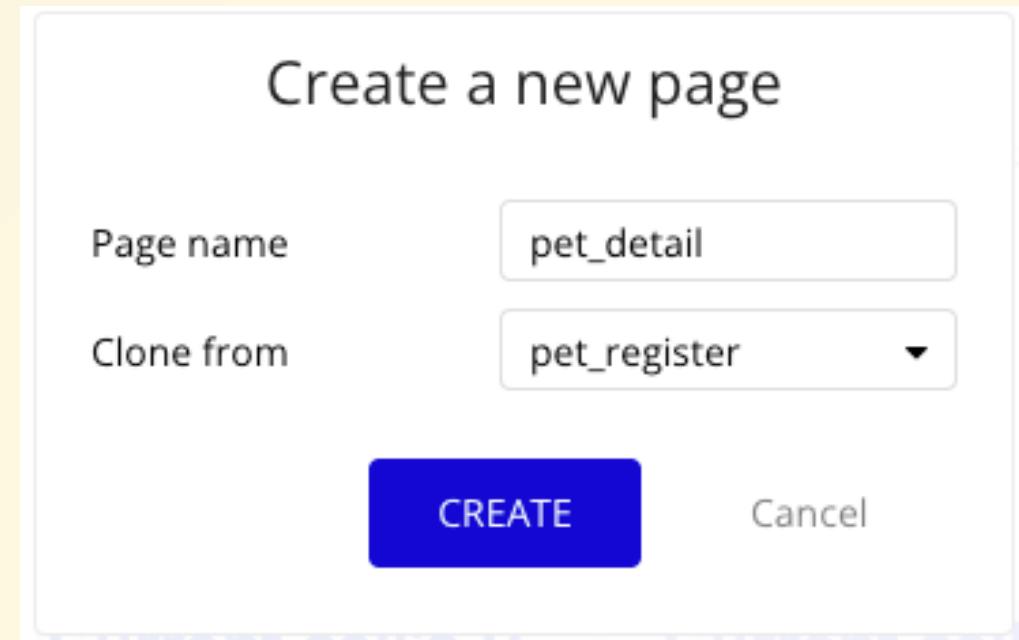
Gender

Male

REGISTER

First, prepare a new detail page

- Add a new page..." from the upper left from the top left corner.
- "Page name" should be `pet_detail`.
- For "Clone from", choose `pet_register` since the screen structure is similar.



- If you have cloned (copied) it, it will be the same as the registration screen, so we will review it.

Before

Name

Image

Click to upload pet image

Birthday

Gender

REGISTER

- First, delete all input elements prepared as input items

After

Name

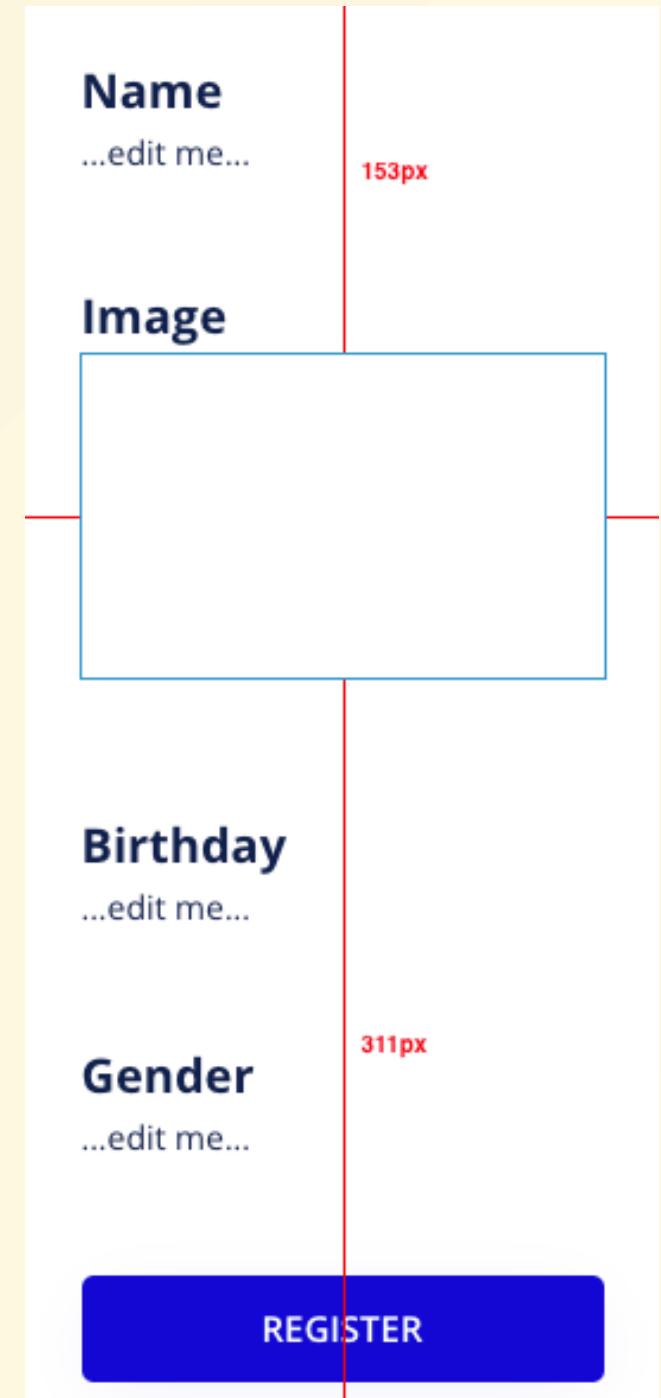
Image

Birthday

Gender

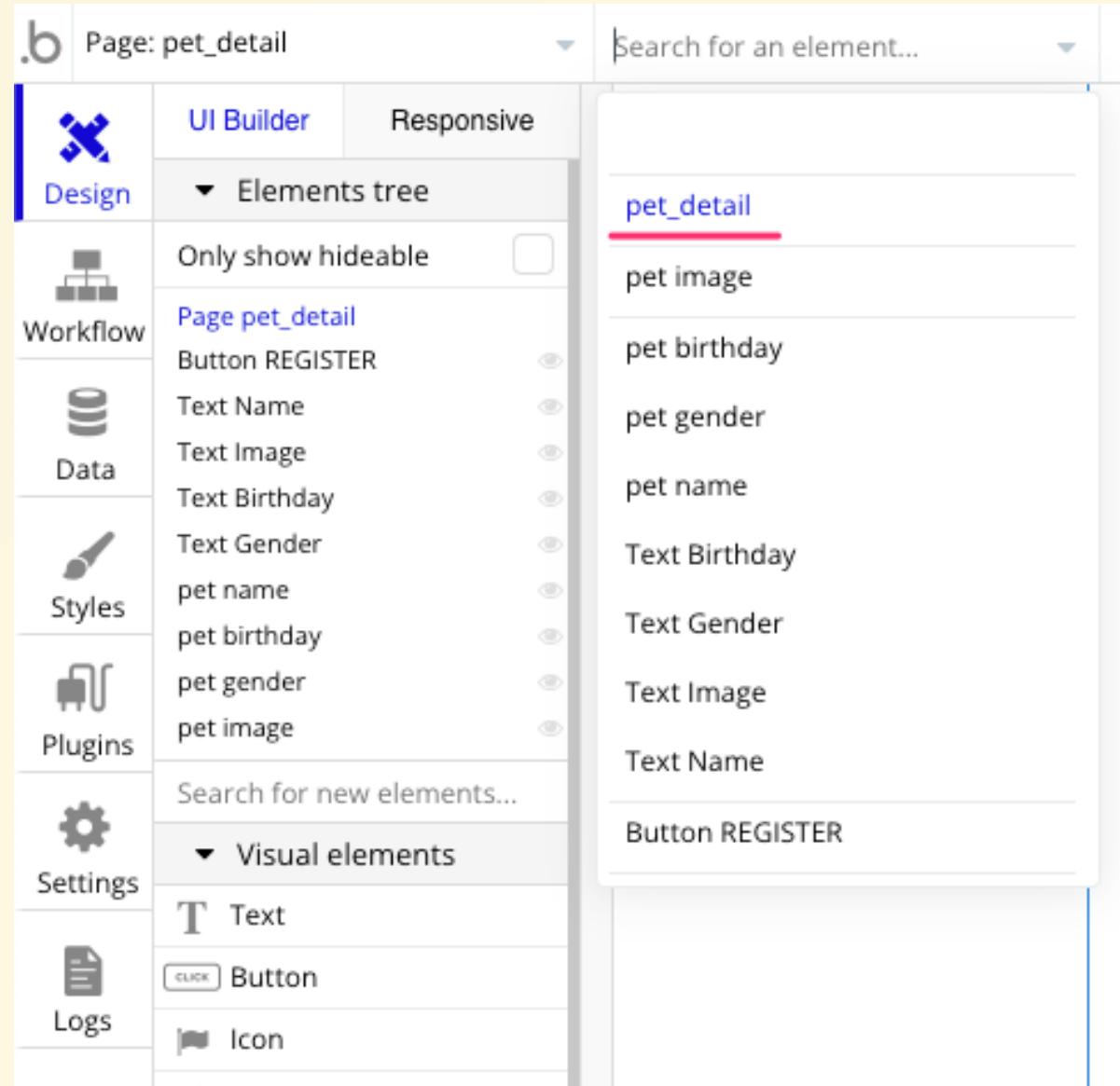
REGISTER

- Next, let's place the elements for display
- Let's drag each element from "Visual elements"
 - Name: Text
 - Image: Image
 - Birthday: Text
 - Gender: Text
- All elements should have a width (w) of 220px

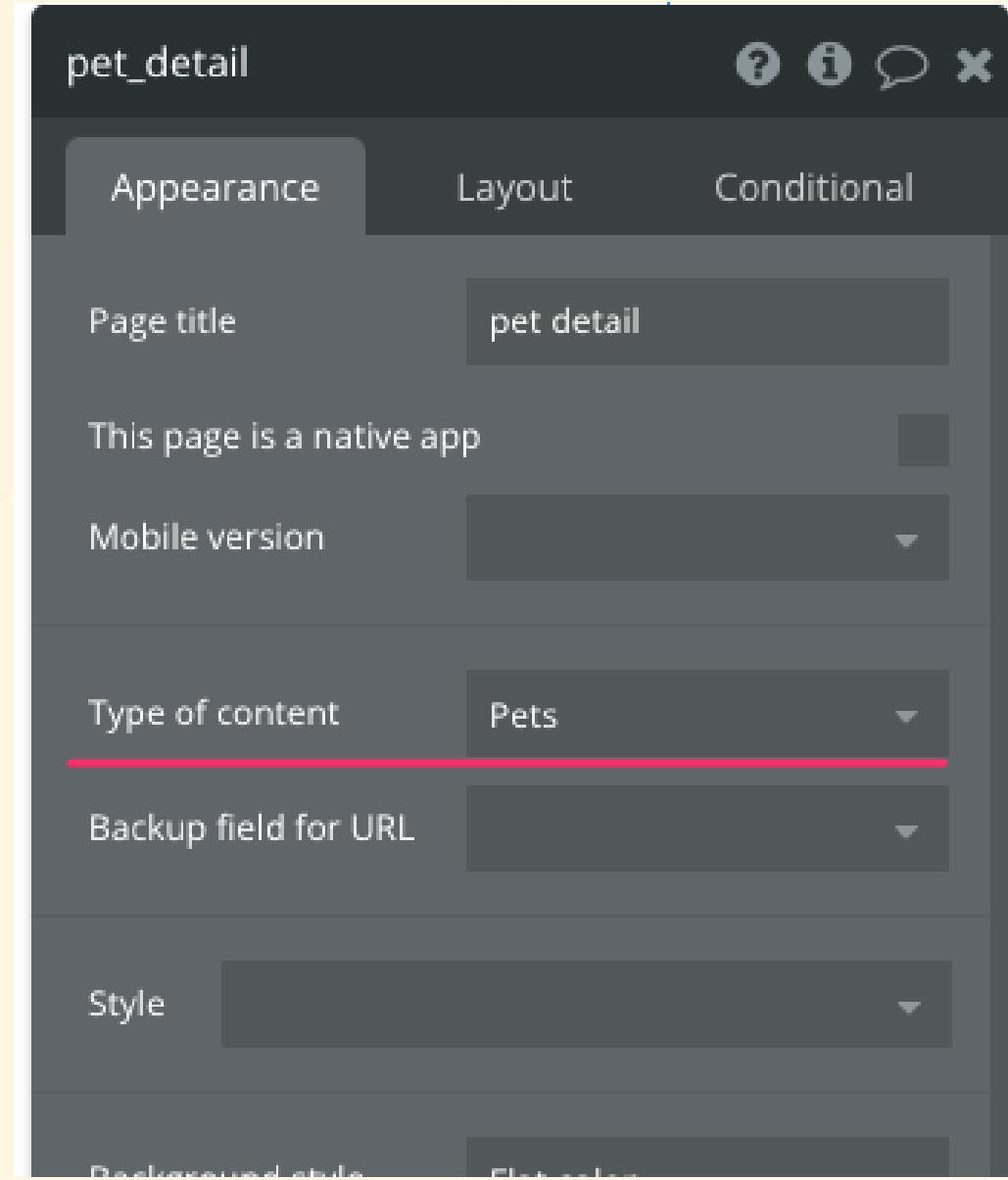


- Once the elements are ready, we will define the actual values to be referenced from the database
- First, we will display a dialog for advanced configuration of the screen itself

- Select **pet_detail** from the upper left part of the right panel to open a dialog box.
 - It is also convenient to select an element here if there are several overlapping elements!



- In the **Type of content** field, specify **Pets**.
- This way, you can specify the type of database from which you want to display this screen, and you only need to specify which fields of that type to use in each field.

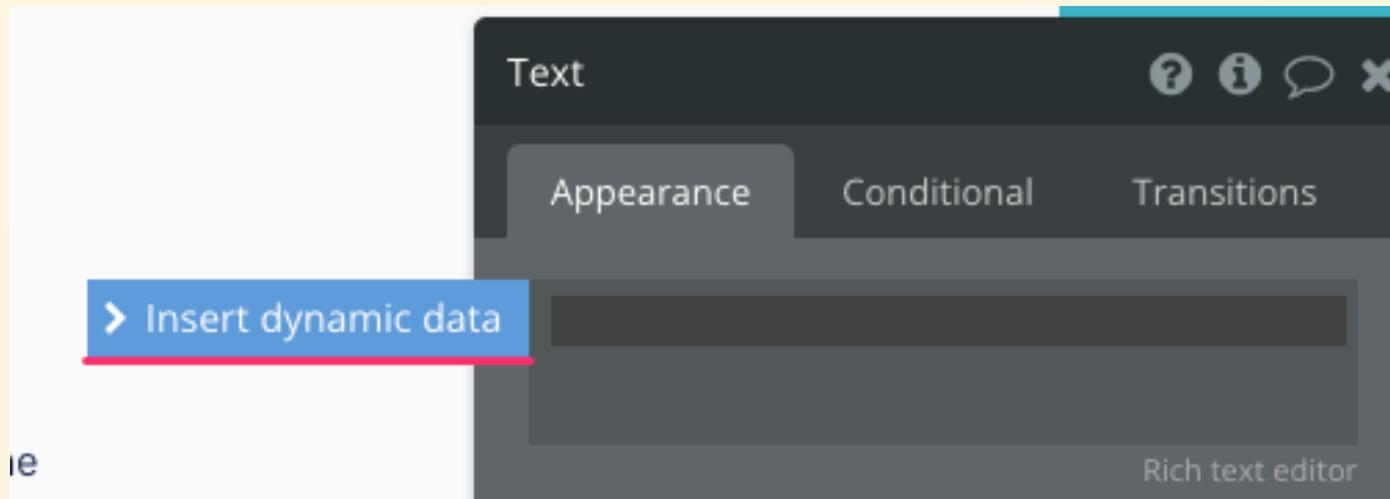


- By the way, although the type could be specified, the specific information on which pet (Porgy or Tama) is specified when moving from the list screen, which will be set later.

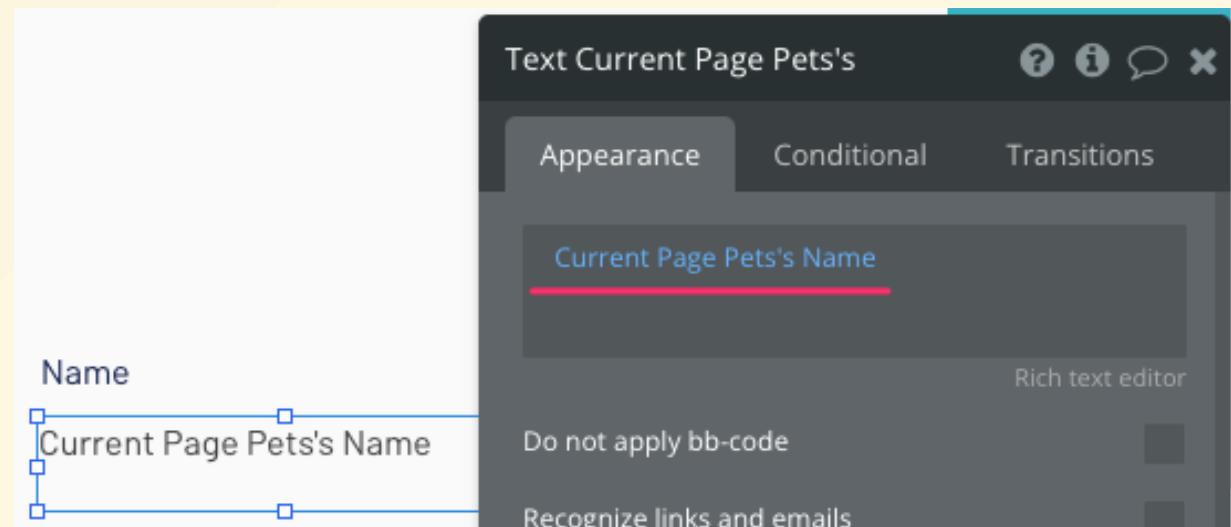
- Now let's link the elements for display with the Pets type field.
- Let's start with Name.
- Remember how we want to dynamically display values retrieved from the database, as in this case?



- Yes! Use "Dynamic data"!
- Double-click on the Name element and select
..... . edit me... Click on the ...edit me...
and you will see a button Insert dynamic data` , click on it.



- Then a pull-down will appear, from which you click **Current Page Pets's**!
 - This literally means the information of the pets assigned to the current page
- You will then see the fields that the Pets type has, select ``s Name



- Let's set up the Dynamic data for Image / Birthday / Gender in the same way
- After setting up, select all elements related to pet details and center them
 - Right click on the selected elements --> Center horizontally

- After setting up, let's preview it!
- The values are not displayed correctly, are they?
- This is because you haven't specified which pets you want to see in the list yet.
This is because you have not yet specified which pet information is to be displayed in the list screen.
 - The **Lo**rem ipsum is a

Name
Lorem ipsum...

Image

Birthday
Nov 12, 2022 10:00 pm

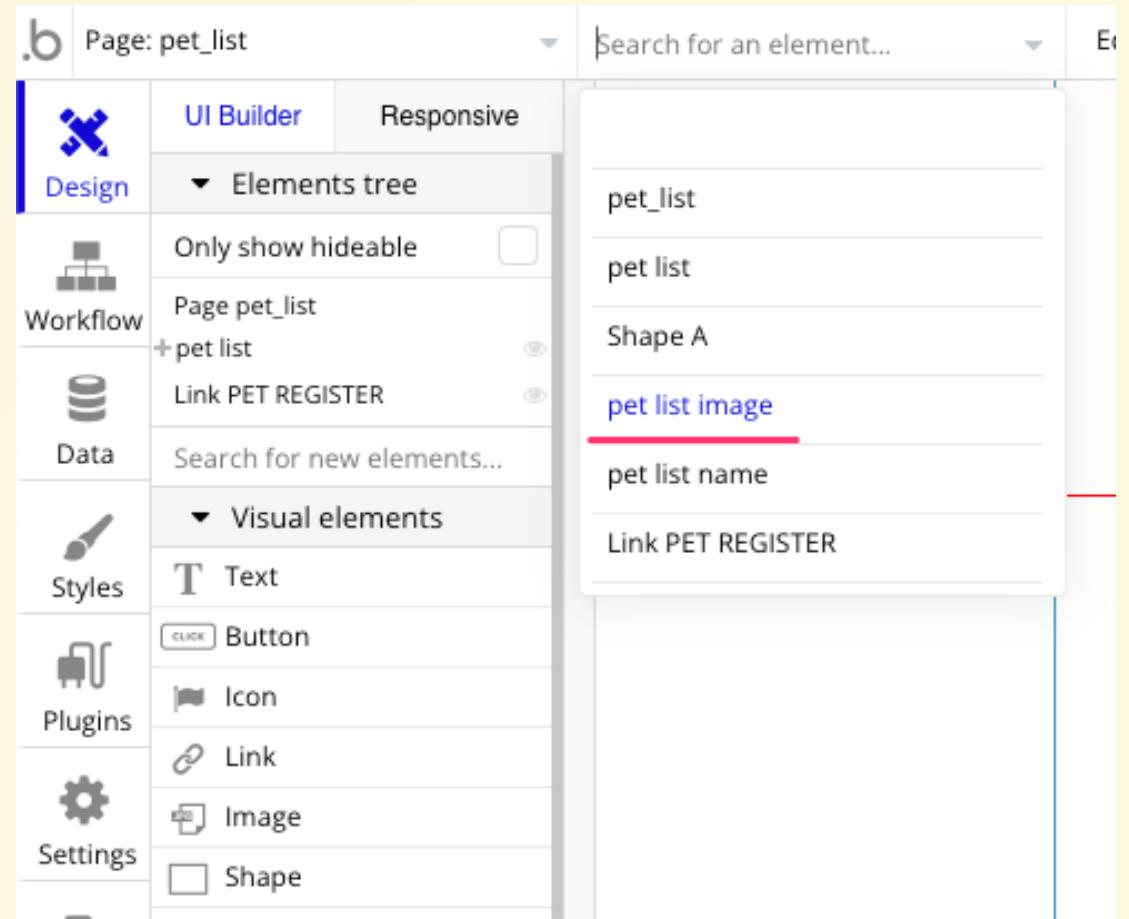
Gender
Lorem ipsum...

REGISTER

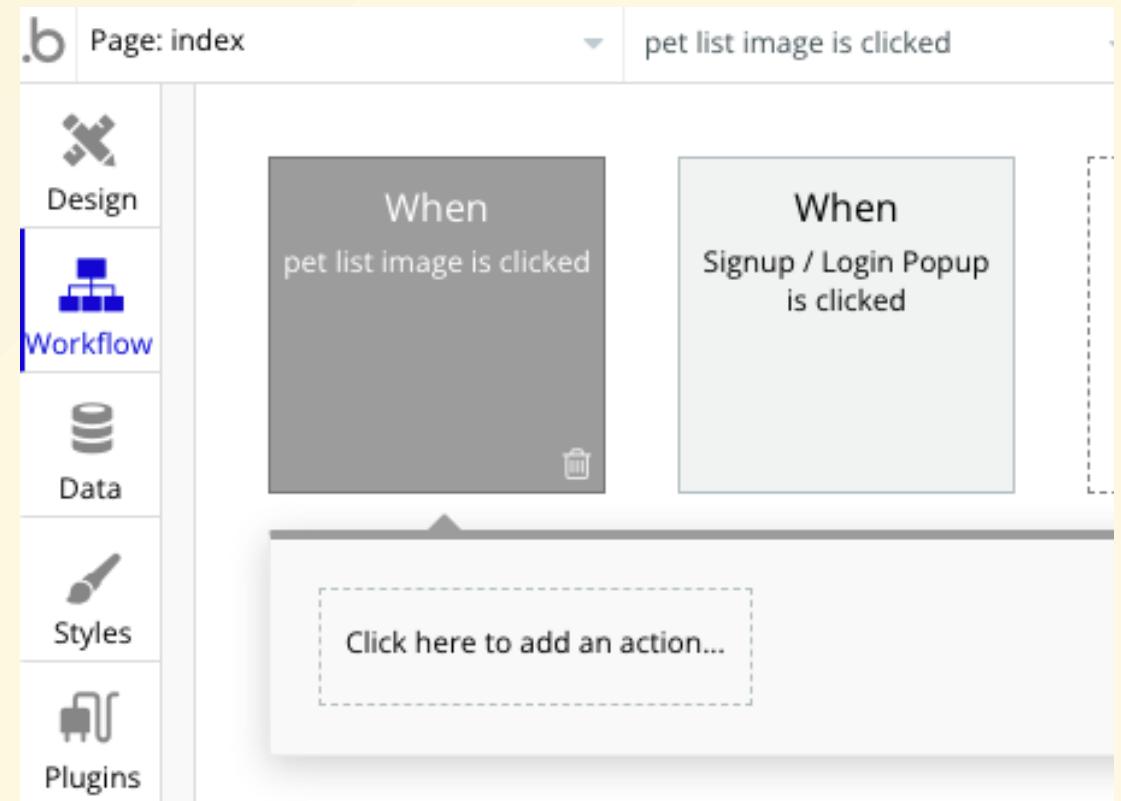
Now let's connect the list page and the detail page

- Switch to the index page from the top left menu
- As an image of the screen operation, we would like to move to the detail screen of the pet when the image of the pet displayed in the pet list screen is clicked.
- So, let's set up a workflow for the pet image on the list screen.

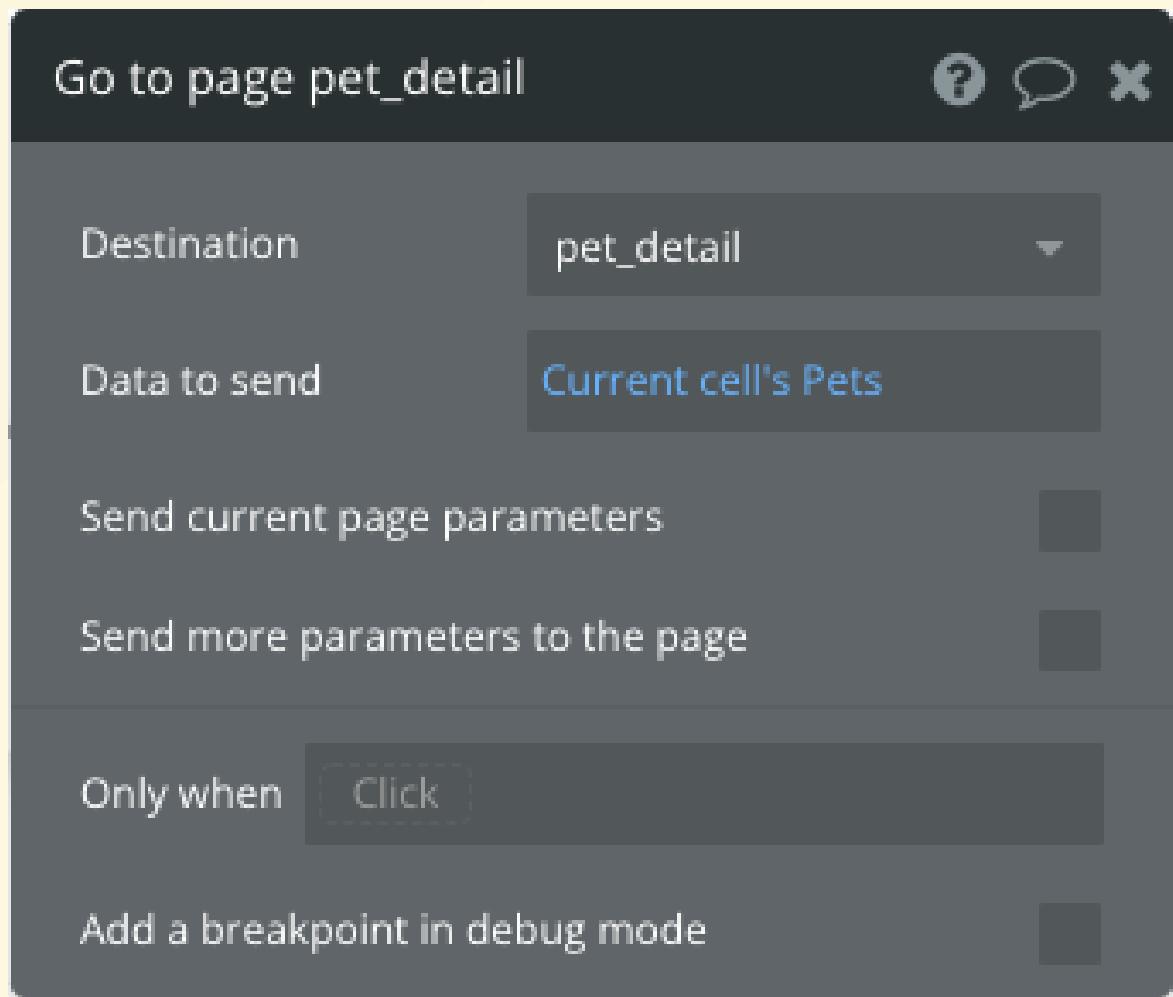
- Click on "pet list image" from the element list
- Click **Start/Edit workflow** in the dialog for advanced settings



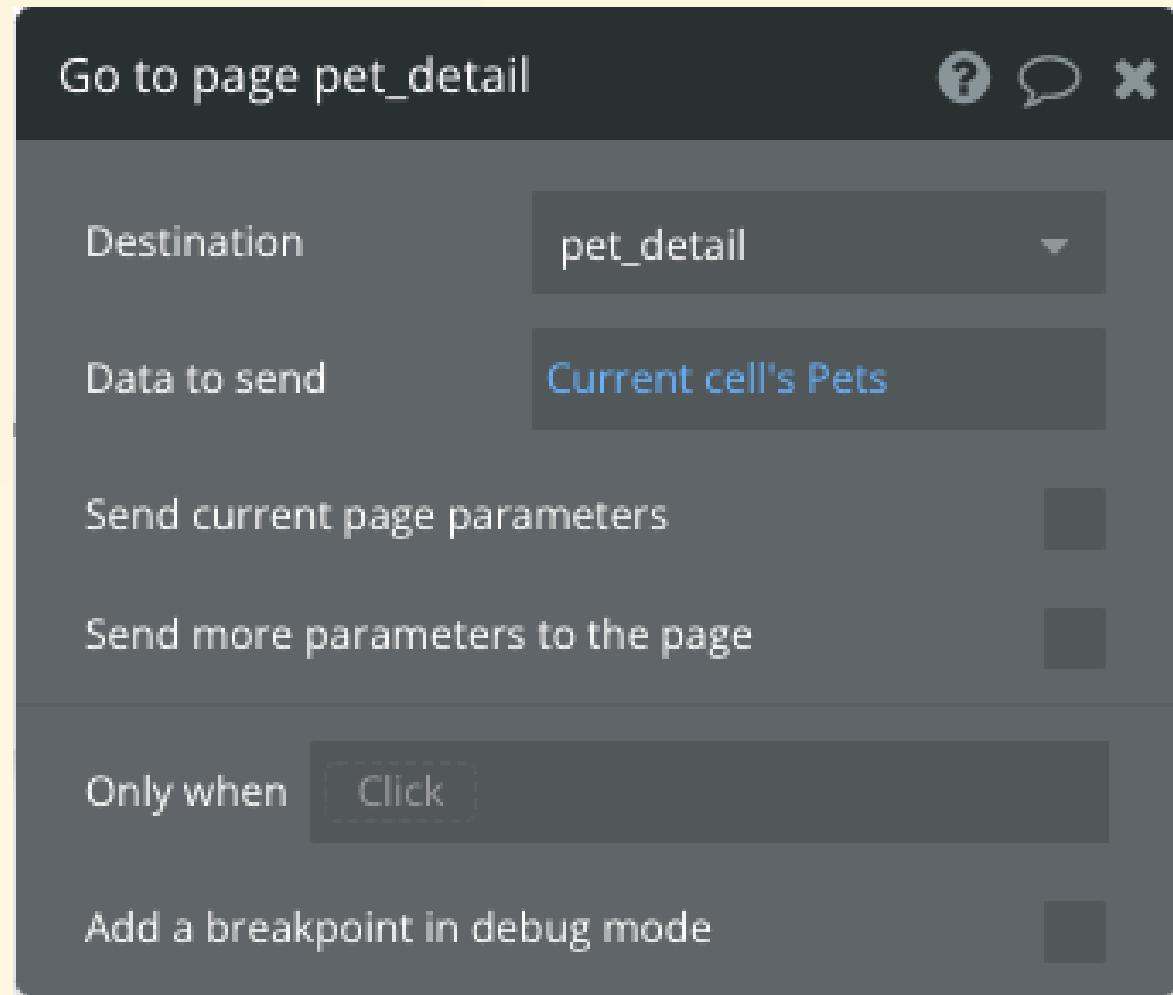
- Then you will see that the "When pet list image is clicked" box is in the state it was in before you set the behavior.



- So from "Click here to add an action" select Navigation -> Go to page
- A dialog box will appear, and for **Destination** (destination), specify "pet_detail".



- Then, set **Data to send** to "Current cell's Pets".
 - Now you can specify that when you move to the pet details screen, the pet information of the current cell is sent to the destination.



- Now let's preview it!
- Did you see that when you select an image of a pet in the list screen, the details screen for that pet appears?

Name

メロ

Image



Birthday

Sep 5, 2022 12:00 am

Gender

Male

REGISTER

- Here's some display advice
- The Birthday and Gender displays are a bit bland, so change the display format!

Name

メロ

Image



Birthday

Sep 5, 2022 12:00 am

Gender

Male

REGISTER

Exercise 2: Changing the birthday format to Japanese style

Name

×□

Image



Birthday

2022年9月5日

Gender

Male

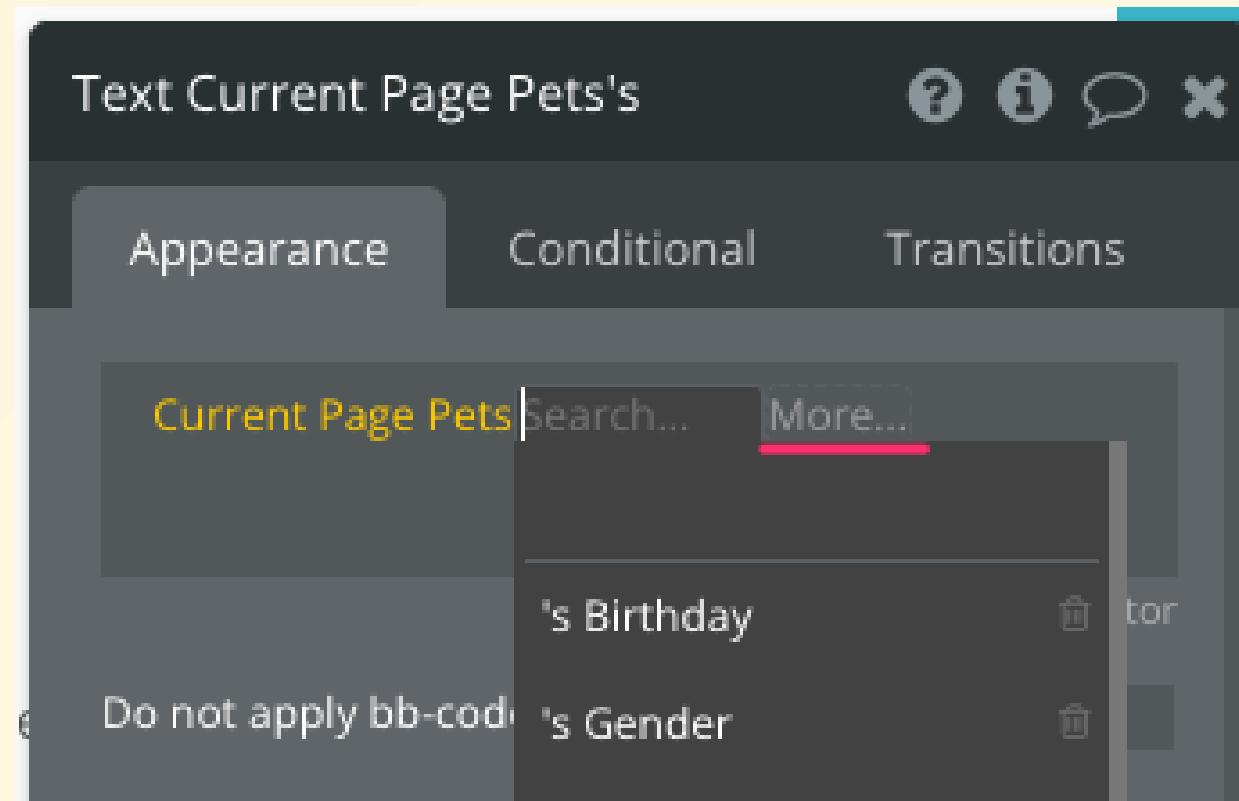
REGISTER

- On the pet_detail screen, open the Design tab.
- Double-click on the Birthday display element to open the Advanced Settings dialog box.

- Then, click on the Birthday section of

Current Page Pets's
Birthday

, and you will see a More item behind it, click on it.



- Then you can further specify the format in which the Birthday values are to be displayed
- This time, click :formatted as YYYY/MM/DD at the top
 - YYYY/MM/DD should contain today's date.

Appearance Conditional Transitions

Current Page Pet's Birthday Search...

:formatted as 11/10/21

:formatted as JSON-safe

<- range ->

+seconds:

+minutes:

+hours:

+days:

+months:

+years:

change seconds to

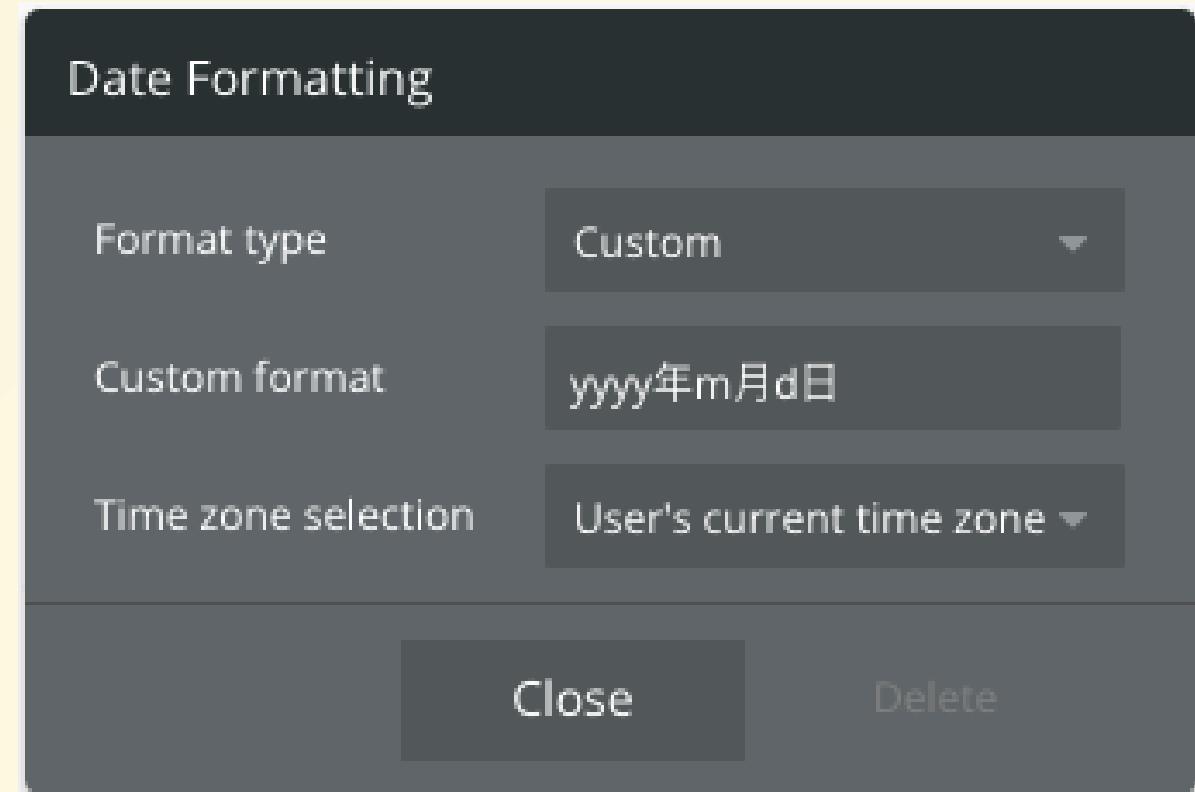
change minutes to

This element isn't clickable

Start/Edit w...

This element is visible on page

- Then you will see the Date Formatting dialog next to it.
 - Here you can specify the formatting in detail.
 - In this case, select Custom for Format type and set Custom format to yyyy year, m month, d day
- .



- Now let's preview it
 - Open the index page once and then preview
- If you open the pet details from the pets list, you will see that the Birthday date is in the format you specified.

Name

×

Image



Birthday

2022年9月5日

Gender

Male

REGISTER

Exercise 3: Let's change the gender labels!

- If the gender is "Male", the label will say "Boy", and if the gender is "Female", the label will say "Girl"!

Name
メロ

Image



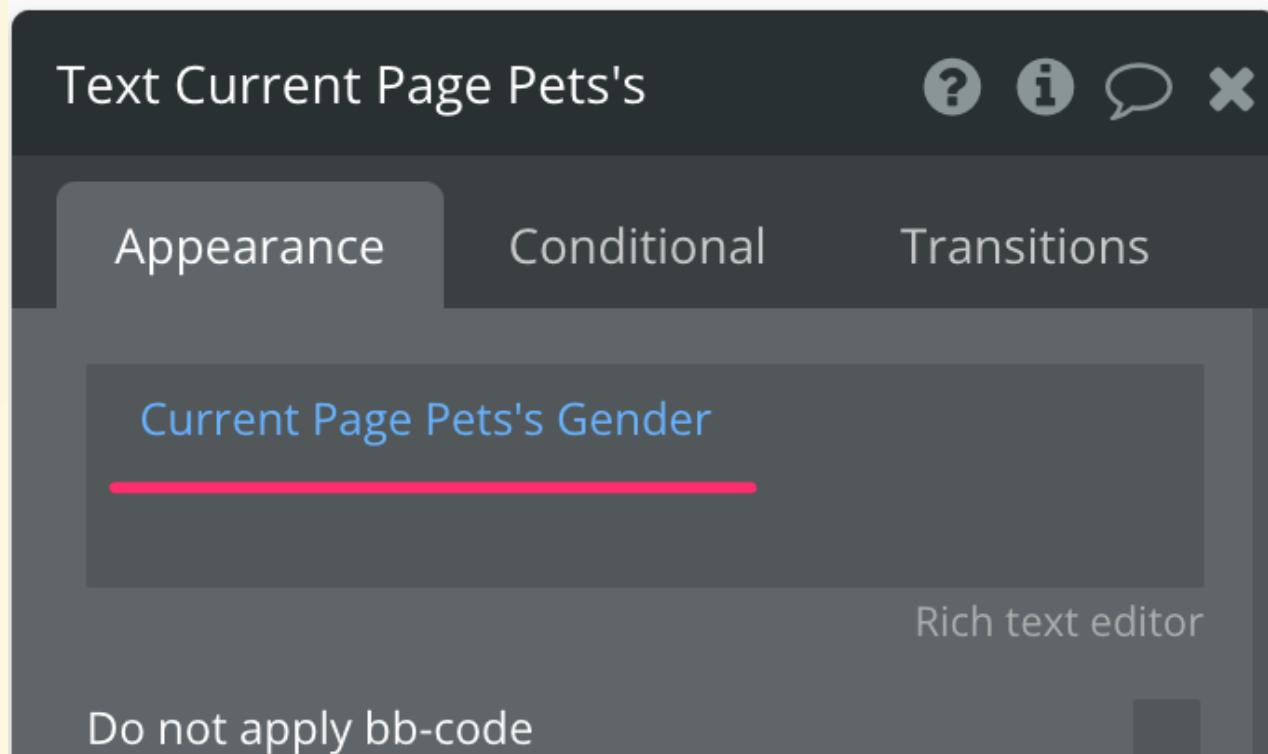
Birthday
2022年9月5日

Gender
男の子

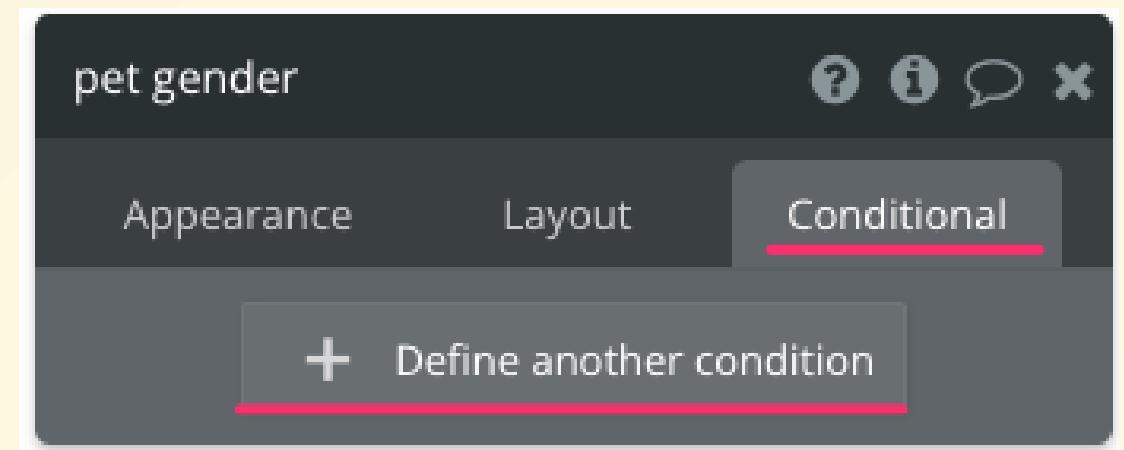
REGISTER

- Open the pet_detail screen in the Design tab.
- Double-click on the Gender display element to display a dialog box for detailed settings.
- This time, let's change the string to be displayed by setting the condition from the Conditional tab

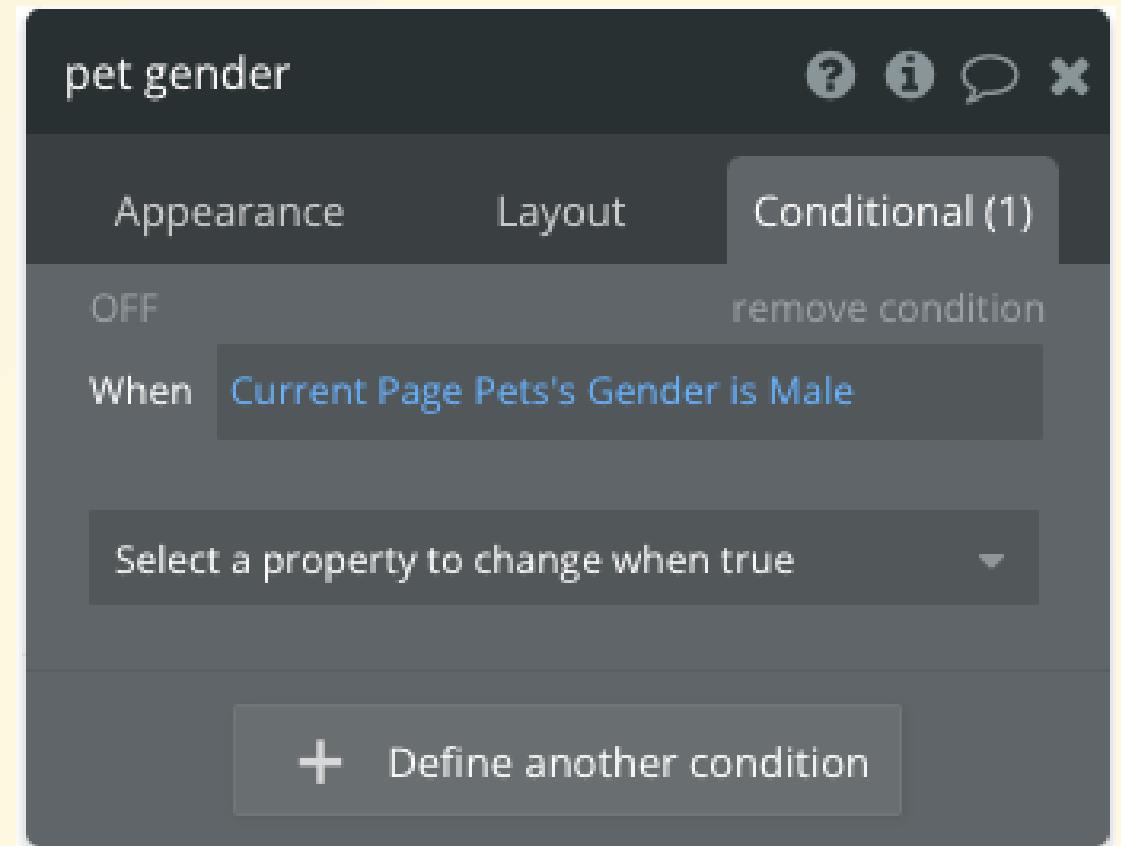
- First, move the cursor to the
Current Page Pets's Gender
field that you have just
entered on the Appearance
tab, and delete the entry
with the Delete button.



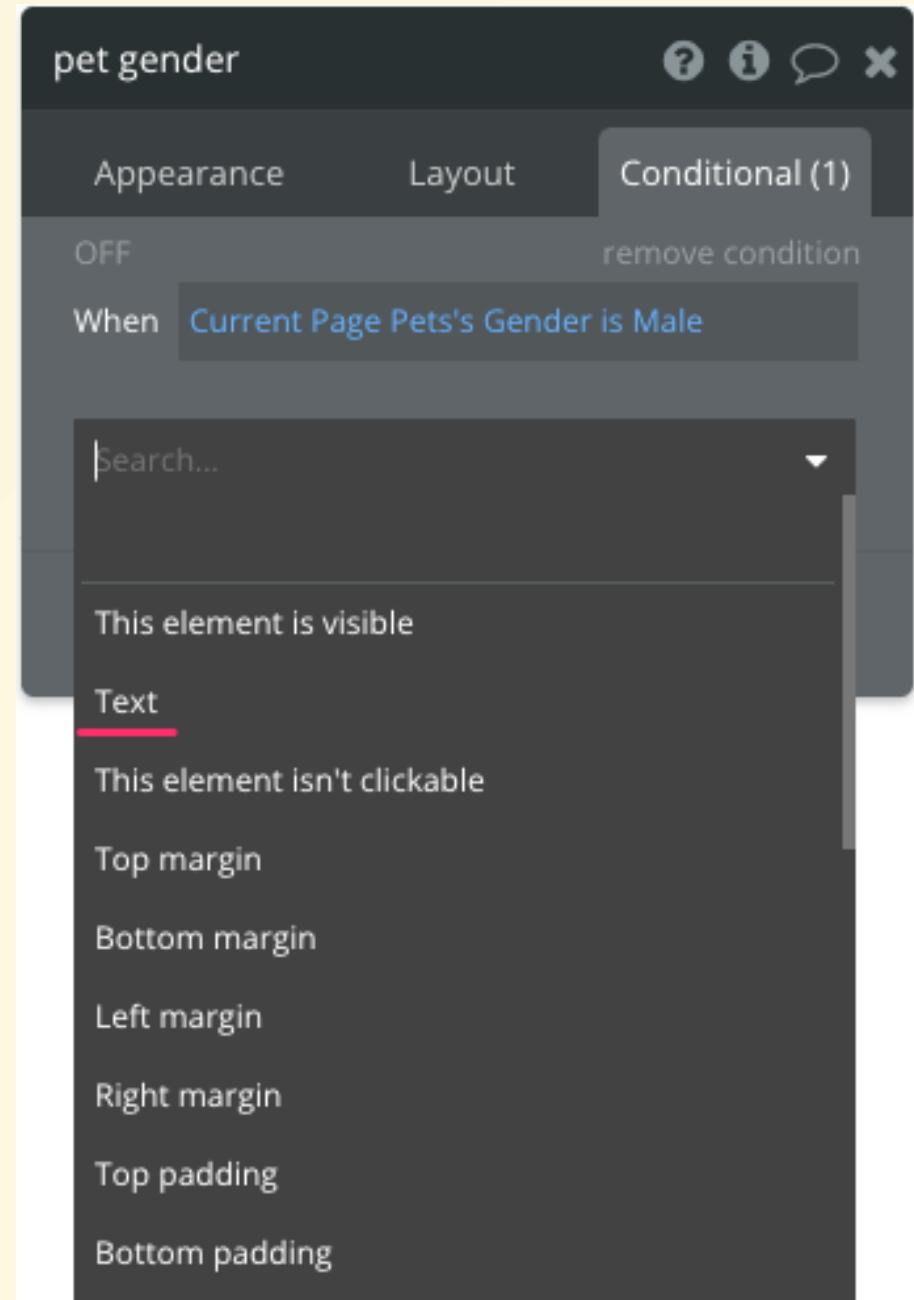
- Then open the Conditional tab and click the Define another condition button to add a condition.



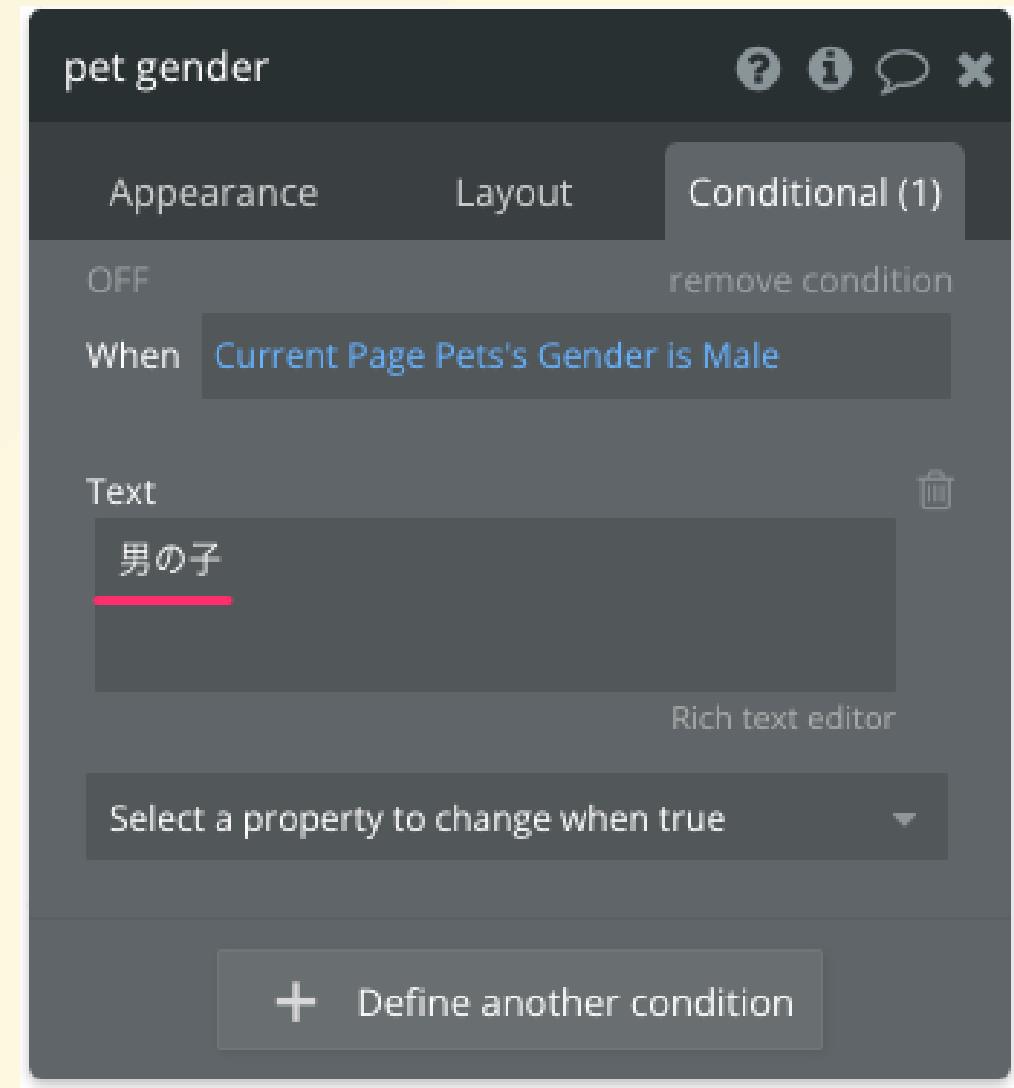
- First, specify the condition "If the currently displayed pet's gender is "Male"" in the When field.
- Only Male is manually entered, the others are selected from the pull-down menu.



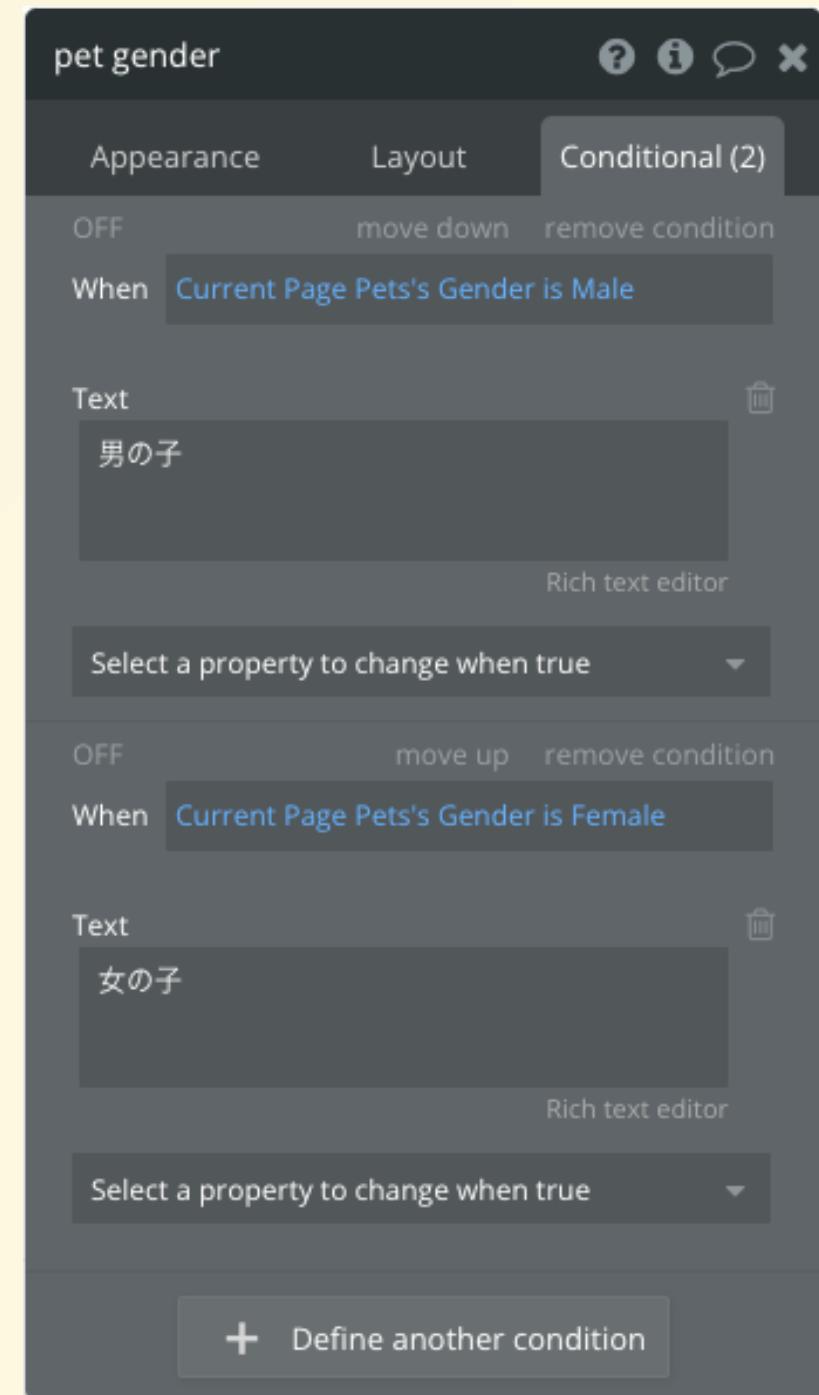
- Once the "When" condition is set, set the behavior when the condition is true
- In this case, we want to display a specific string, so we select "Text" from the "Select a property..." pull-down menu. In this case, we want to display a specific string, so select "Text" from the "Select a property..." pull-down menu.



- Then, a text input field will appear, and enter "boy" there.



- Let's set up the "girl" in the same way



- This would be writing the following process

If the current pet's Gender is Male, it will display "Boy".
If the current pet's Gender is Female, it will display "Girl".

We're getting a bit of a programming element here!

- If you want to process the value of the element itself, as in Birthday, you should specify Format, but if you want to do something based on the value of the element, as in this case, you should use Conditional!

- Let's run the preview from the index page.
- If you open the pet details from the pet list, you will see that the Gender values are "boy" and "girl".

Name

メロ

Image



Birthday

2022年9月5日

Gender

男の子

REGISTER

Exercise 4: Let's set up a lead from the details screen to the list screen

- Let's prepare a lead from the detail screen to the list screen.
- I'm sure you have a good idea of what you want to do, so I'll just give you a hint and an idea of what to do.

Hint

- The element to use is **Link** of **Visual elements**.
- For the Link element, you can set the "Destination page" to the element itself instead of the Workflow for the screen transition.
- Completed image

Name

メロ

Image



Birthday

2022年9月5日

Gender

男の子

[← Back to list](#)

REGISTER

Let's create a header component.

- We will now leave screen creation for a moment and create "common parts".
- A "common part" is a collection of elements that are used in the same way on multiple screens.

Function of the header component to be created

- Link to pet registration page
- Control login/logout buttons based on login status



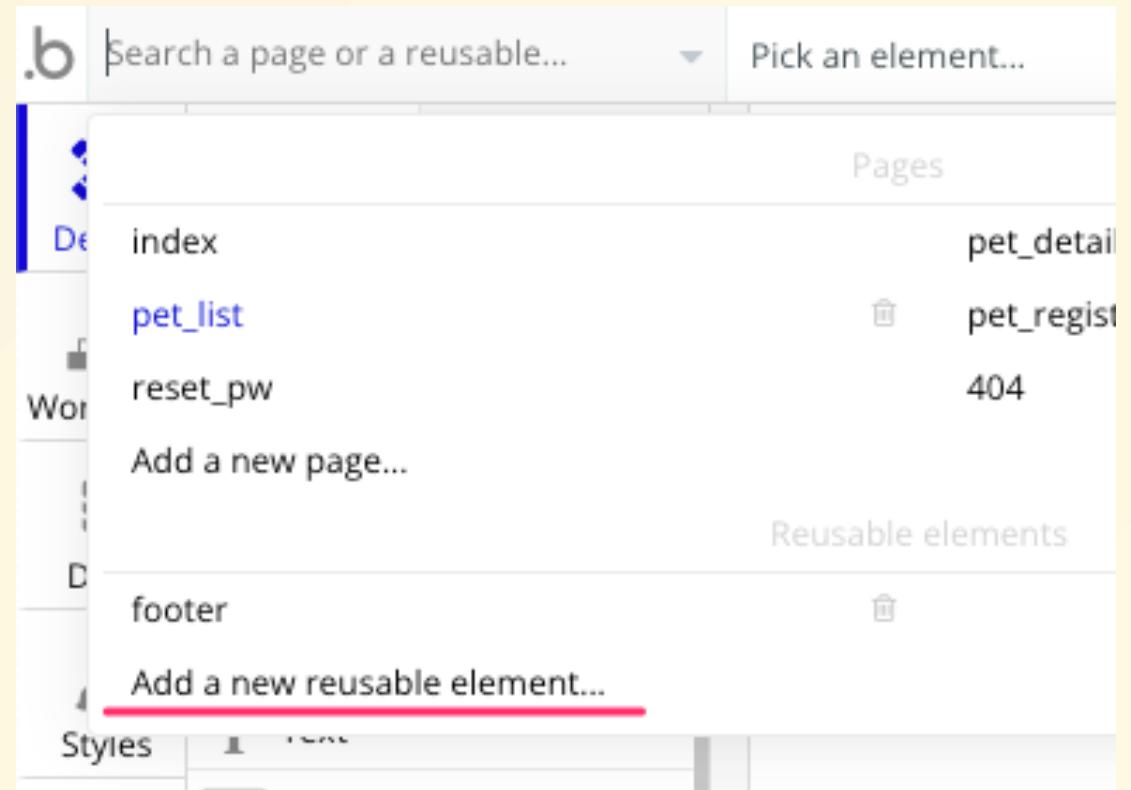
Why create common parts?

- It is time-consuming to prepare elements, such as header parts, that you want to use on multiple screens separately for each screen, isn't it?
- Also, if you prepare them individually, when you want to change their contents, you need to modify them for each screen, which increases the time and effort of development.

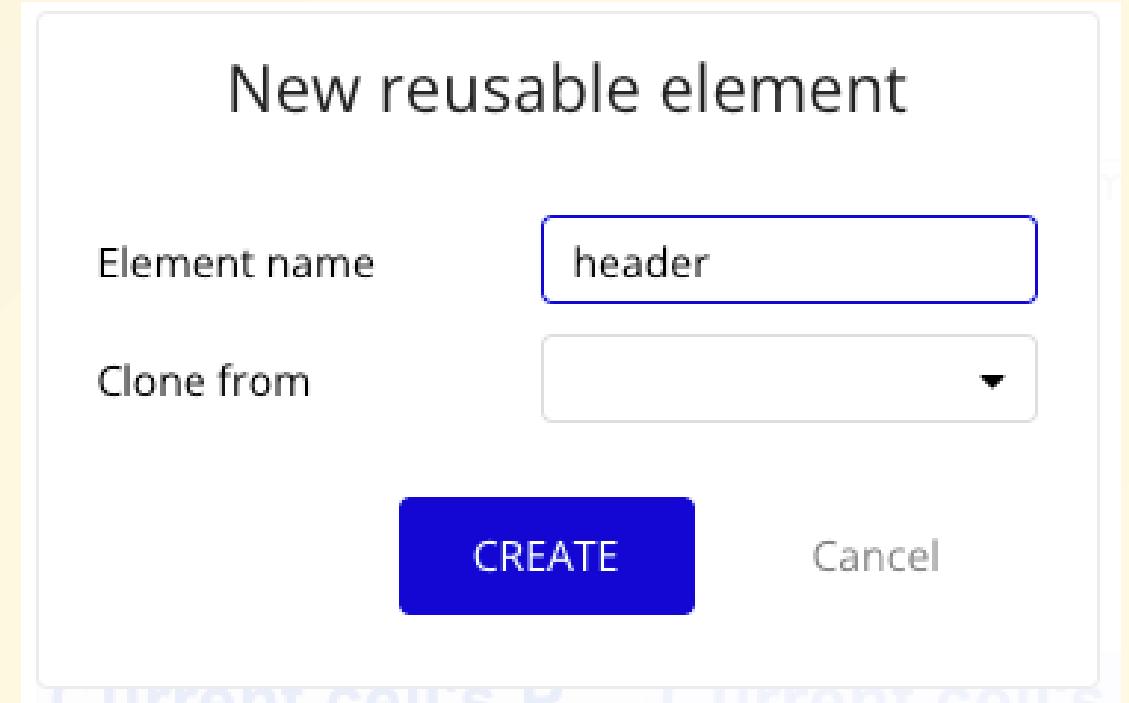
- Bubble solves such problems by using "Reusable elements" components!
- In this case, we will create a header component as "Reusable elements" and place it on the pet list, registration and details screens

Creating a header component

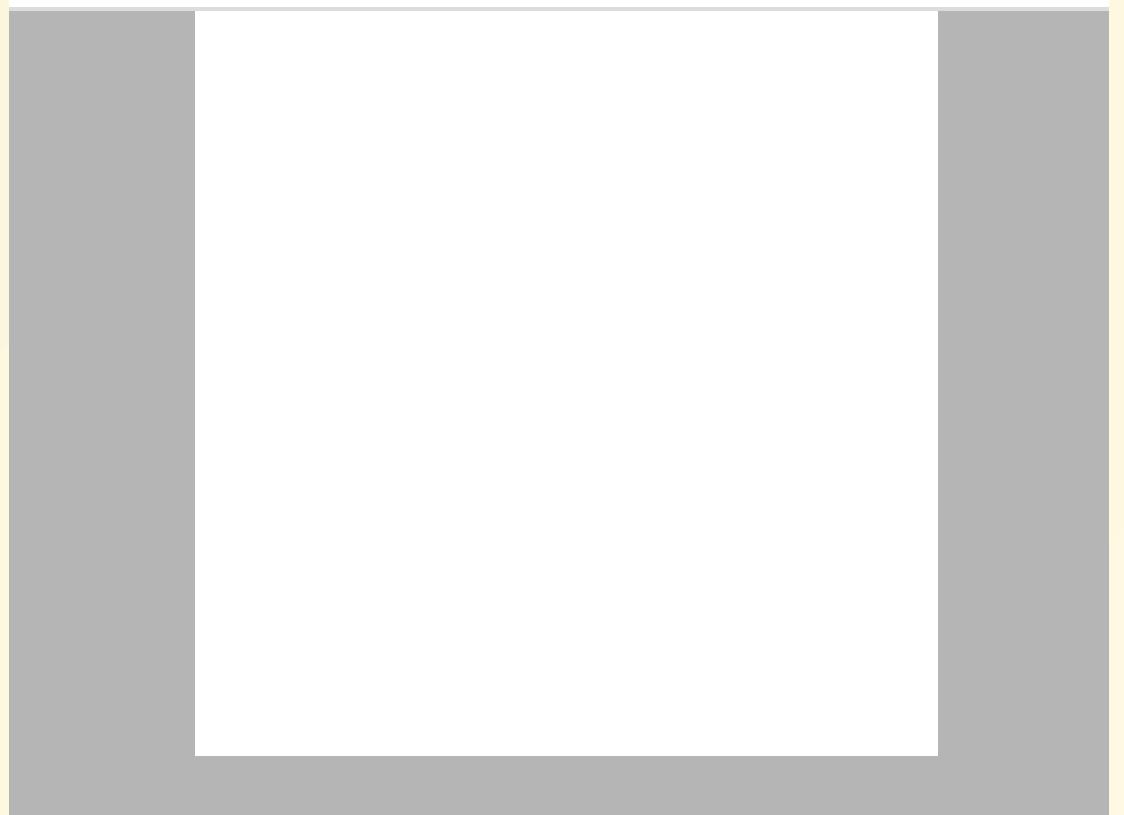
- Open the list of screens in the upper left corner and in it "Add a new reusable element..." Click on "Add a new reusable element..." in the list.



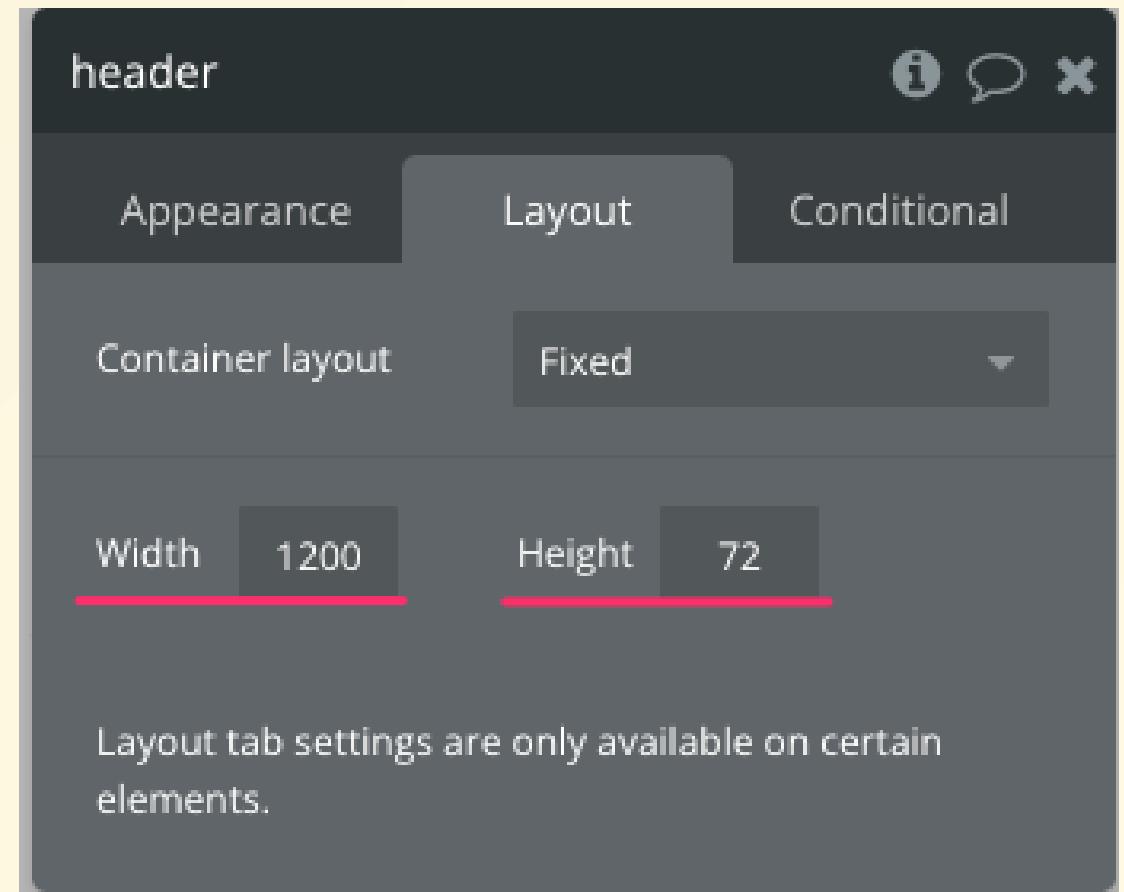
- Then, the same popup as when creating the screen is displayed. This time, enter "header" as the name of the common part and click CREATE.



- Then an area of 200px in width and height will appear in the right panel.
- The width of the area is a little narrow as it is, so first adjust the area.



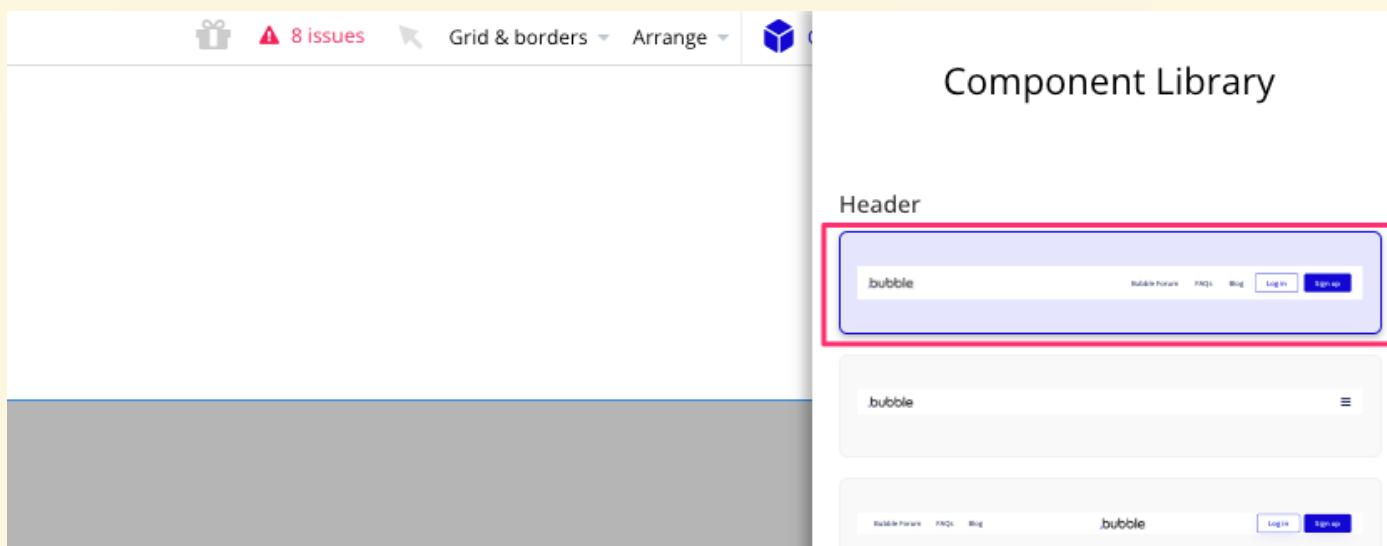
- Right-click in the right panel area --> Edit
- Select the "Layout" tab in the familiar pop-up window.
- Change **Width** to "1200" and **Height** to "72"



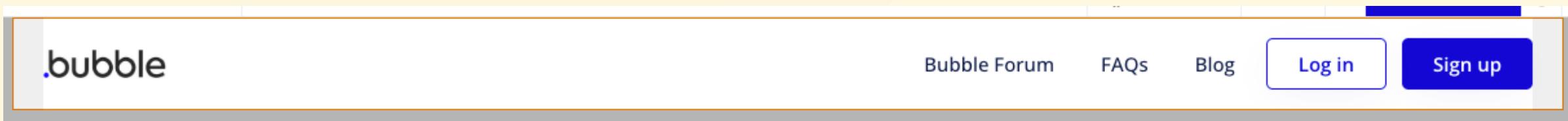
- Now we will create the common components of the header, but here we will use a useful feature
- In the menu at the top of the Bubble screen, you will find the "Components" menu



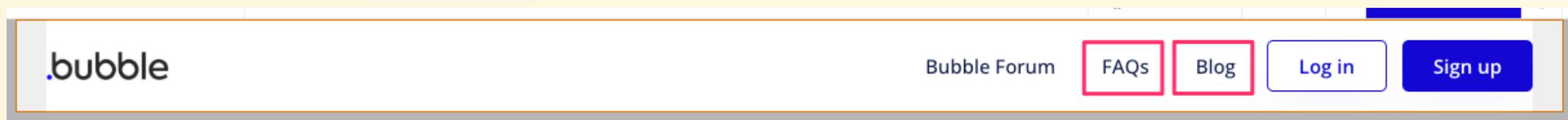
- This is a library of common parts that Bubble provides by default.
- In this case, we will select the topmost component from the "Header" components in this library and drag it to the right panel.
- Once placed, click the Close button under the Component Library to close it.



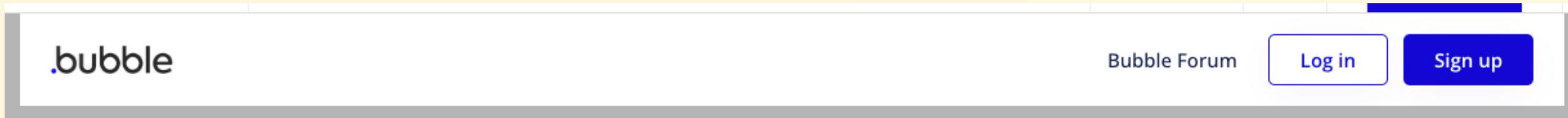
- Then, the sample Header that Bubble provides is displayed on the right panel.
- This time, we will customize it.



- First, let's delete what we don't need as a header component this time
- Select and delete the following two elements
 - FAQs
 - Blog
- Be careful not to delete other elements by mistake!
- Leave the "Bubble Forum" element as it will be used in the next step.



- Here's what it looks like after deletion

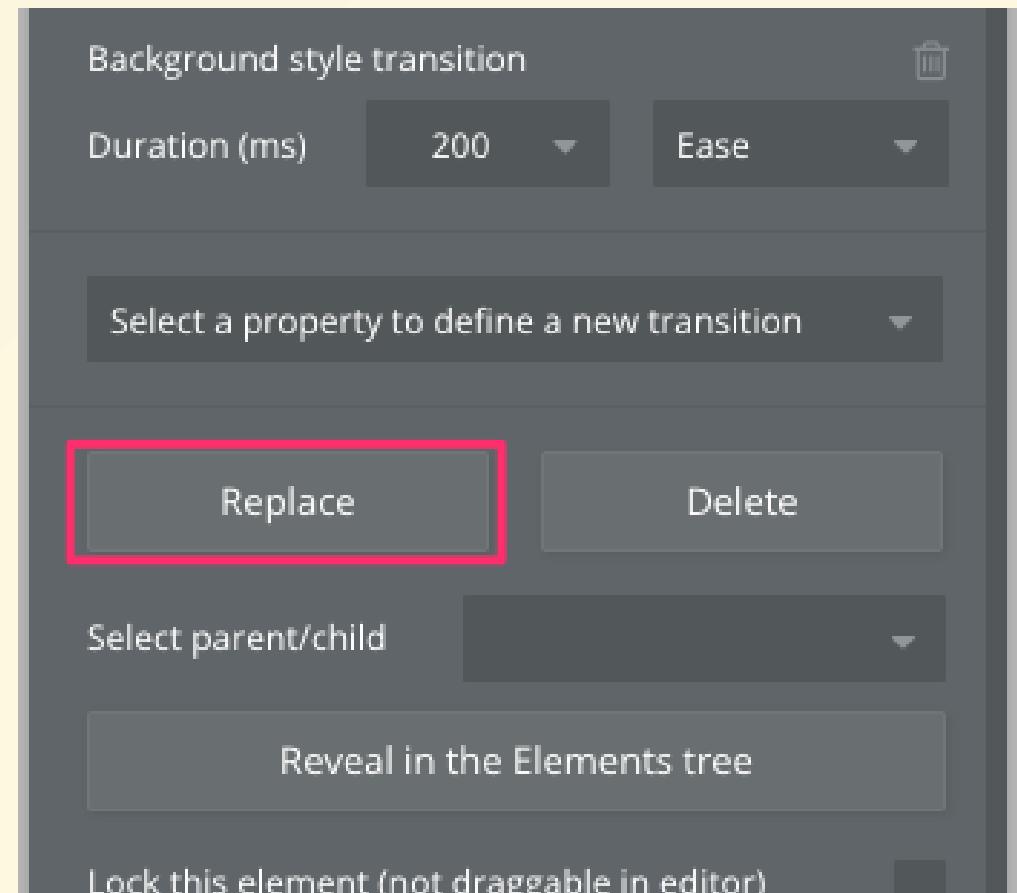


Setting up a link to the pet registration page

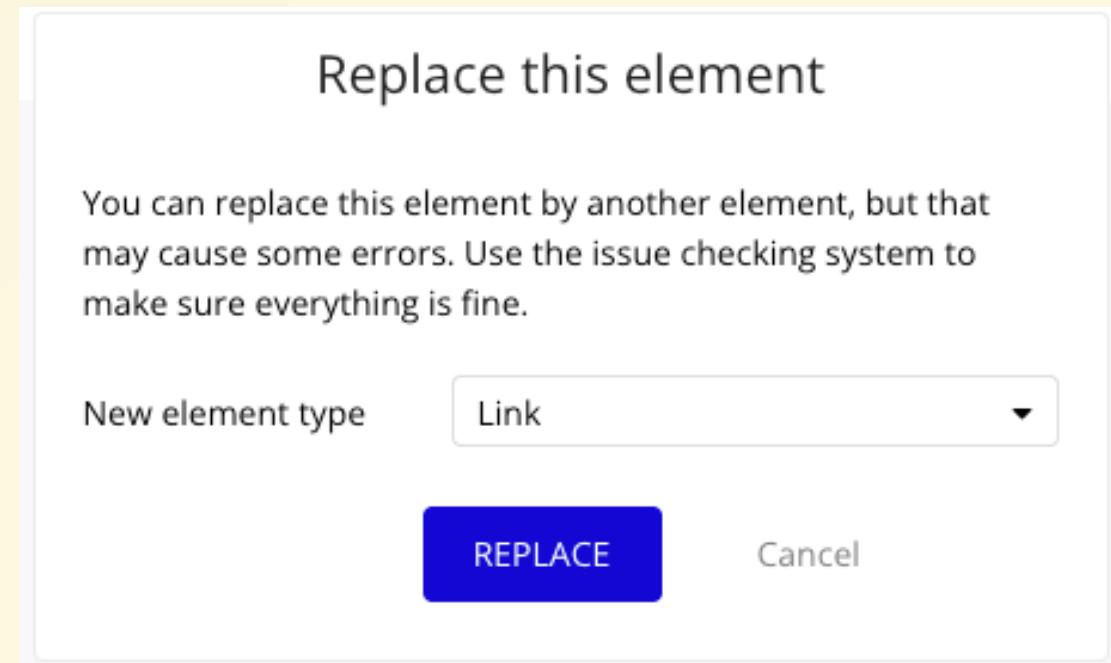
- First, let's set up a link to the pet registration page.



- Double-click on the "Bubble Forum" element you just left to bring up the edit popup
- Click on the button Replace at the bottom of the popup

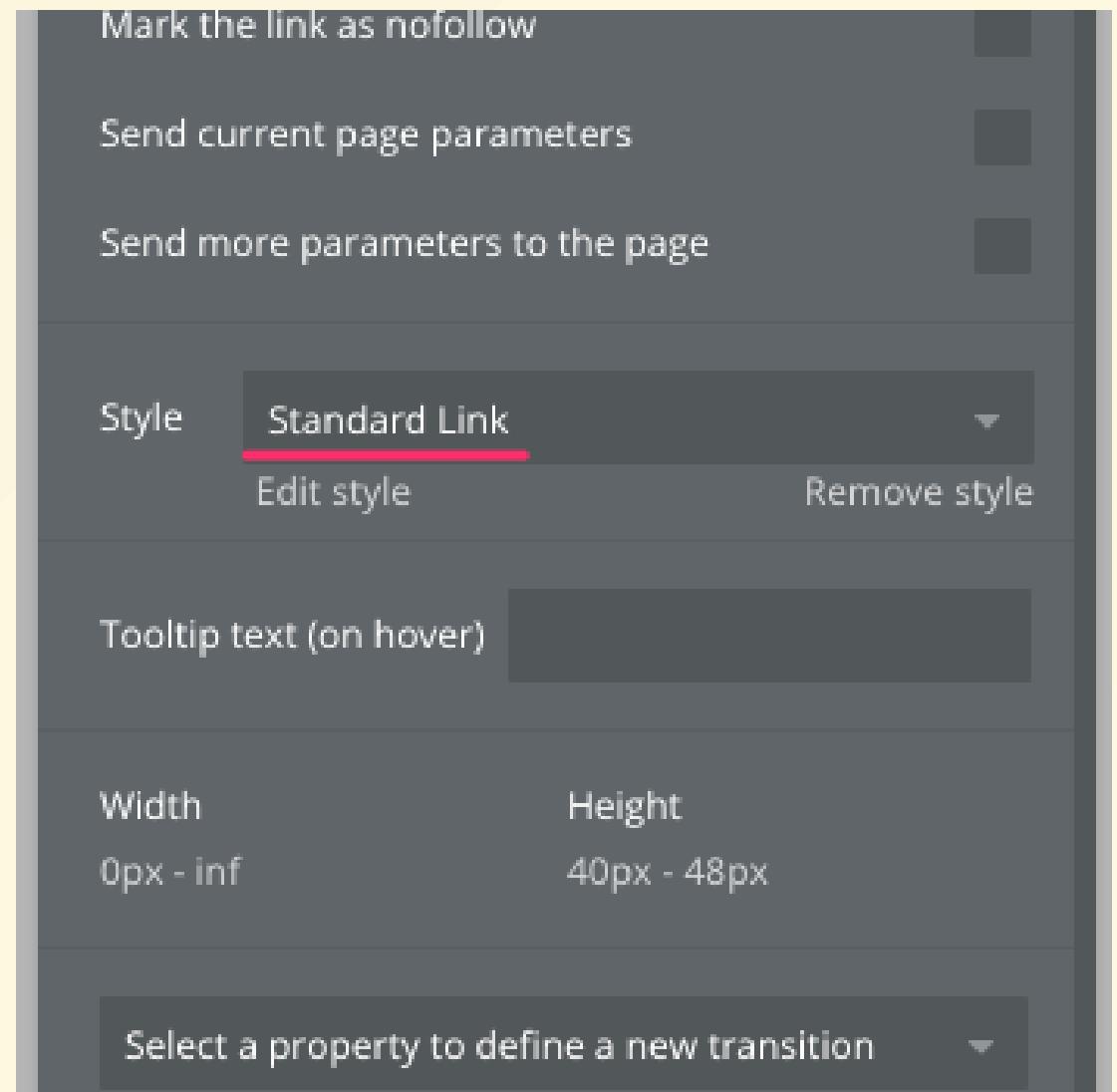


- This function changes the type of the currently selected element.
- In this case, we will replace the Text element with a Link element.
- Select "Link" in the New element type and click REPLACE

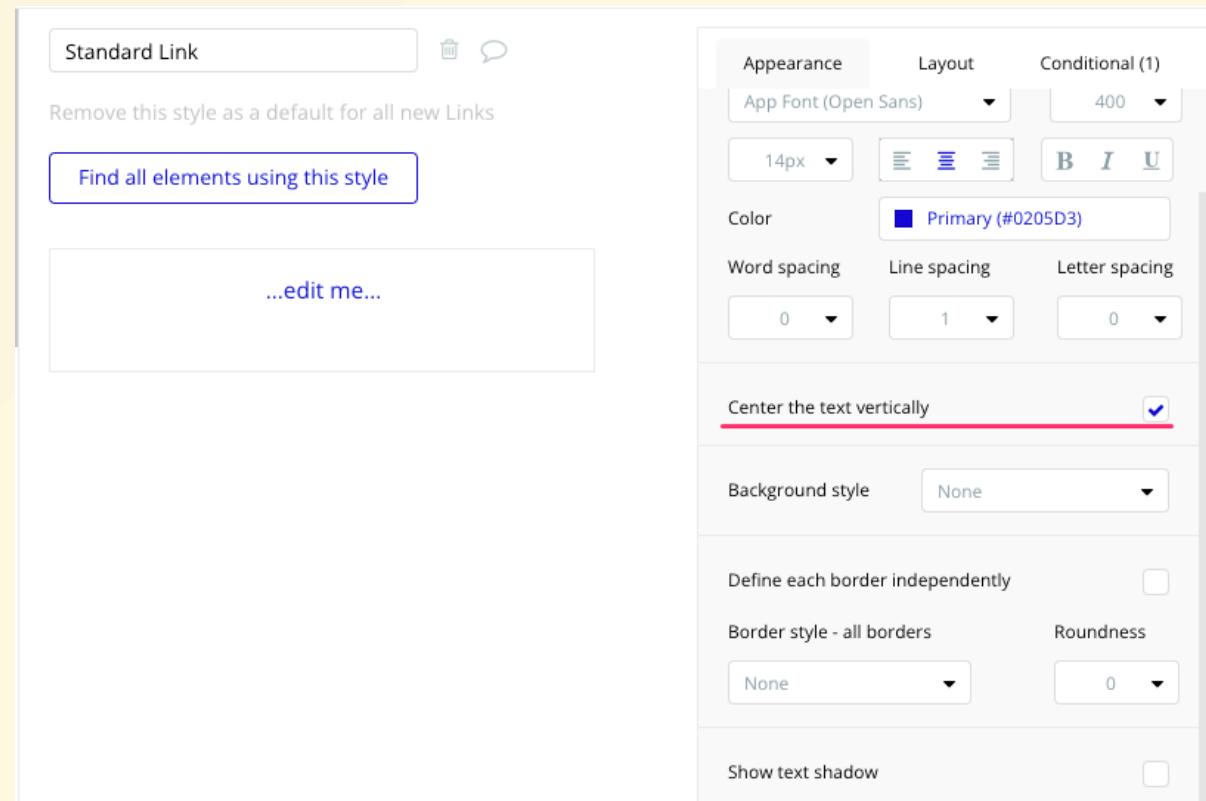


- The element type is now replaced by Link.
- However, the `Style` and `Conditional` elements that were set on the original element are still there, so we will change them.

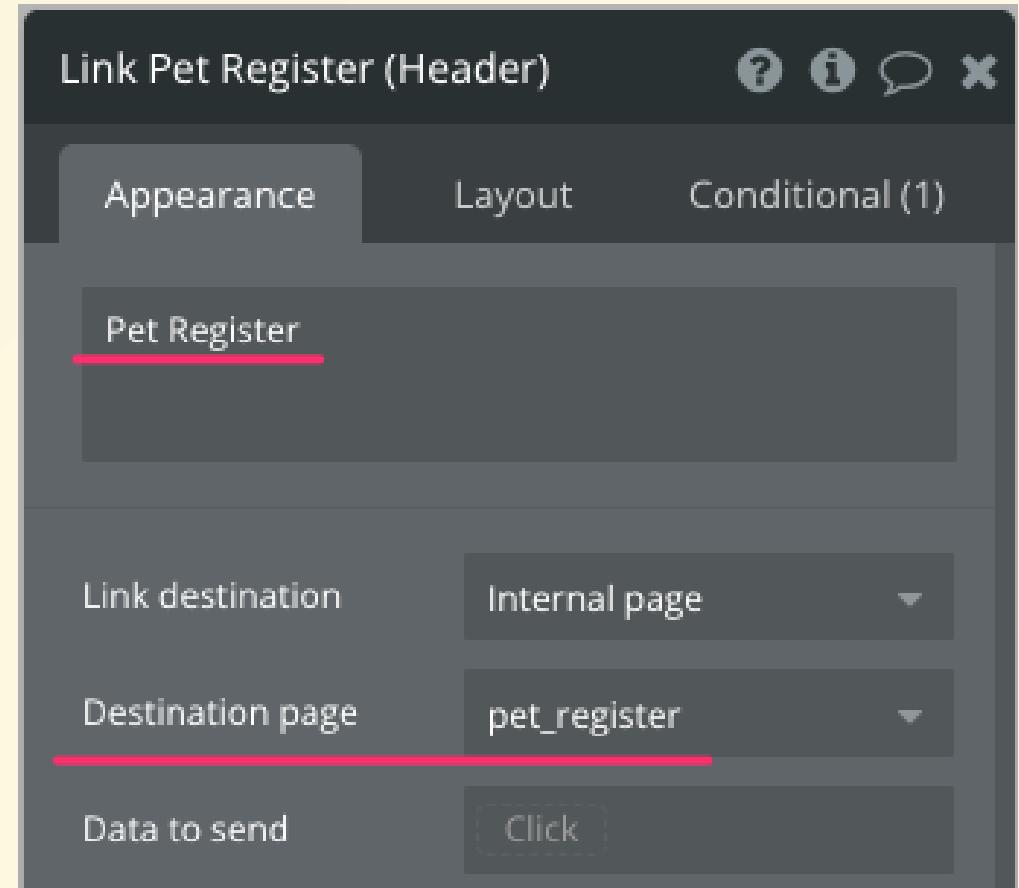
- First, change the **Style** to **Link**
- Select "Standard Link" for **Style** from the element's edit popup.
- If you leave it as it is, the text in the element is top-aligned, so center it vertically.



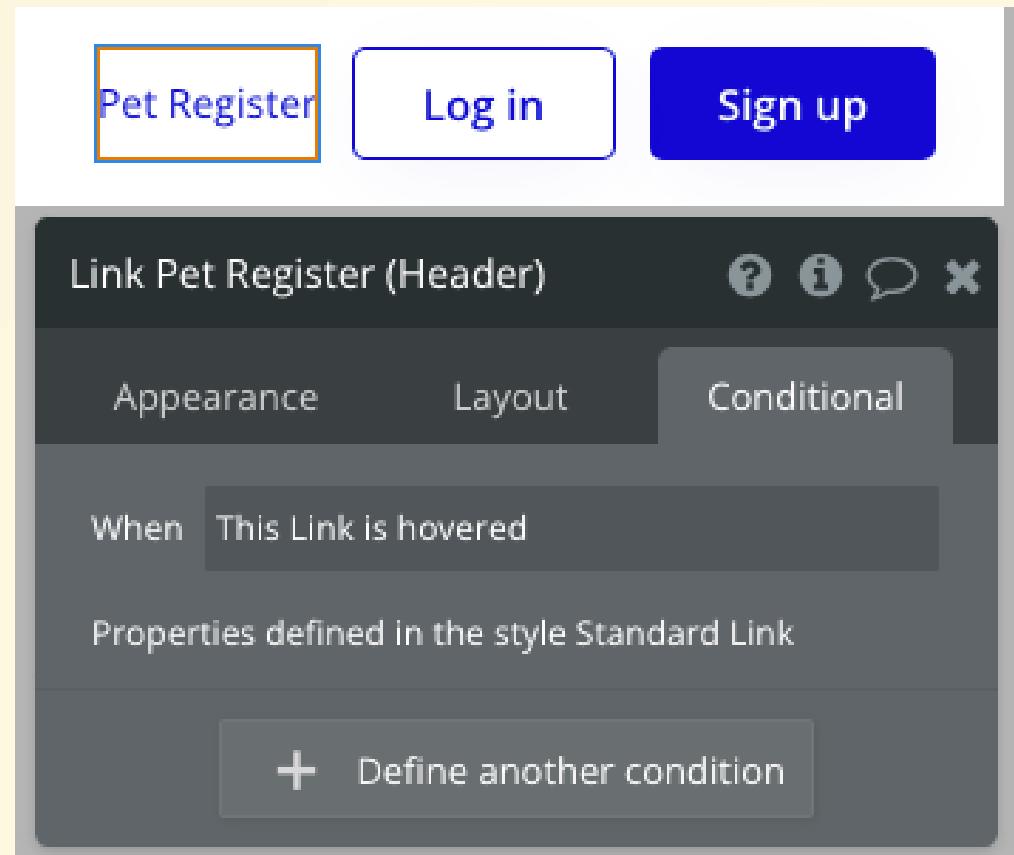
- Under "Standard Link" click on "Edit style".
- This will take you to the Styles tab, where you can edit the style of the Standard Link.
 - Now the text of the link will be vertically centered.



- Go back to the Design tab, and if the text that was previously centered vertically is now centered vertically, it is OK.
- Change the string content to "Pet Register".
- Set the destination to "pet_register" as well.



- The **Conditional** content has also been removed by changing the Style.
- If it is as shown in the capture, it is OK.



- This completes the installation of the link to the pet registration page.



Control Login / Logout buttons based on login status

- Next, we will prepare the login and logout buttons

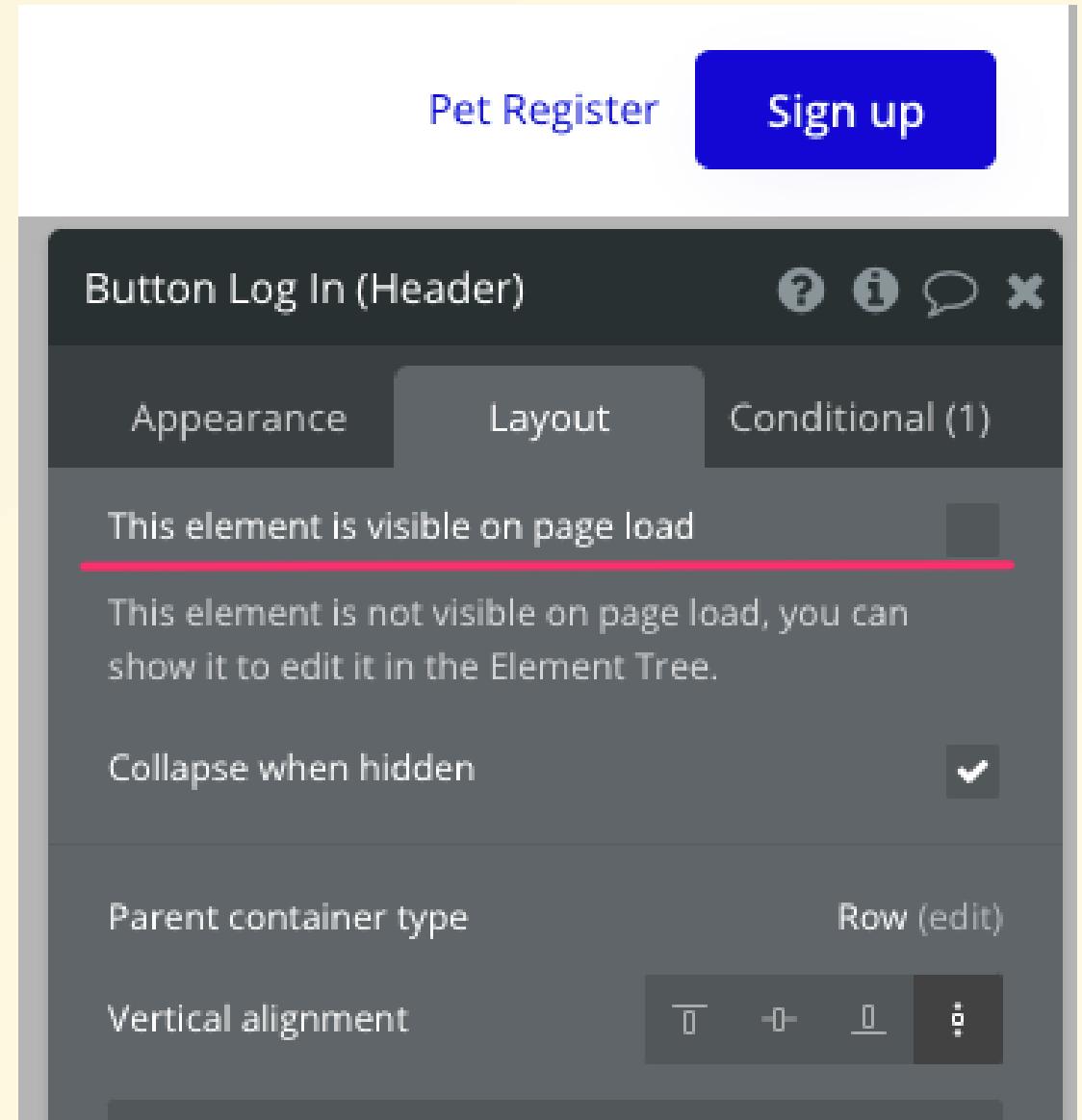


- There are four things to do here
 1. display the login button only when the user is "not logged in"
 2. make the login button go to the login screen (index) when it is pressed
 3. change the signup button to a logout button and display it only when the user is already logged in
When the logout button is pressed, the user is logged out, and then moves to the login screen (index).
- First, we will incorporate the control of the login button.

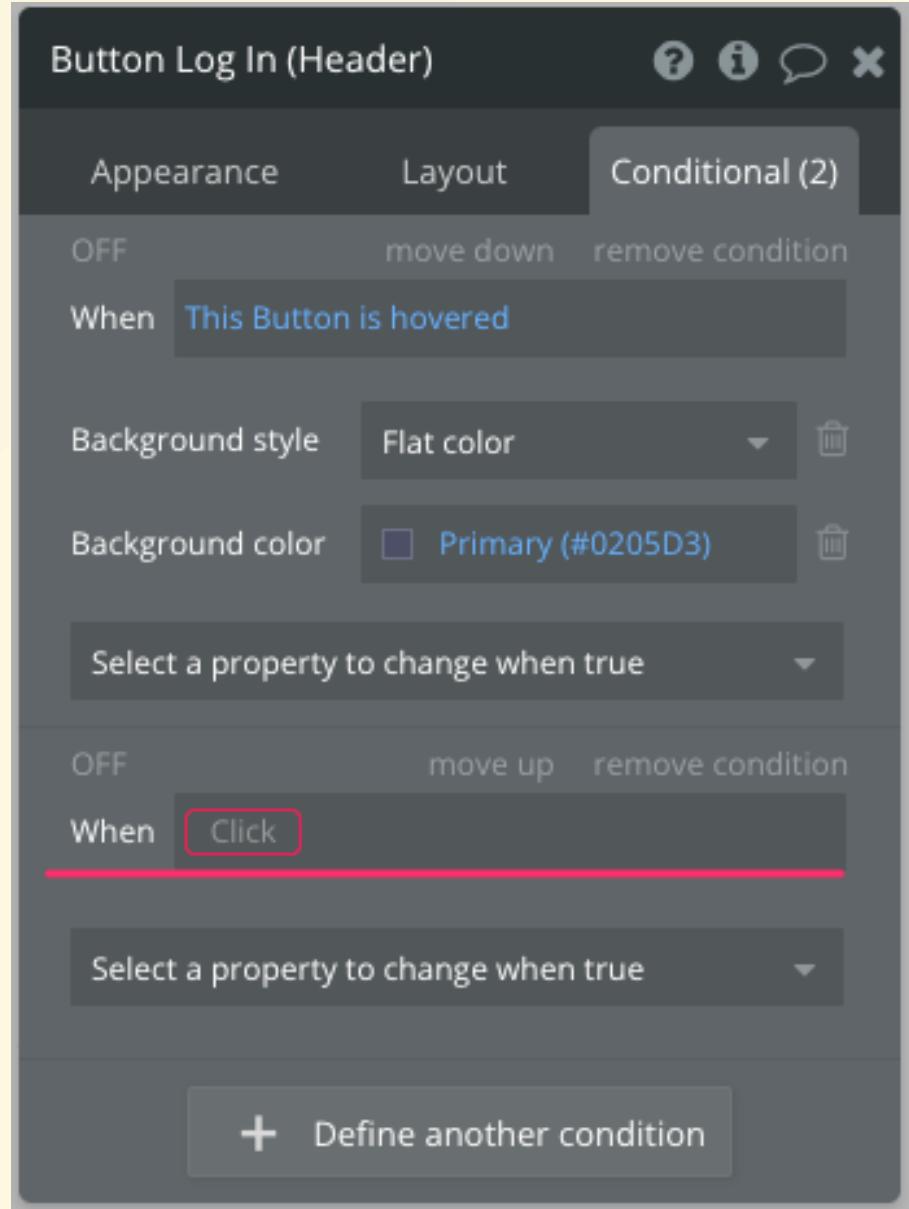
1. the login button is "displayed only when the user is not logged in".

- The control image is as follows
 - The login button is hidden.
 - If the user is currently not logged in under the Conditional condition, the login button is displayed.

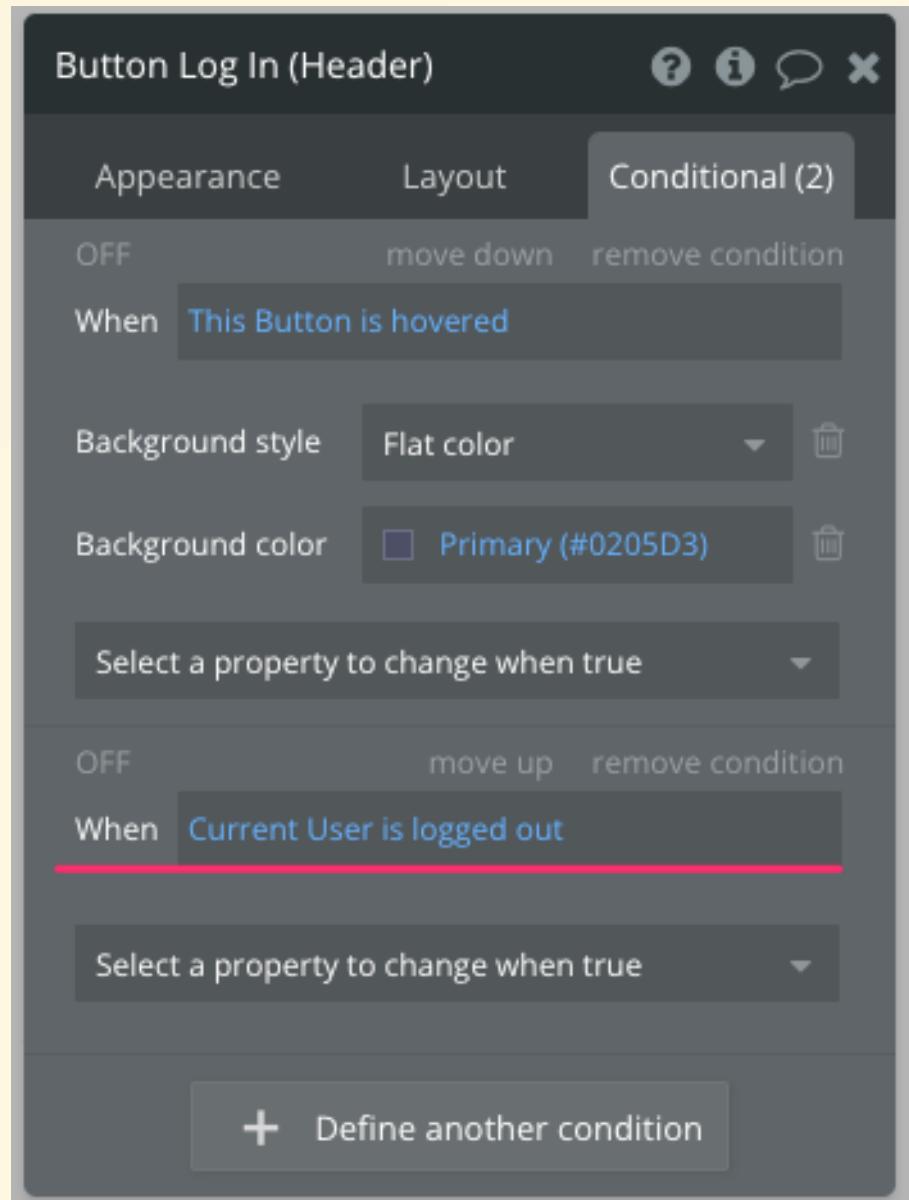
- First, display the Edit Login Button dialog
- On the Layout tab, uncheck `This element is visible on page load
- This will make this element invisible on page load.



- Next, control by login status
- From the Conditional tab, click 'Define another condition'
- Set the When condition to "The user is currently not logged in".



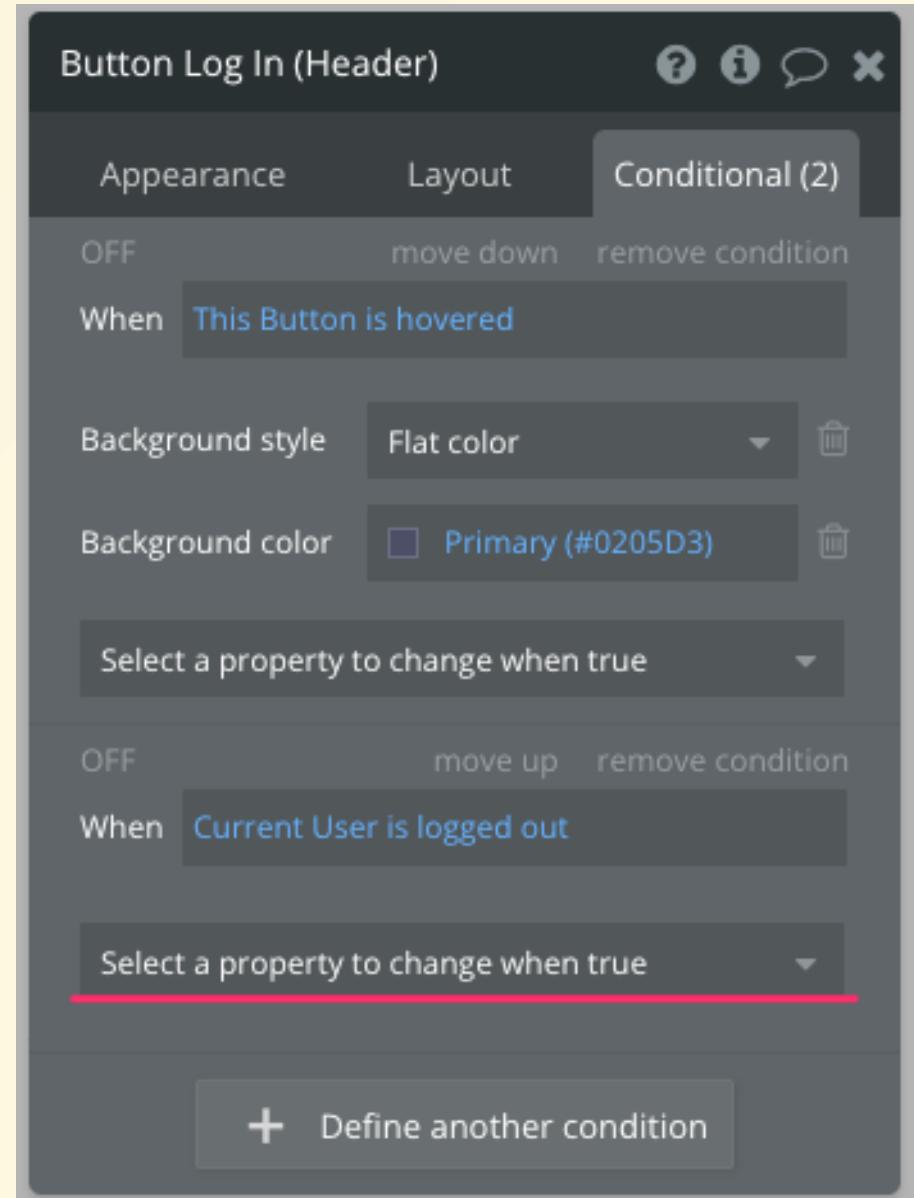
- It looks like this.
 - "Current User is logged out"



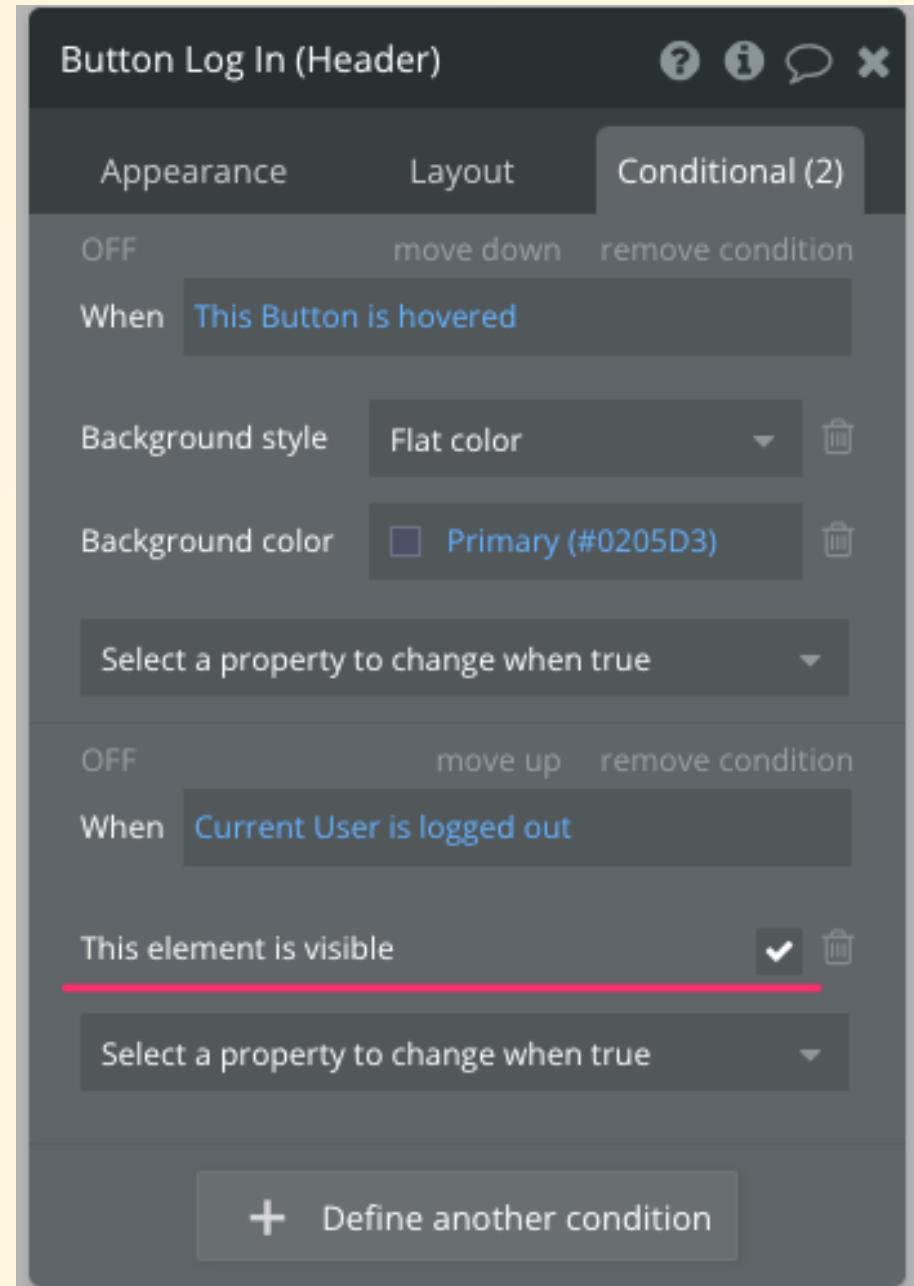
- Then, select the process to be performed when the When condition is met with Select a property to change when true

!

- Do you know what to select?



- "This element is visible"
- Select this and check the box.
- Now you have the following ready to go
 - The login button should be hidden.
 - Show the login button if the user is currently not logged in under the Conditional condition

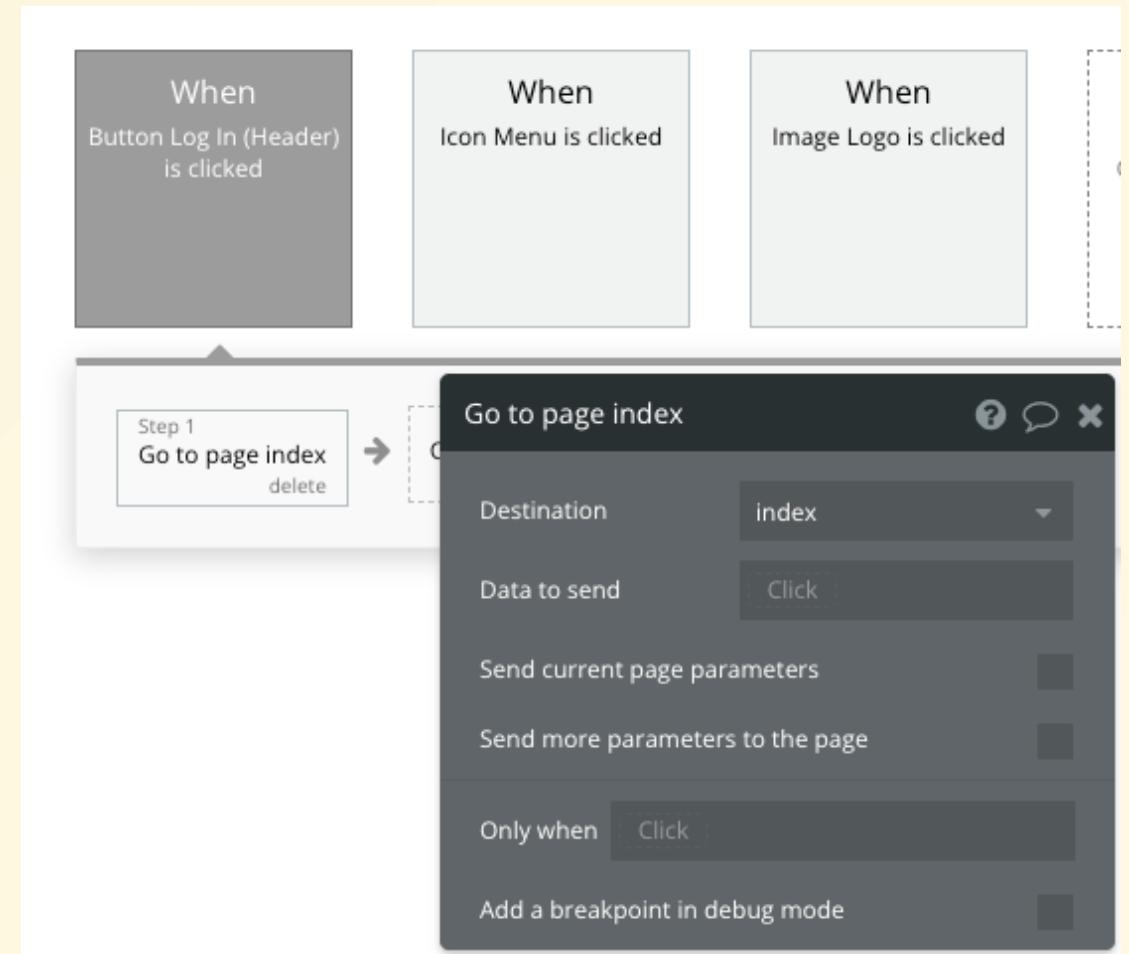


2. Pressing the "Log in" button takes the user to the login screen (index).

- Next, let's set up a workflow that takes you to the login screen when you press the "Log in" button.
- It's just a simple screen transition, so let's set it up!



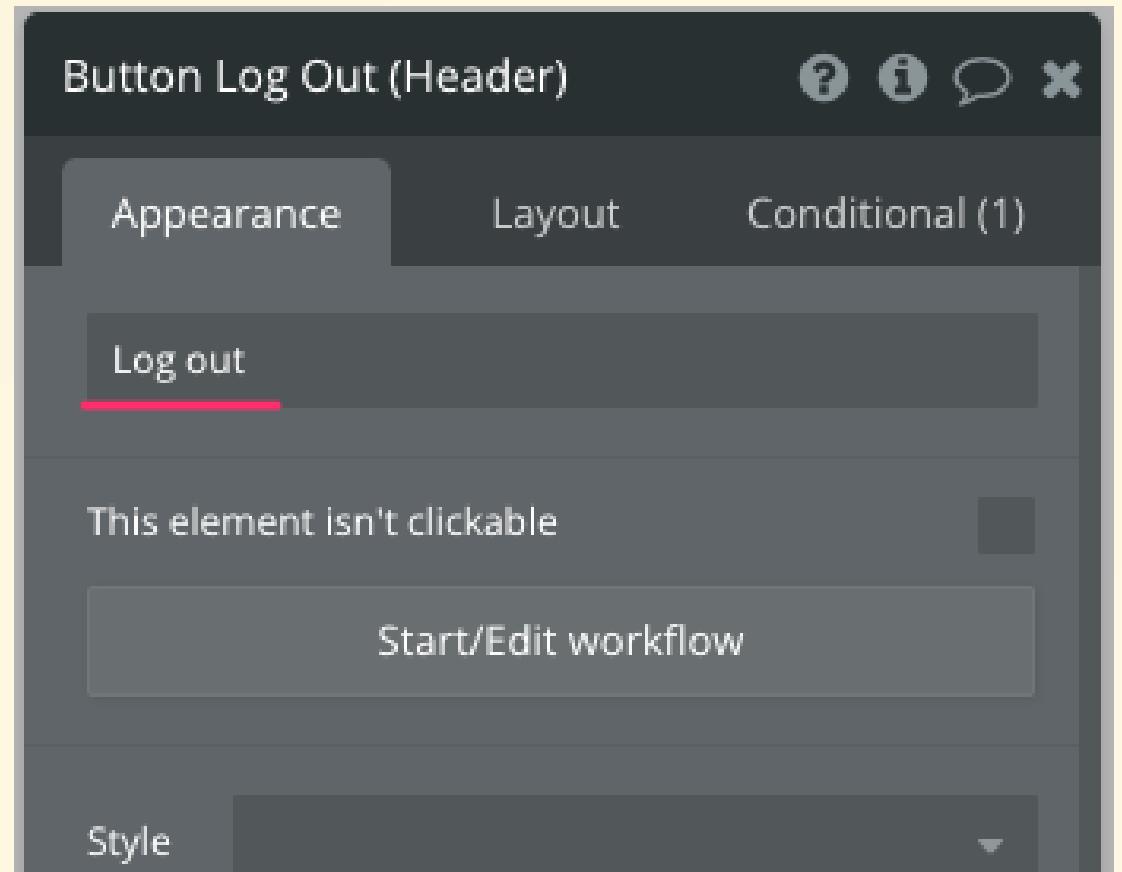
- The workflow to set up is Go to page
- Select the index of the login screen as the Destination and you are good to go.



- This completes the control of the login button.
- Next, let's control the logout button

3. Change the "Sign up" button to a "Log out" button and display it only when the user is in the "Logged in" state

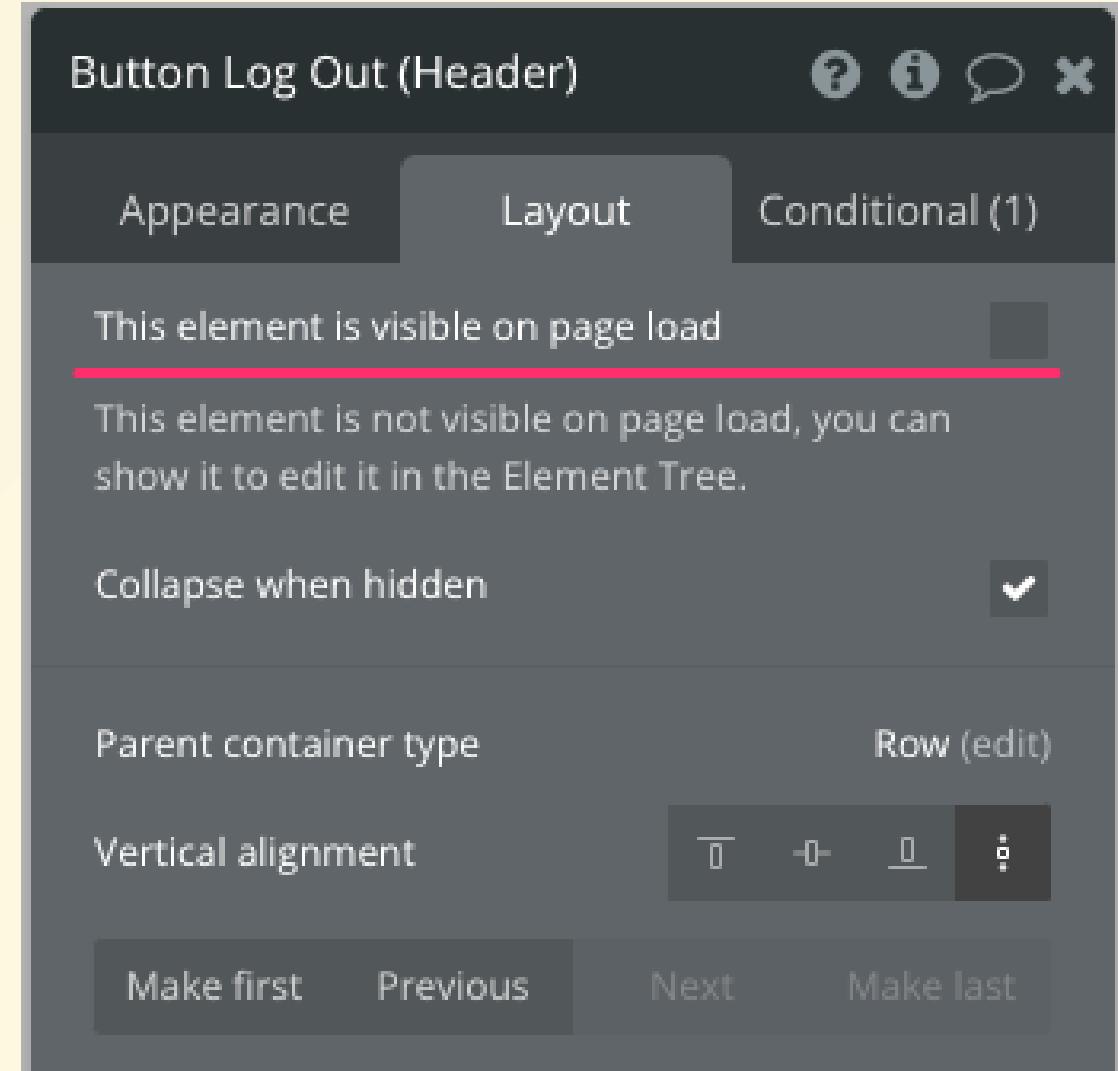
- First, change the label of the "Sign up" button to "Log out"



- And the controls for showing and hiding are almost the same as for the login button mentioned earlier
 - The Logout button should be hidden.
 - If the Conditional condition is that the user is currently logged in, display the login button
- Let's set it up with reference to the previous section



- First, display the Edit Login Button dialog
- On the Layout tab, uncheck `This element is visible on page load
- This will make this element invisible on page load.



- And here is what the Conditional looks like
- The condition for When is "Current User is logged in".
- Now the following conditions are ready
 - The logout button is hidden.
 - If the Conditional condition is "Current User is logged in", the logout button is displayed

Button Log Out (Header)

Appearance Layout Conditional (2)

OFF move down remove condition

When This Button is hovered

Background color Primary (#0205D3) █

Select a property to change when true

OFF move up remove condition

When Current User is logged in

This element is visible █

Select a property to change when true

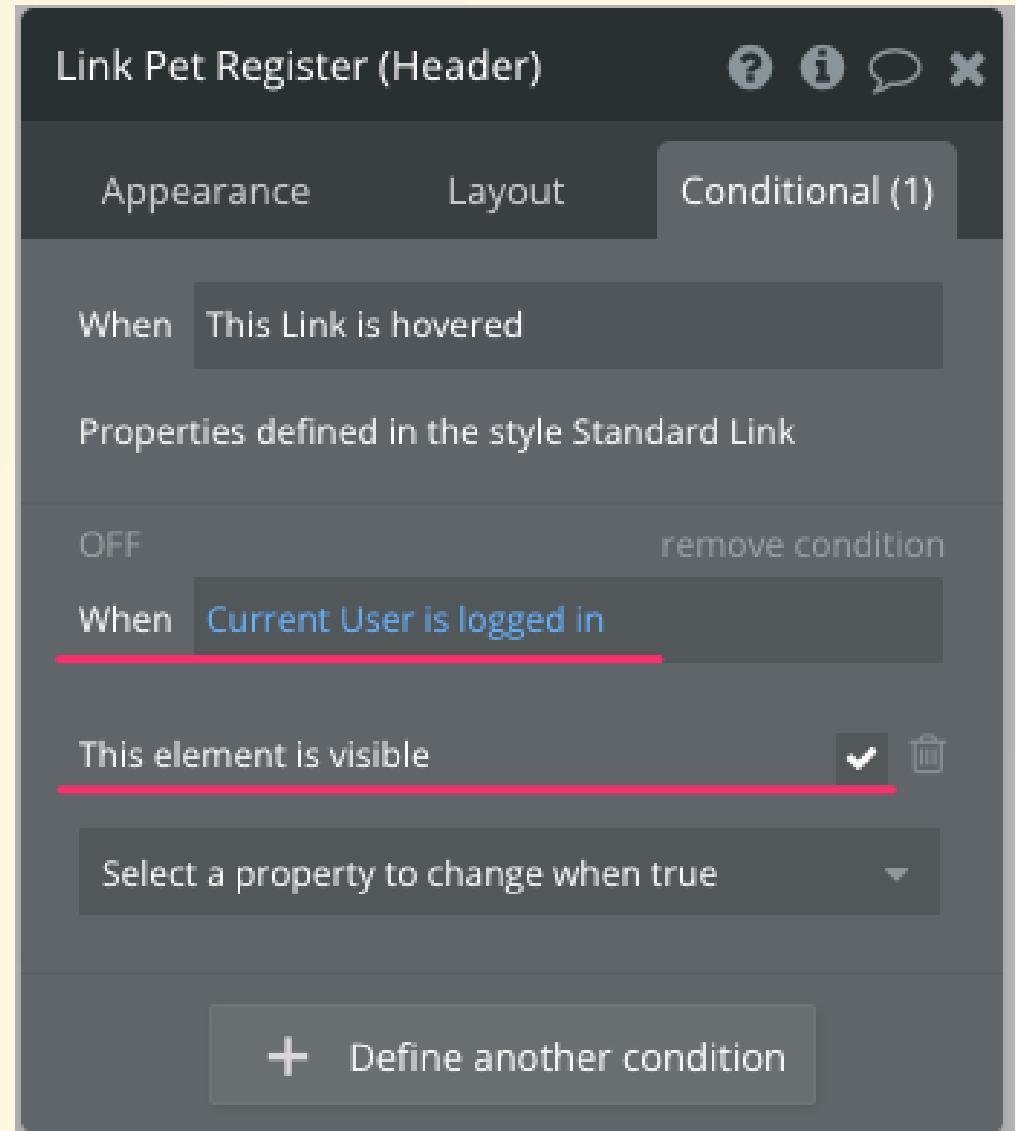
+ Define another condition

- In fact, there is one more place where this same Conditional condition is set to control the display of elements only when the user is currently logged in
- Can you guess where it is?

- Here is the link to the first "Pet Register" we added
- Pet data is data associated with a logged-in user.
- Therefore, we should also set the "Pet Register" link to "Conditional" so that the user can go to the pet registration page only when he/she is logged in.

- You can tell because it's the same control as the logout button I mentioned earlier. 😊

- It looks like this. 



4. When the "Log out" button is pressed, the user is put in a logged out state and then transferred to the login screen (index)

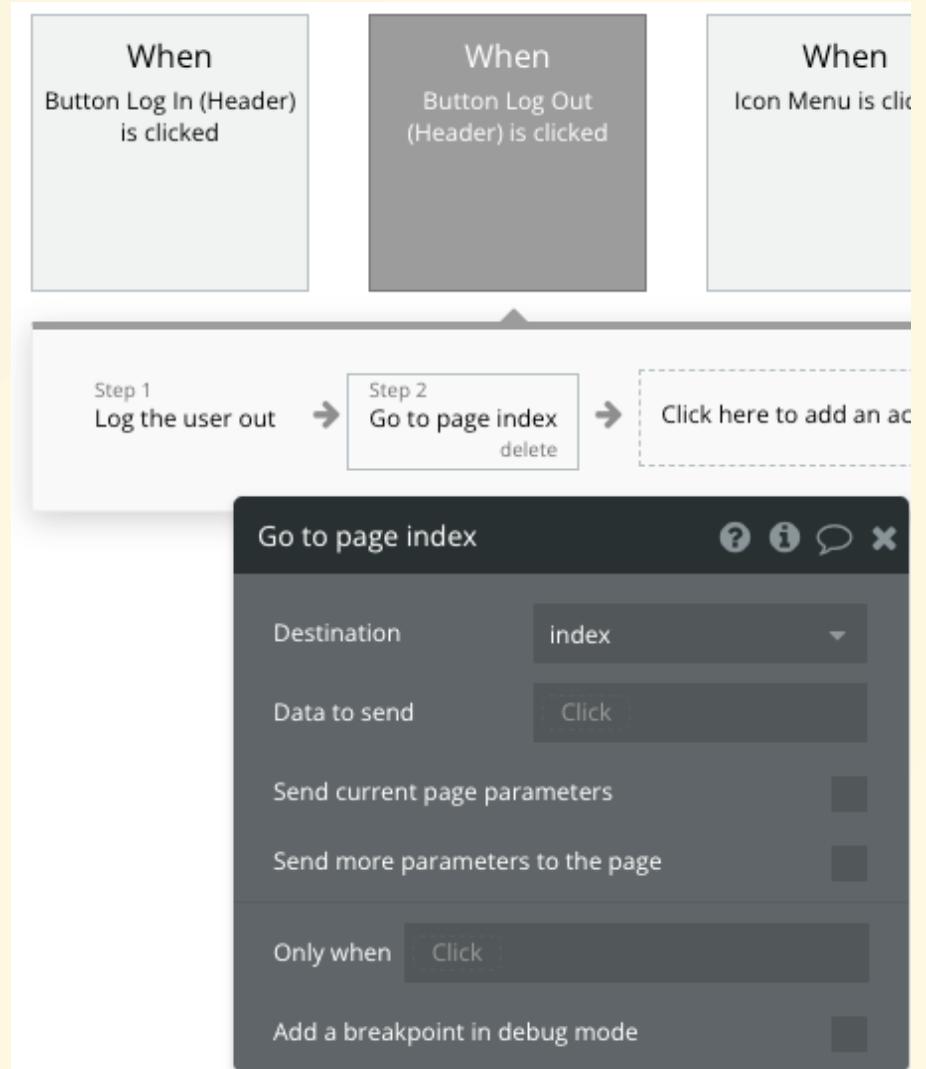
- Finally, set up the workflow when the "Log out" button is pressed.
- Let's set up the workflow when the "Log out" button is pressed.

- The workflow hint for logout status is one of the actions in "Account".

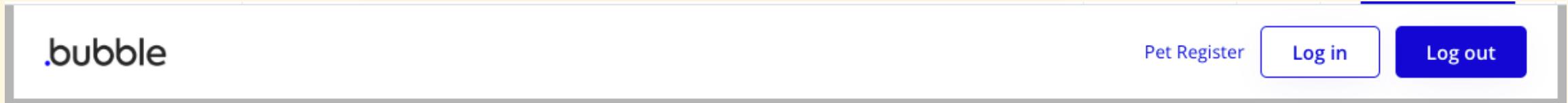
Search for an action...	
 Account	Sign the user up
 Navigation	Log the user in
 Data (Things)	Signup/login with a social network
 Email	Log the user out
 Payment	Update the user's credentials
 Analytics	Make changes to current user
 Element Actions	Send confirmation email
 Plugins	Send password reset email
 Custom Events	Send magic login link
	Create an account for someone else
	Check password for the current user
	Assign a temp password to a user
	Change the email for another user
	Log out other user's sessions



- The action to log out is "Log the user out" in "Account".
- After that, set the action for the screen transition and you are good to go.



- This completes the configuration for the header component.



What we did

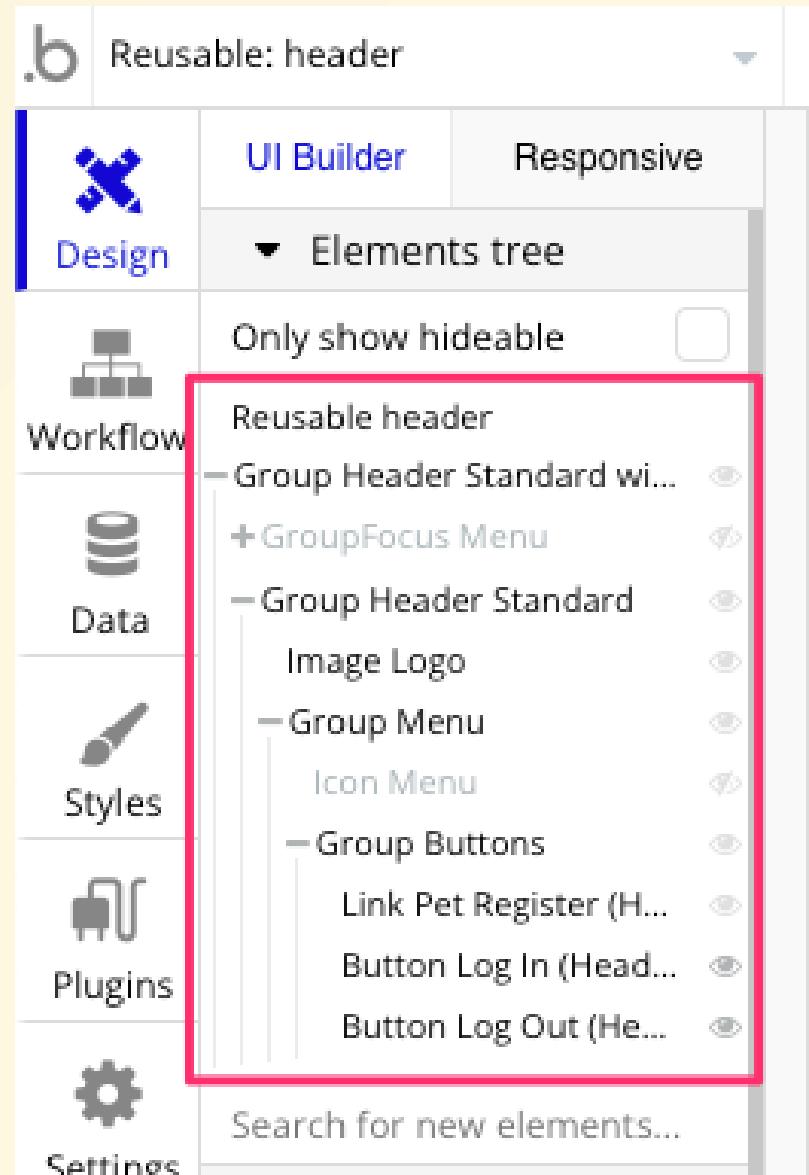
- Link to pet registration screen
- Control Login / Logout buttons based on login status
 1. the login button is displayed only when the user is "not logged in"
 2. move to the login screen (index) when the login button is pressed
 3. change the signup button to the logout button and display it "only when you are already logged in" 4.
 4. when the logout button is pressed, the user is logged out and then moves to the login screen (index).

By the way...

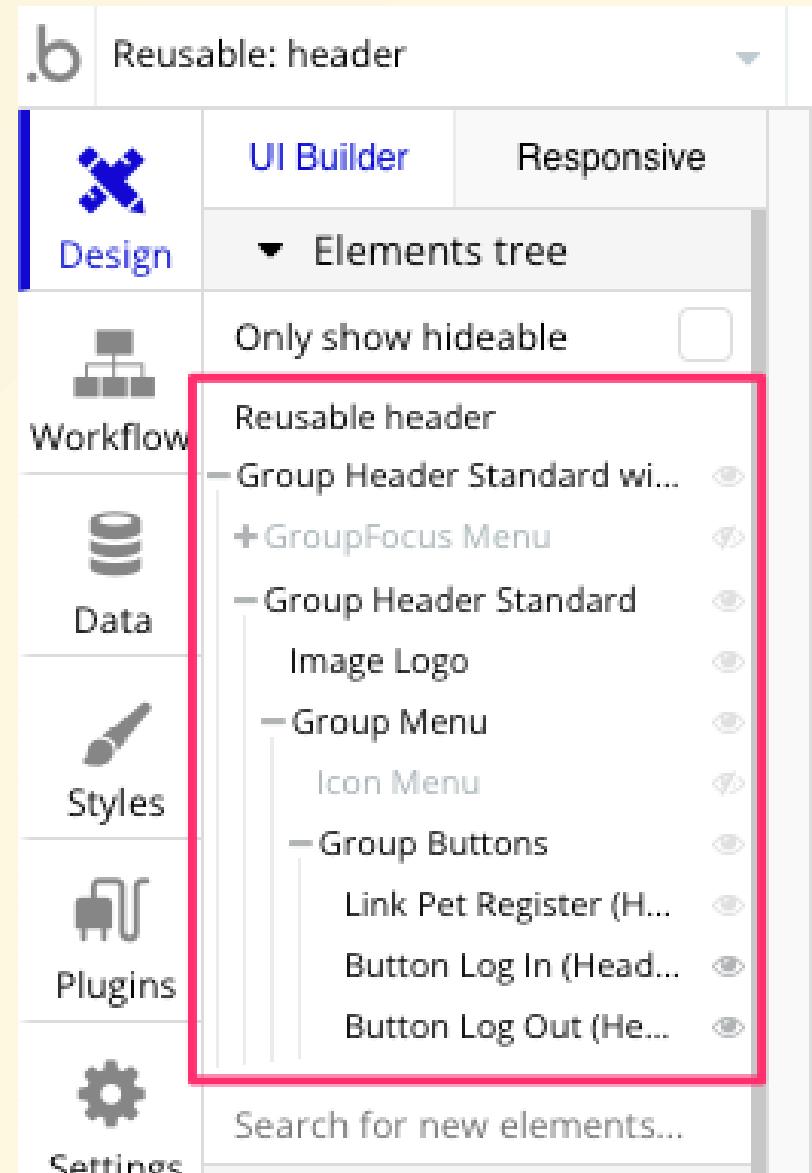
- I assume that the "Log in" and "Log out" buttons are not visible in your header part.
- This is because they are initially hidden in the settings, but there is a way to display the elements even in that state.



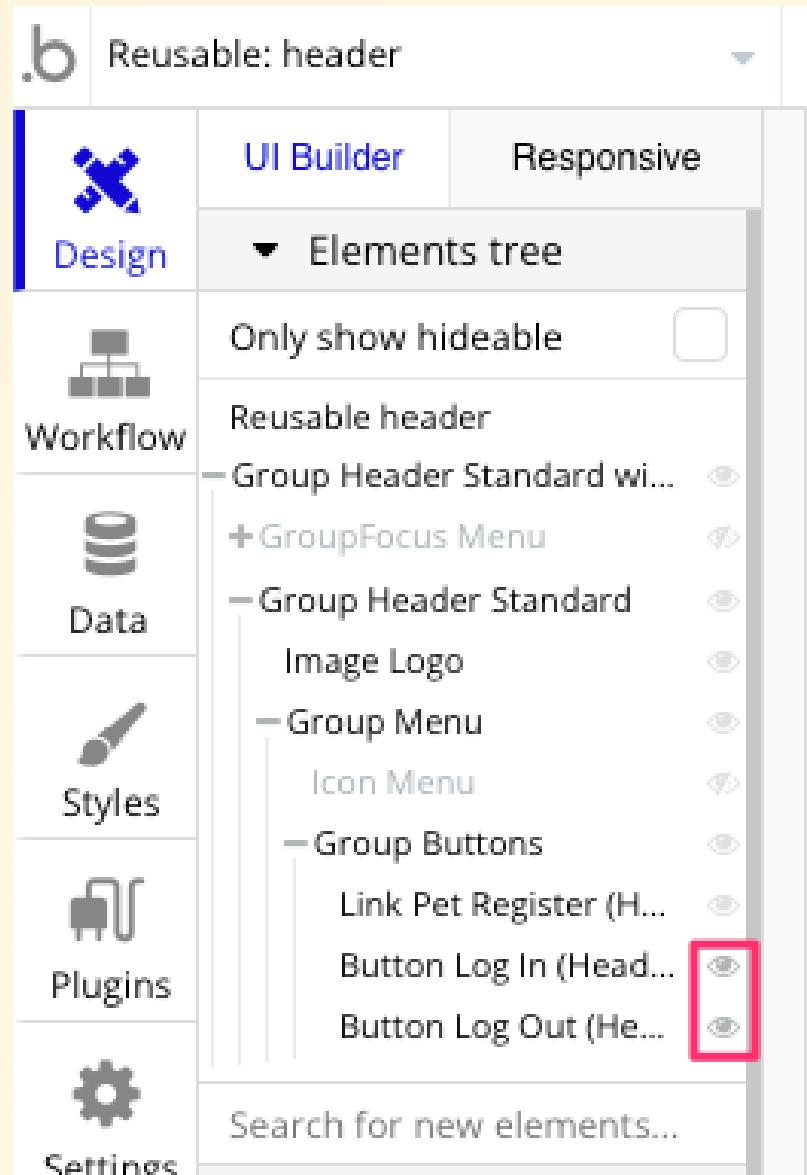
- Open the **Elements tree** from the left panel and press **+** on the objects in it to display the elements



- The elements shown here are all the elements contained in the screen (common parts) that is currently displayed.
- The order of display also has a meaning: the elements displayed at the bottom are those at the top of the screen.



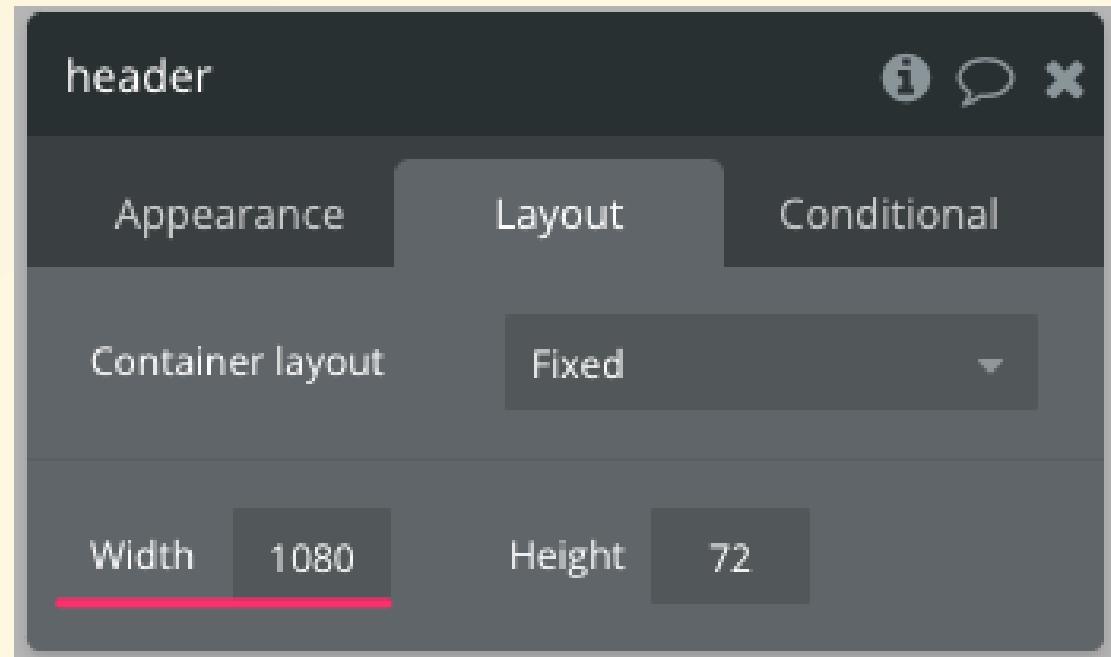
- And among the elements at the very front (bottom) of the panel, the "eye" icon to the right of the "Log in" and "Log out" (Sign in) buttons.
- This has an ON and OFF status, with ON being displayed on the right panel and OFF being hidden on the right panel.



- This time, if you turn off the "eye" icon on the "Log in" and "Log out" (Sign in) buttons and then turn them on again, the buttons will appear on the right panel as well!
- (Note that this is a display / non-display control on the right panel and has nothing to do with the actual display / non-display on the application.



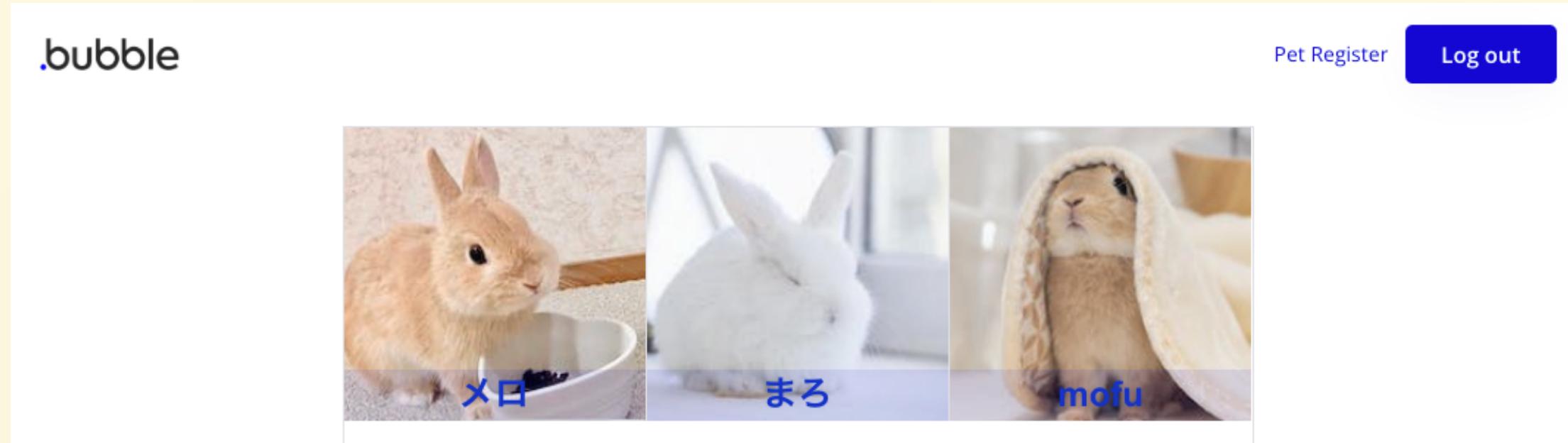
- Finally, to match the width of the current header component with the width of the screen, select the following element from the **Elements tree** and change the value of **Width** to "1080".
 - "Reusable header"
 - "Group Header Standard with Menu"



- If you can no longer see the "Pet Register" link or the login button because you have changed the `Width`, follow the same procedure as before and click on the "eye" icon of the button or other element in the `Elements tree` and turn on the display.

Let's incorporate it into each screen.

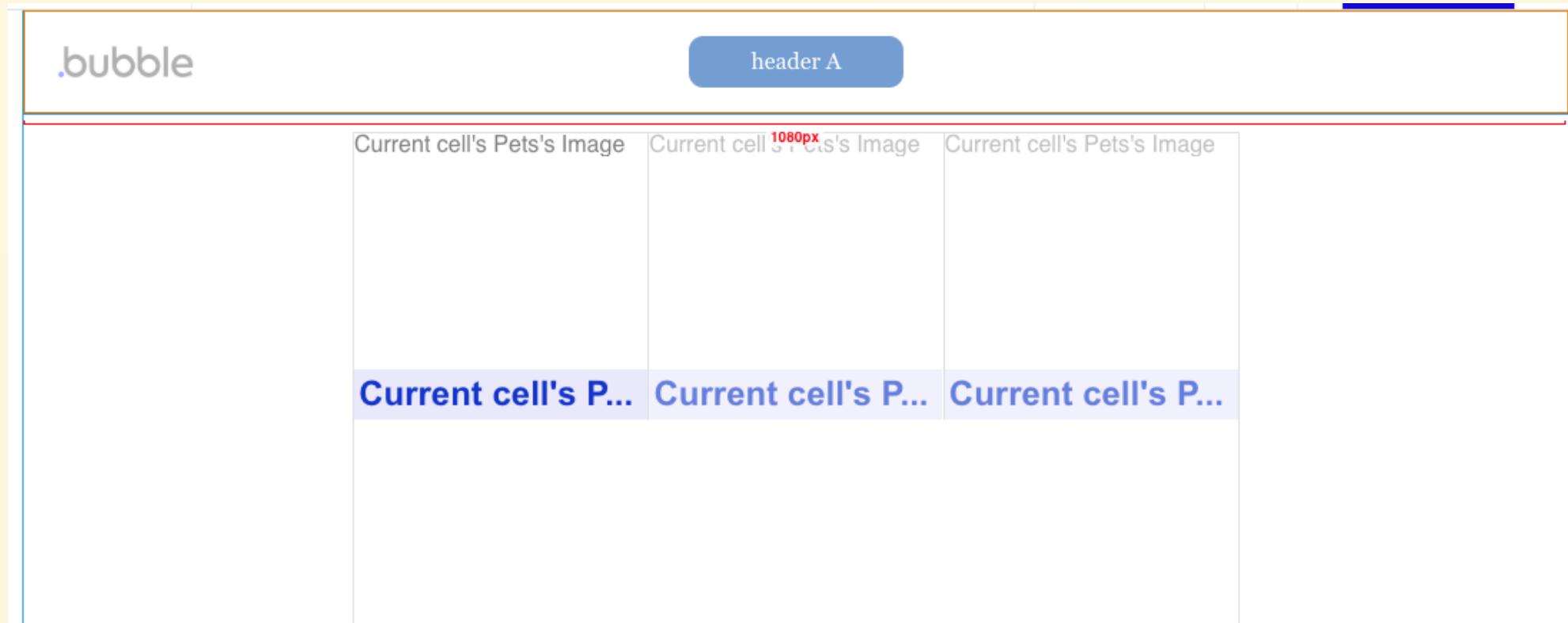
- Now that the header component is ready, let's first incorporate it into the Pets List screen.



- Open "pet_list" and remove the "PET REGISTER" link you placed first



- From the left panel, click on "header" in **Reusable elements** and drag it to the top of the right panel
- Place it at the top of the screen!



Now let's preview it. 

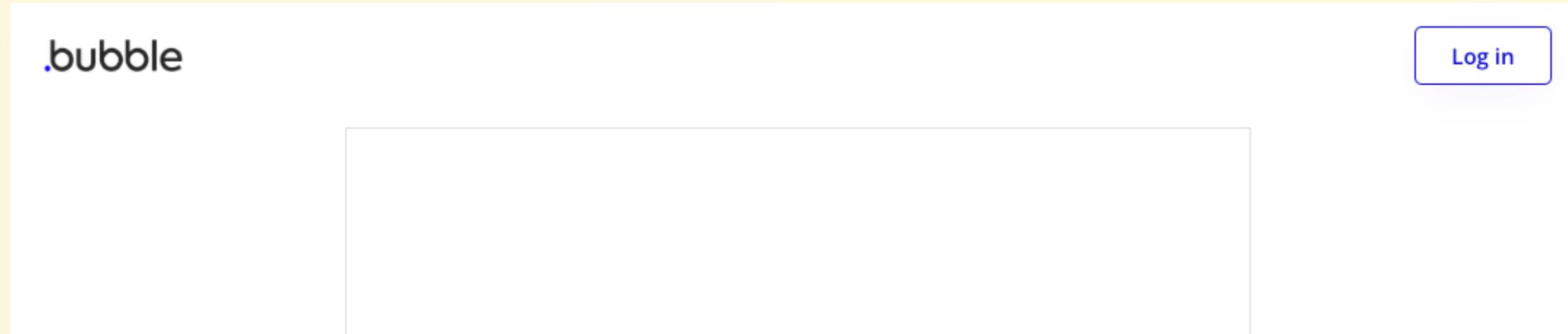
If you are logged in

- The link for pet registration should be displayed
- Logout button should be displayed
- Logout button must be pressed to go to the sign-up page



If you are in a logged out state

- The link for pet registration is not displayed.
- Login button should be displayed.
- When the login button is clicked, it should be redirected to the sign-up page.



- If all goes well, we would like to incorporate the same workflow into the rest of the screen, but before we do that, we need to set up one more workflow.
- The workflow is to move to the Pets List screen upon successful sign-up/login.

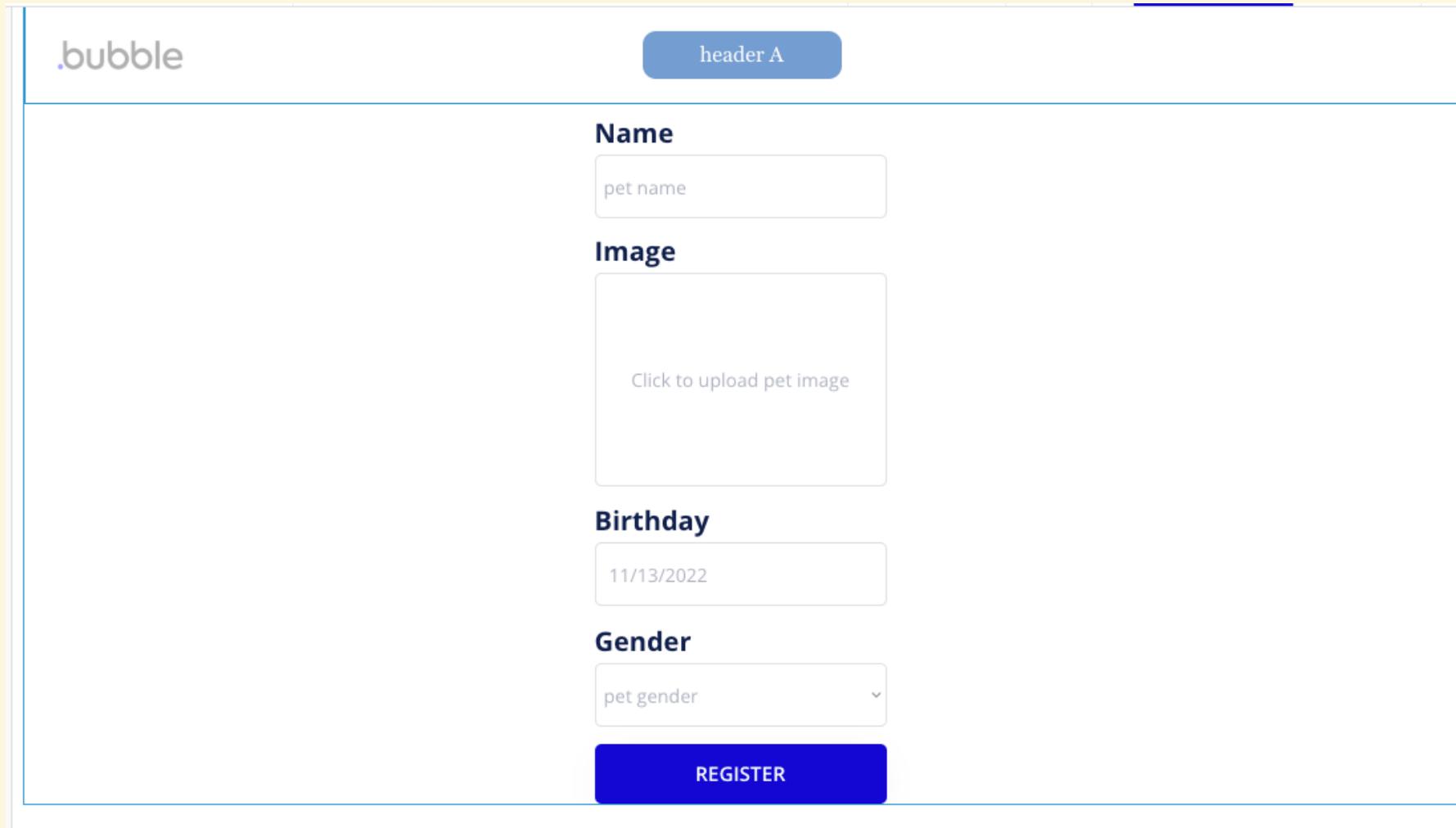
Let's Try!! 🔥



- Well, that's easy enough for all of you right now!
- Just open the index page workflow and add a "Go to page pet_list" action for the following two Whens
 - "When Button Log in is clicked"
 - "When Button Sign up is clicked"

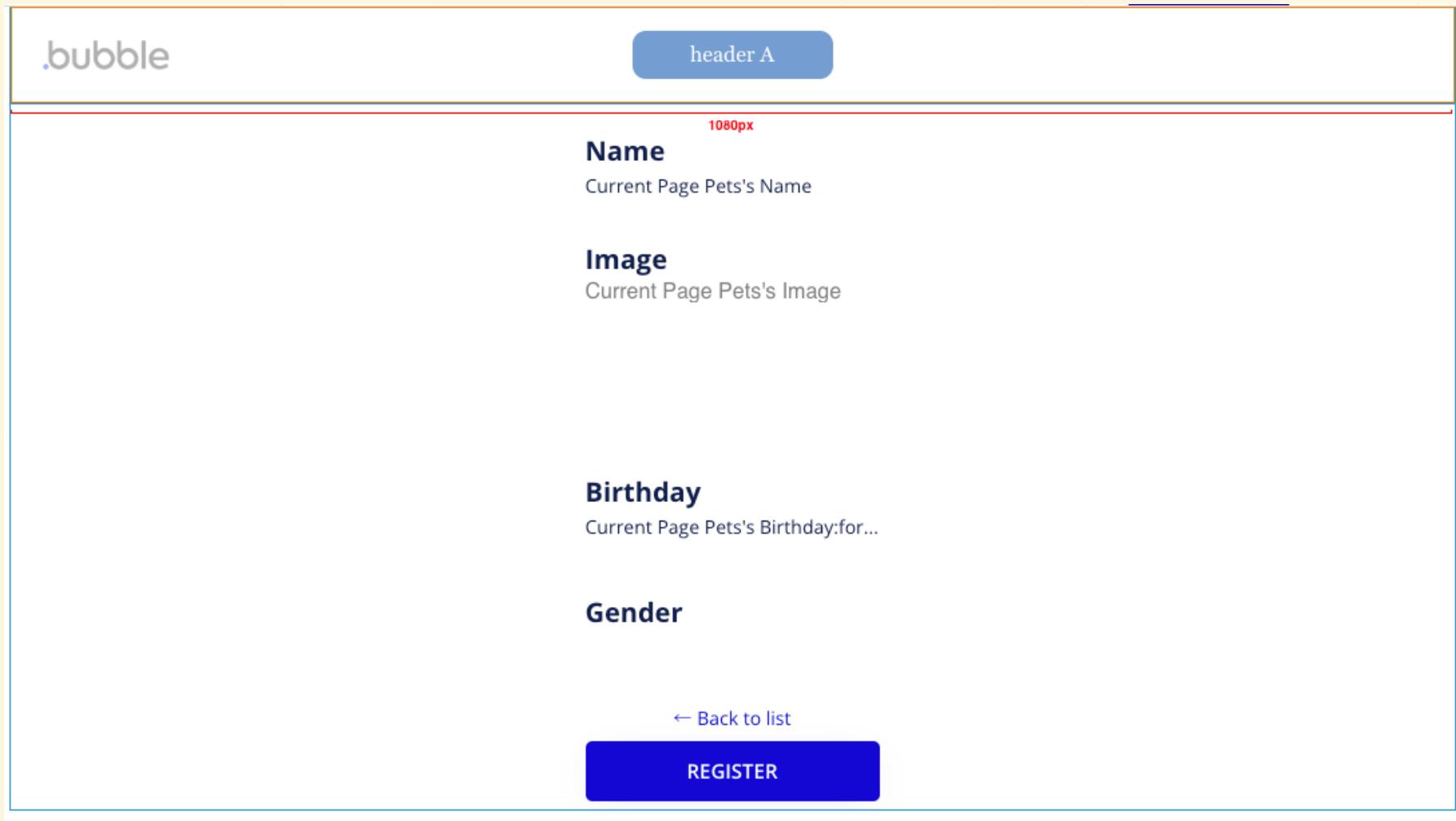
- Preview the page to make sure that the login button and the sign-up button will take you to the Pets List page.
- If the preview is OK, let's place the common header components on the other two screens as well.
 - Pet Registration Screen
 - Pets Detail Screen

- Pet Registration Screen



A wireframe diagram of a Pet Registration Screen. The screen has a header labeled "header A" at the top right. On the left side, there is a logo ".bubble". The main content area contains four input fields: "Name" (text input placeholder "pet name"), "Image" (button placeholder "Click to upload pet image"), "Birthday" (text input placeholder "11/13/2022"), and "Gender" (dropdown menu placeholder "pet gender"). At the bottom center is a large blue button labeled "REGISTER".

- Pet Detail Screen



- 登録画面と詳細画面もプレビューで確認してみましょう

Pet Registration Screen

.bubble

Pet Register Log out

Name
pet name

Image
Click to upload pet image

Birthday
11/13/2022

Gender
pet gender

REGISTER

Pet Detail Screen

.bubble

Pet Register Log out

Name
メロ

Image



Birthday
2022年9月5日

Gender
男の子

[← Back to list](#)

REGISTER

Exercise 5: Separate links for the pet registration page

- If you can afford it, let's try to hide the "Pet Register" link in the header on the pet registration page.
- Hint 
 - "Current page name"
 - The page name of the pet registration screen is "pet_register".

Keep track of your pet's weight!

- Finally, the last one 🔥
- Let's take a look at how to manage your pet's weight.



- First, let's prepare the database Type.
- From the Data tab, enter `PetWeightLogs` as New type, check `Make this data type private by default` and Create.
- Next, we will set up the field.

- Weight of the pet entered from the screen
 - WeightKg: number
- Associate with which pet's weight
 - Pet: Pets

The screenshot shows the Backstage application interface for managing custom data types. The left sidebar has tabs for Design, Workflow, Data (which is selected), Styles, Plugins, Settings, and Logs. The main area shows a navigation bar with 'Page: pet_weight_register', 'Edit', 'Help', 'Saved', and various status icons. Below this is a tab bar with 'Data types', 'Privacy', 'App data', 'Option sets', and 'File manager'. The 'Data types' tab is active, showing a list of existing types: 'Pets' (Publicly visible), 'PetWeightLogs' (Publicly visible, currently selected), and 'User' (Privacy rules applied). A 'New type' section allows creating a new data type with a text input field and a 'Create' button. On the right, under 'Fields for type PetWeightLogs', there is a table listing fields: 'pet' (type 'Pets'), 'WeightKg' (type 'number', highlighted with a red border), 'Creator' (type 'User', 'Built-in field'), 'Modified Date' (type 'date', 'Built-in field'), 'Created Date' (type 'date', 'Built-in field'), and 'Slug' (type 'text', 'Built-in field'). A 'Create a new field' button is at the bottom.

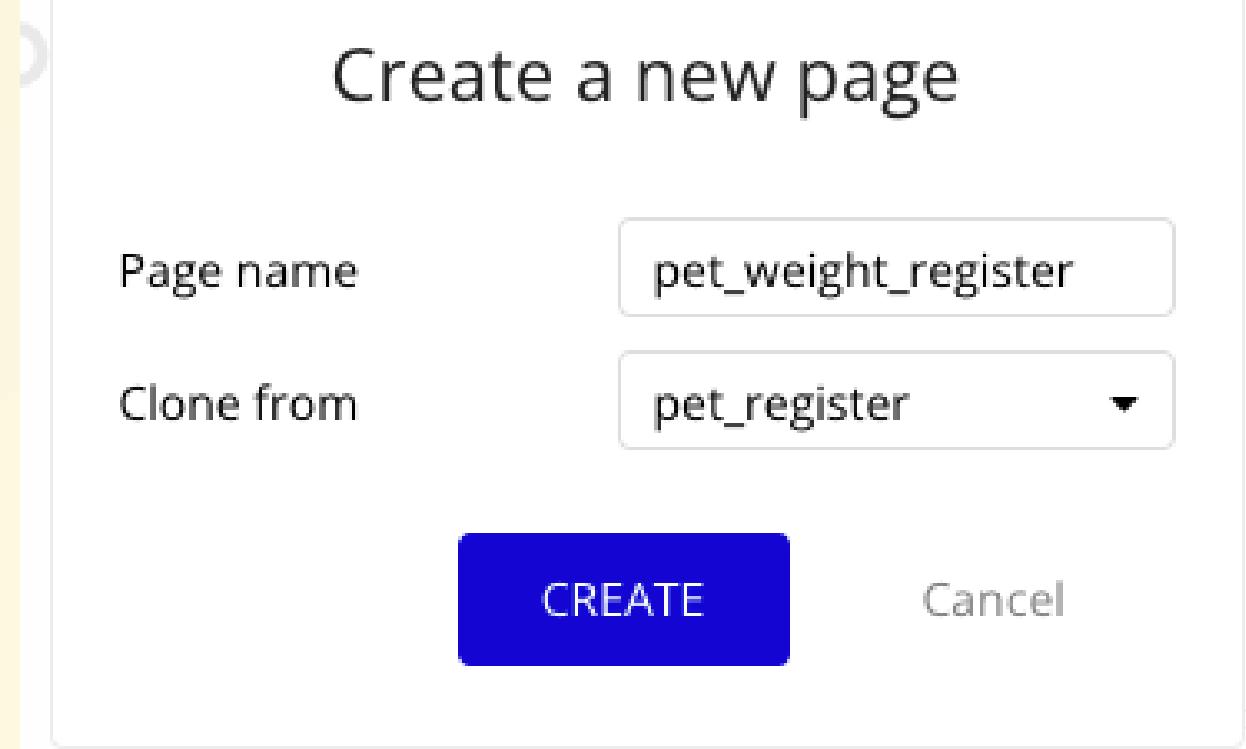
- Now that the database is ready, let's set up the screens!
- First, let's set up a new page for pet weight management.
- From **Add a new page**, select **pet_weight_register** as the page name and **pet_register** as the copy source, and **CREATE**

Create a new page

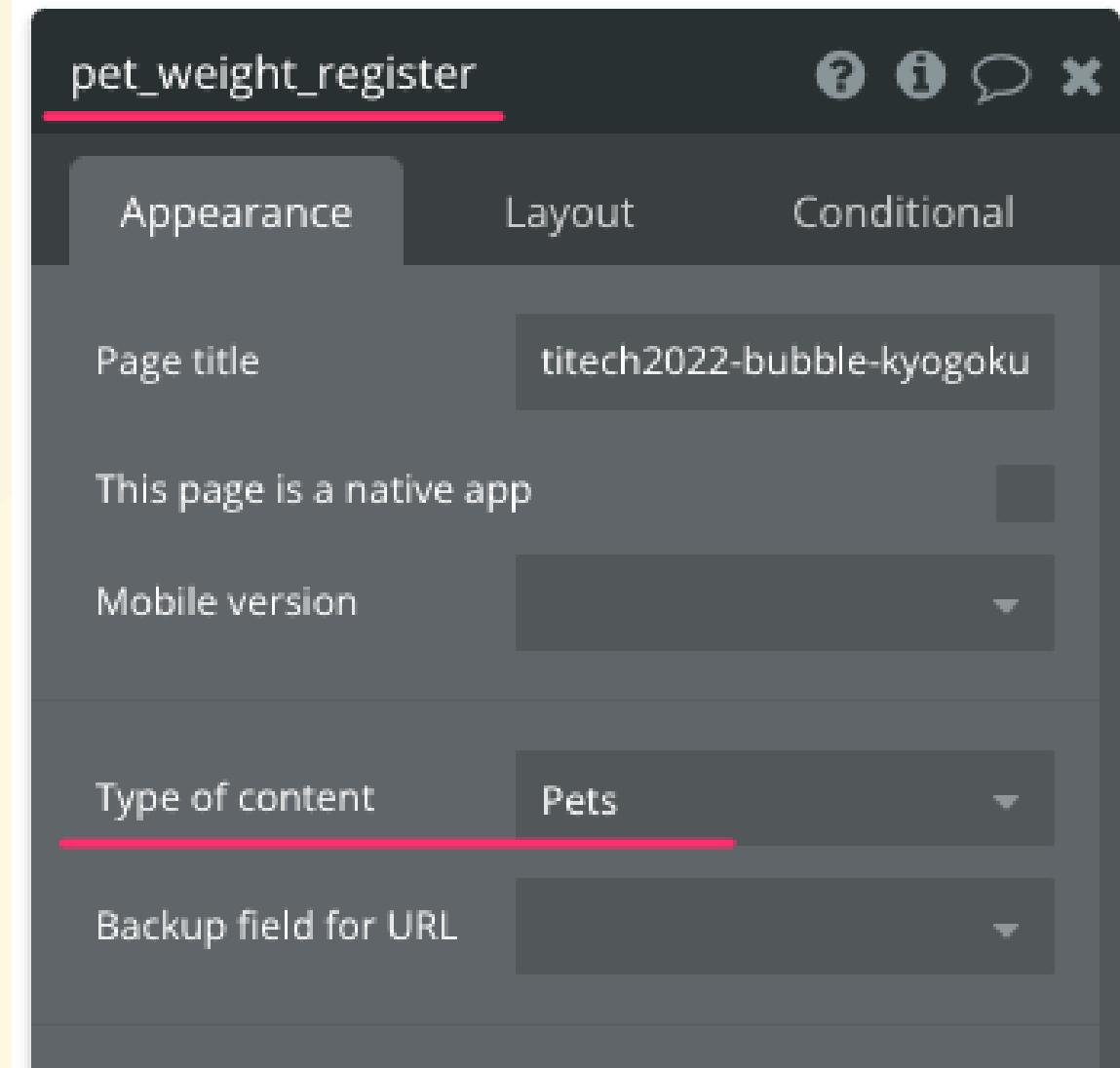
Page name

Clone from ▾

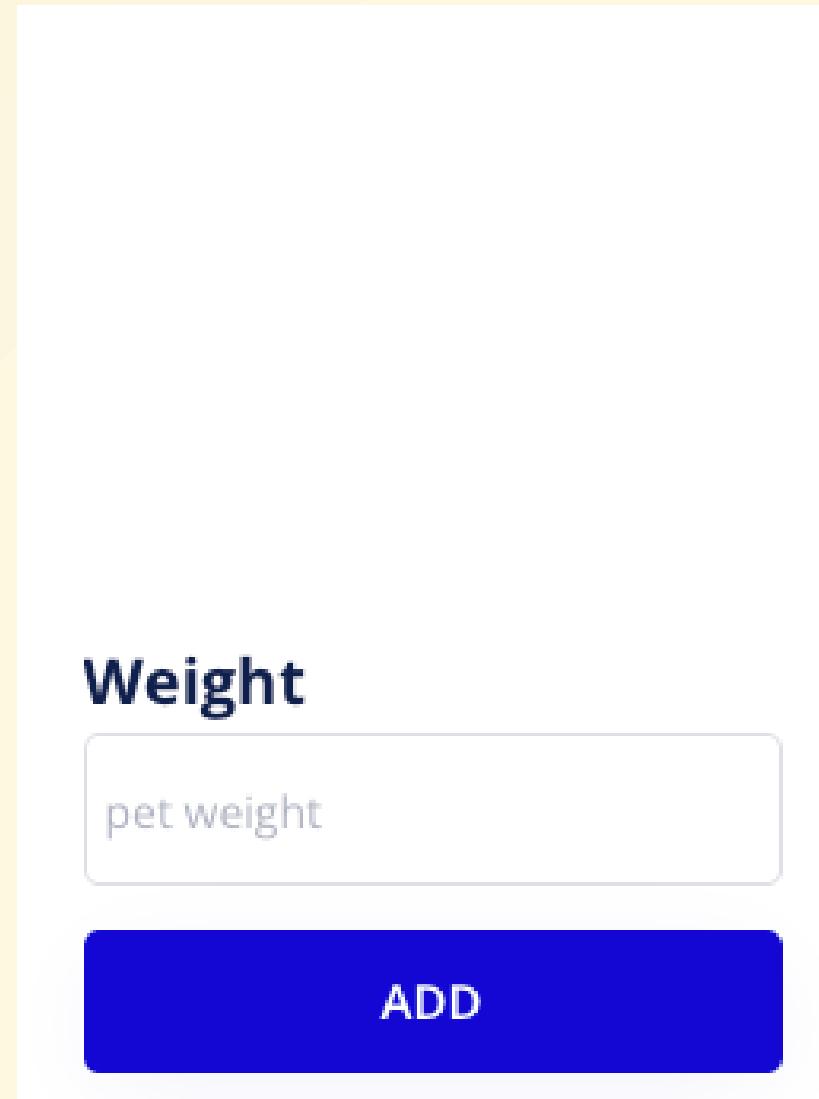
CREATE Cancel



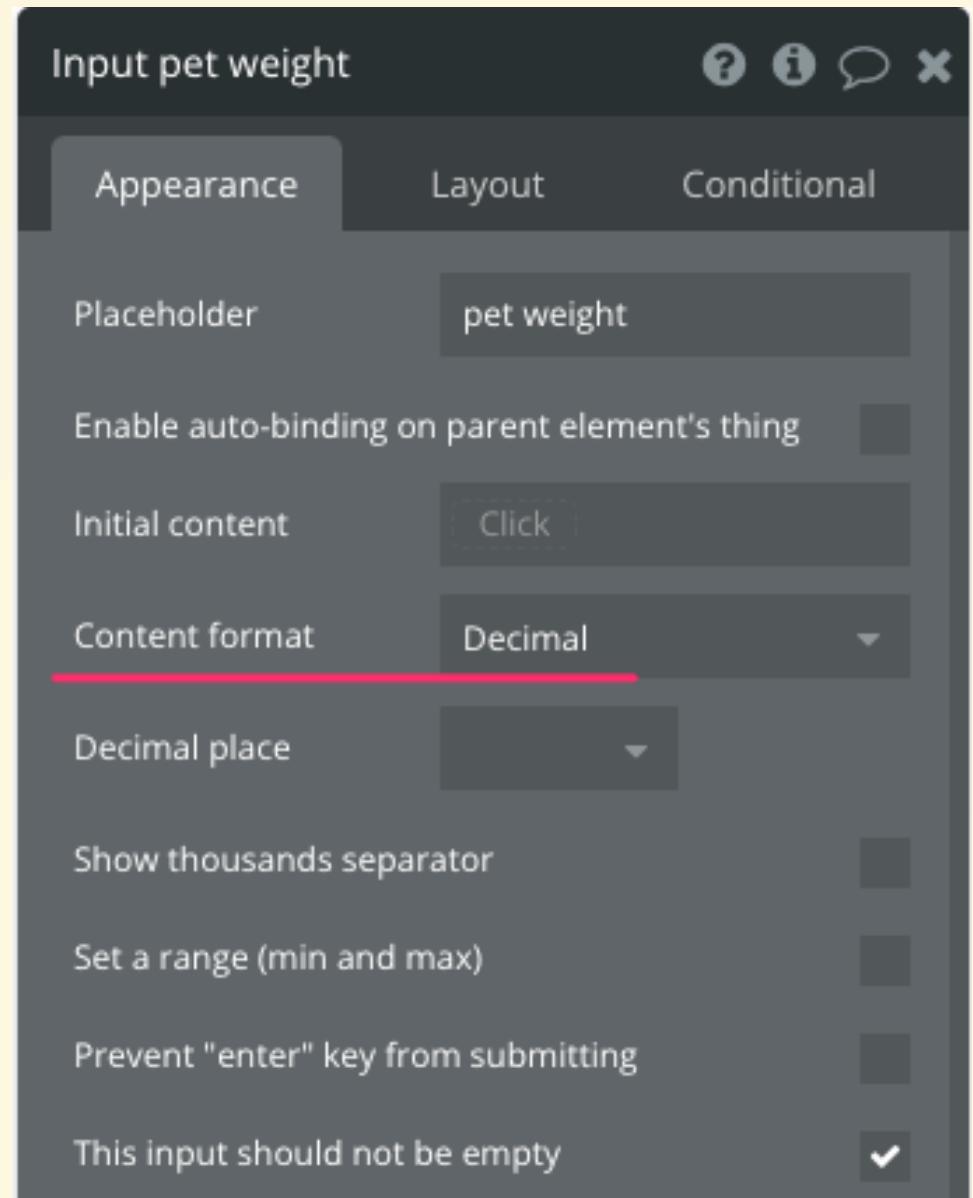
- This screen is also supposed to be accessed from the pet details, so from the advanced settings dialog for the `pet_weight_register` screen, specify "Pets" for the Type of content.



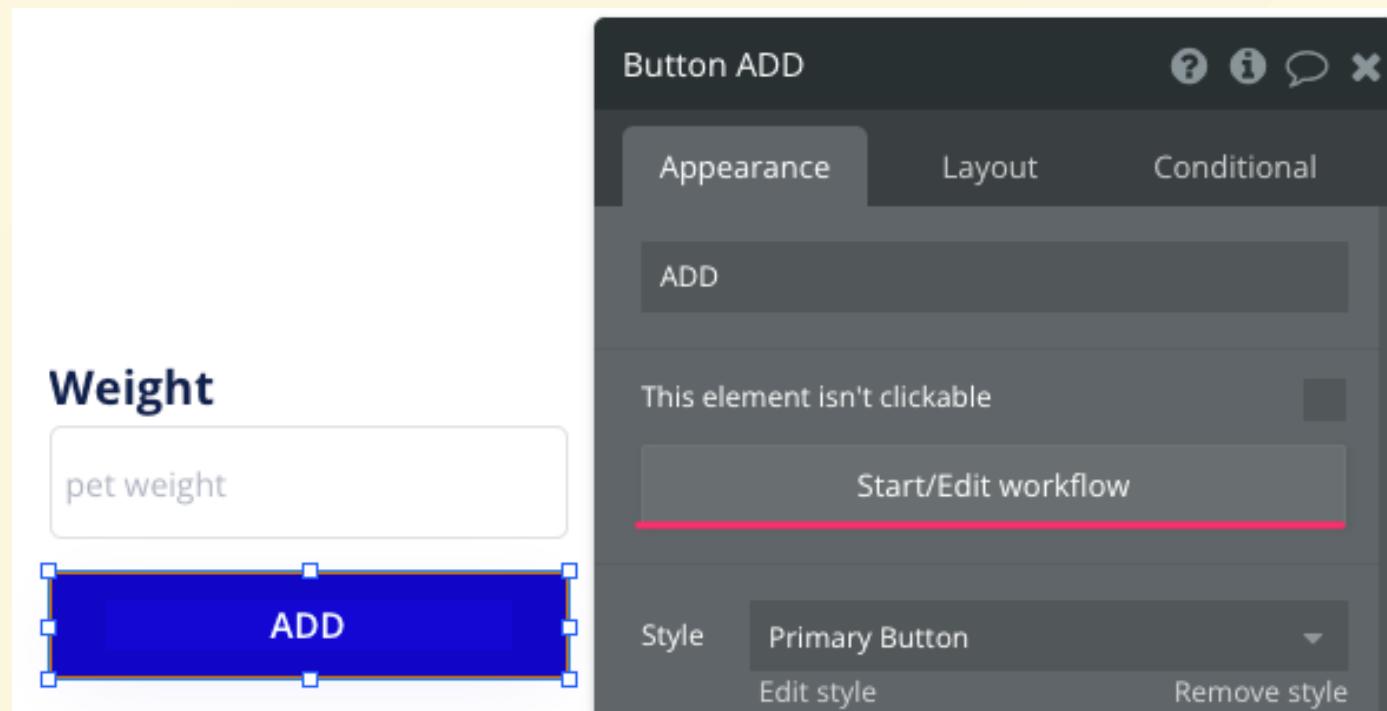
- The elements of the original pet registration screen are still there, so we will place the elements as shown in the reference image while deleting the unnecessary ones.
- However, the graph drawing area is not placed at this time.



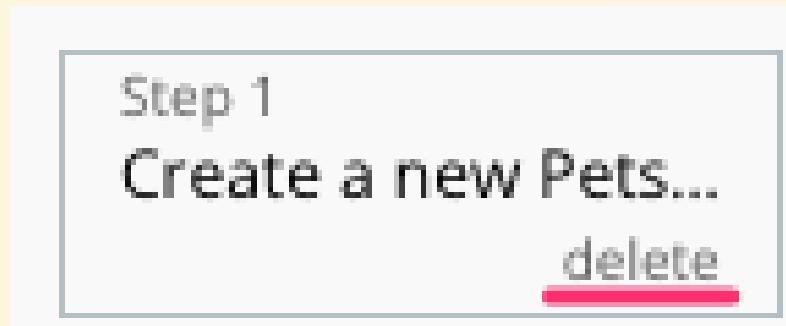
- In this case, we only need to set the weight input element to accept only numbers with decimal points.
- Double-click the pet weight input element and select **Decimal** for **Content format**.
 - By the way, select **Integer** for integer input without decimal point.



- Next, we will set up the workflow for the weight saving button clicks
- Double-click the ADD button and select Start/Edit workflow from the Advanced dialog



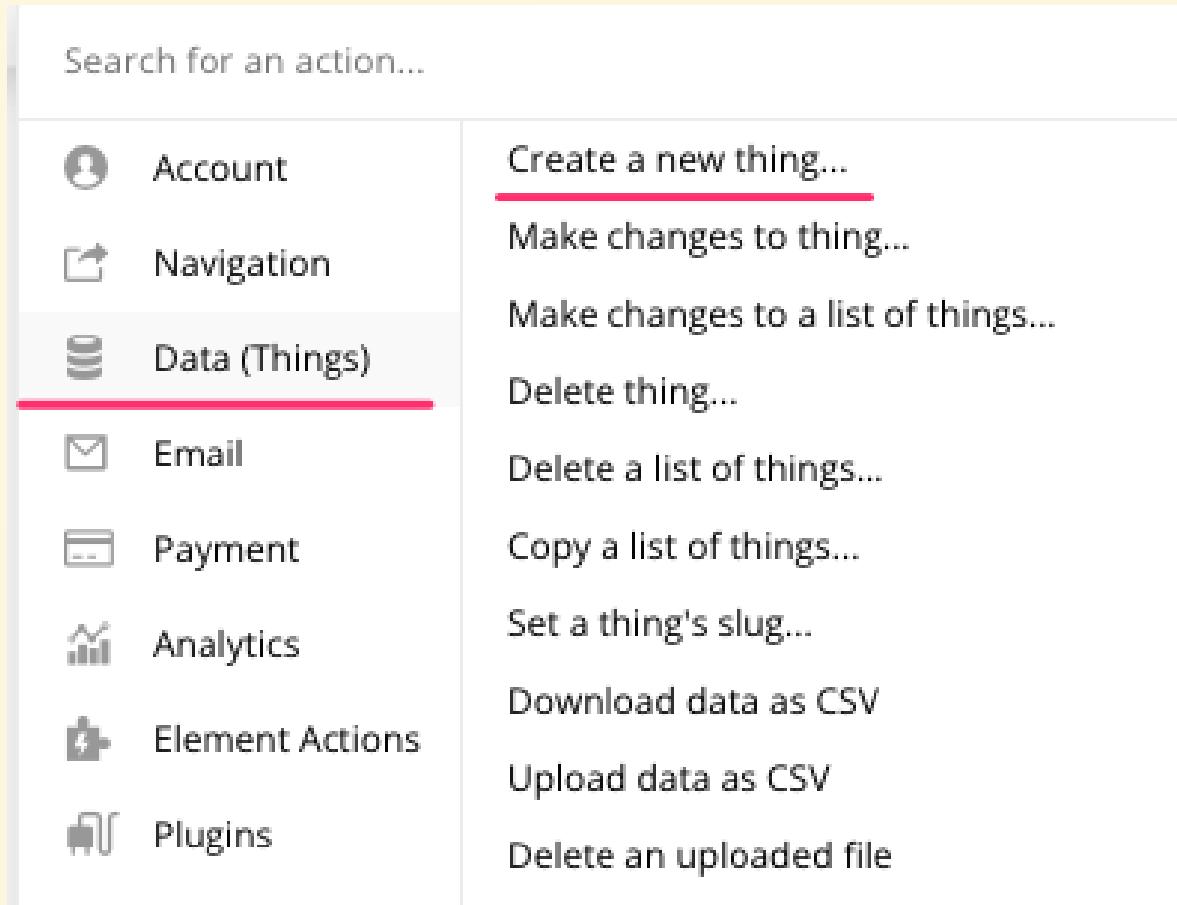
- Then, for `When Button ADD is clicked`, some behaviors are already set!
- This is because we copied `pet_register` when we created the page, so the workflow settings are inherited from that time.
- This time, we will define a new behavior, so we will delete the existing workflow.
- When you select a behavior in Step1 / Step2, you will see `delete` in the lower right corner.



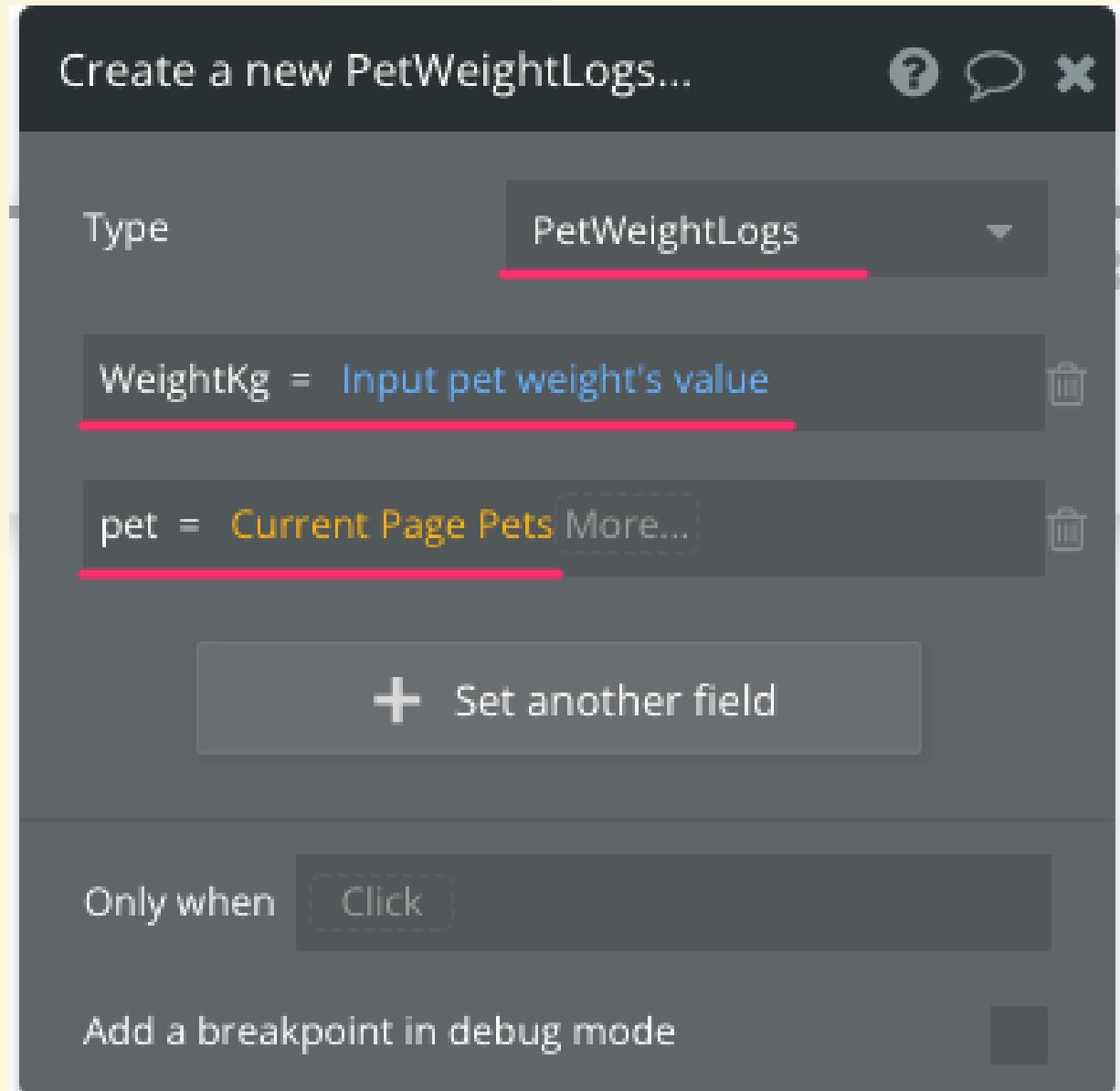
- Now let's set up the workflow.
- The first thing we want to do is "save the weight values entered on the screen to the PetWeightLogs we prepared earlier"!
- You'll want to set this up based on what we've covered in the lecture so far!
- I've included an explanation on the next page, so let's do that first!



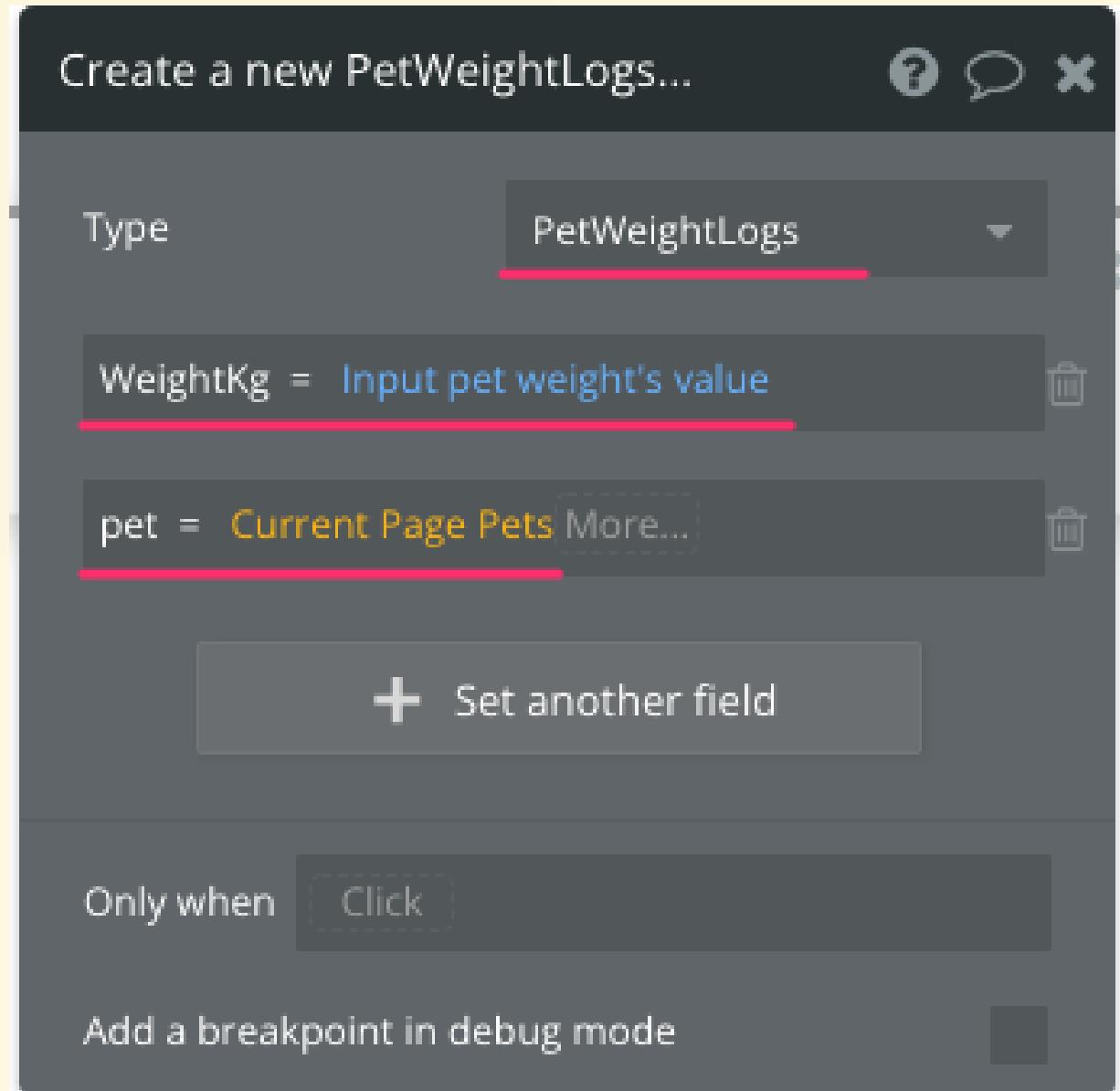
- The first action to set up is **Data(Thing)** --> **Create a new thing...**



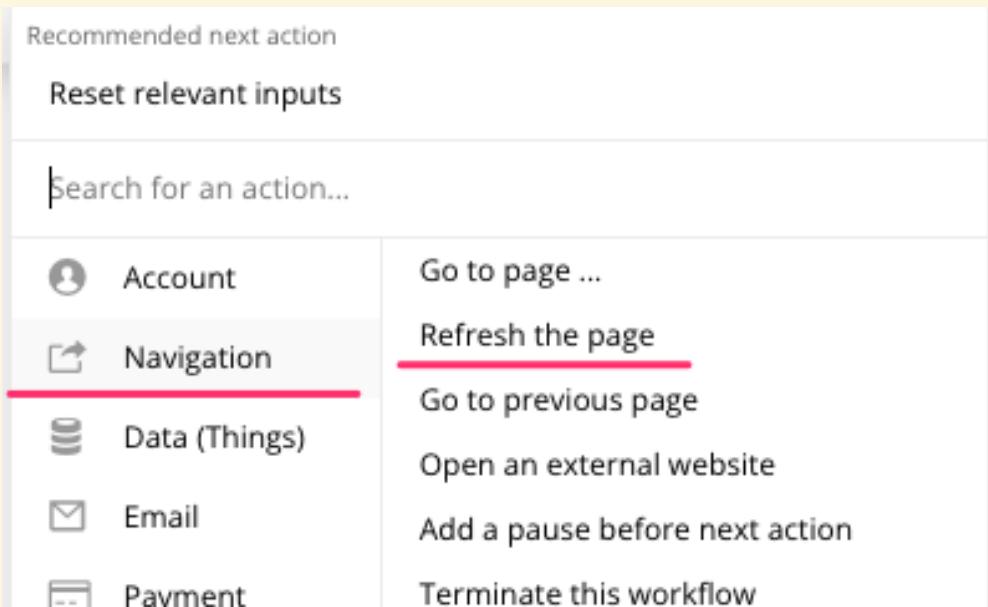
- Then, specify the type and fields to register
- Since we are registering pet weight, the Type is PetWeightLogs and the Field is PetWeightLogs.



- The fields to set are `WeightKg` and `pet`.
- If you do not set the `pet` field, it is impossible to determine the weight of any pet.
- The `pet` field must be set to `Current Page Pets`, which is the Pets on the current page.



- We want to redraw the screen after registration to refresh the graph to be displayed later, so let's define the behavior to redraw the screen after registration
- The action for manipulating the screen is **Navigation**.
- If you look at the submenu, you will find a suitable action **Refresh this page**.



- Now that we have registered the pet's weight, let's prepare a link from the detail screen
- Open the detail page, and provide a link between Image and Birthday.
- If your screen is not tall enough, increase the height of pet_detail itself.
- Use **Link** of **Visual elements** for the link.

- There are two key points
 1. Set the destination to Destination page .
 2. Pass the current page's pet information to the link destination.
- The following page shows an image of the setup, so let's start by setting it up yourself!

- Like this!

The screenshot shows a web editor interface with a dark-themed dialog box titled "Link Weight Logs". The dialog has three tabs: "Appearance" (selected), "Layout", and "Conditional". Inside, there's a "Rich text editor" containing the text "Weight Logs". Below it are settings for "Link destination" (set to "Internal page") and "Destination page" (set to "pet_weight_register"). The "Data to send" field contains "Current Page Pets". On the left side of the editor, there are two sections: "Image" (with "Current Page Pet's Image" and a blue-bordered "Weight Logs" link) and "Birthday" (with "Current Page Pet's Birthday:for..."). The "Weight Logs" link in the "Image" section is highlighted with a blue border and has a blue outline around its text.

Image
Current Page Pet's Image

Weight Logs

Birthday
Current Page Pet's Birthday:for...

Link Weight Logs

?

?

?

?

Appearance Layout Conditional

Weight Logs

Rich text editor

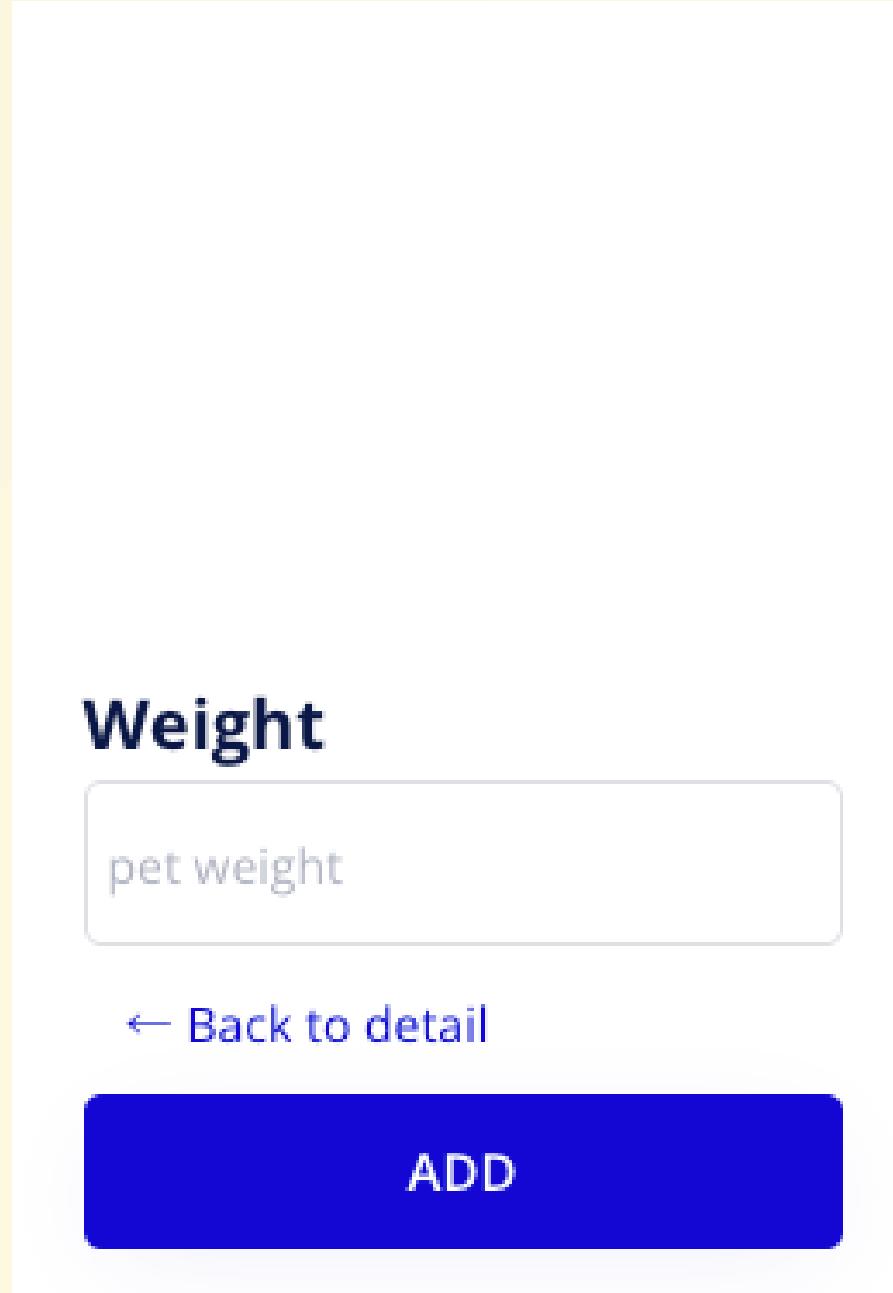
Link destination Internal page

Destination page pet_weight_register

Data to send Current Page Pets

- Once you have a lead from the detail screen to weight management, you should also have a reverse lead!
- The concept is the same as the link on the detail screen.

- Once set, let's go back to the list screen and run the preview
- After moving to List, Details, and Weight Management, enter a value in Weight and press ADD to redraw the screen?



- Just to be sure, let's check that the data is registered in the database
- If you select **All PetWeightLogs** from the App data tab in Data, do you see the data registered on the left side?

The screenshot shows the Airtable interface with the following details:

- Left sidebar:** Includes tabs for Design, Workflow, **Data** (selected), Styles, Plugins, and Settings.
- Top navigation:** Data types, Privacy, App data (selected), Option sets, File manager.
- Database views:** Application data - All PetWeightLogs - Development version.
- Actions:** Copy and restore database, Switch to live database.
- Tool buttons:** New view, Primary fields, Search, 2 entries (displaying 2), New entry, Delete (0), Upload, Modify, Export, Bulk.
- Search bar:** Search view names or data types...
- Table:** Shows data for All Pets. The table has columns: Pet, WeightKg, Created Date, Modified Date, Slug, and Created By. Two rows of data are displayed, both with the same Pet ID (3554x124158377302949890). The first row has a WeightKg of 12.7 and was created on Nov 13, 2022, at 11:29 pm. The second row has a WeightKg of 12.3 and was created on Nov 13, 2022, at 11:29 pm. Both rows were modified on Nov 13, 2022, at 11:29 pm. The Created By field for both rows is test@example.com.
- Bottom navigation:** All Pets, All PetWeightLogs (selected).

Let's display your pet's weight graphically

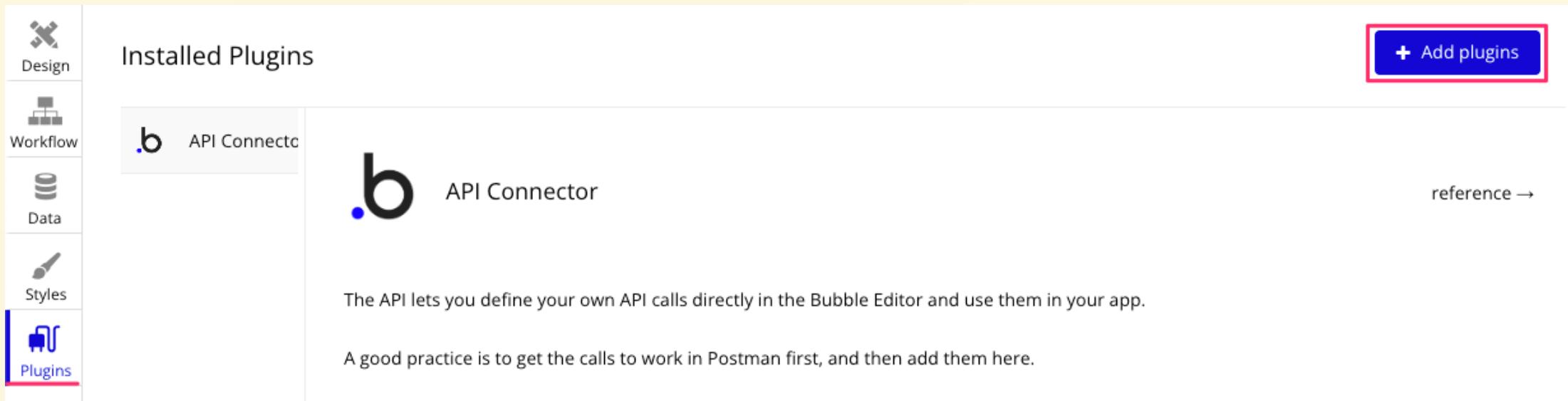
- Let's display the weight of your registered pets using a graph as in the case of Adalo



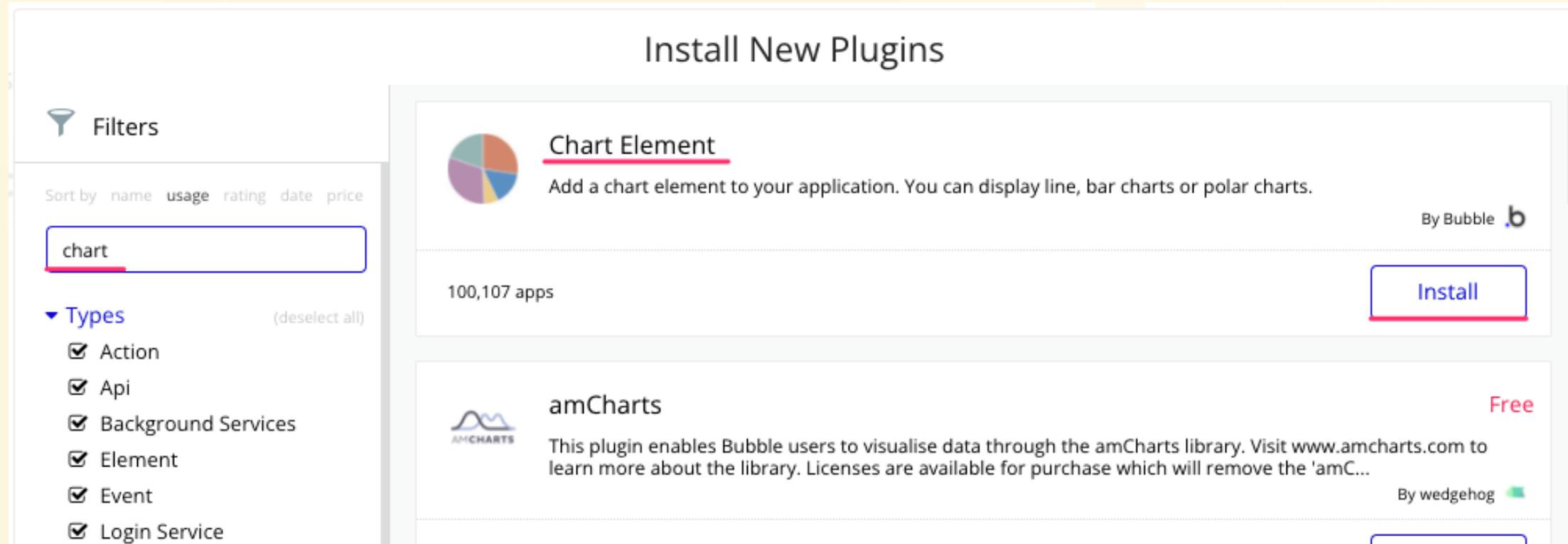
First, we will install a plugin.

- Bubble does not have the standard graph display functionality, so we will add this functionality in the form of a plugin.
- In the case of Adalo, it was in the form of Component, but in the case of Bubble, we will introduce it from Plugins.

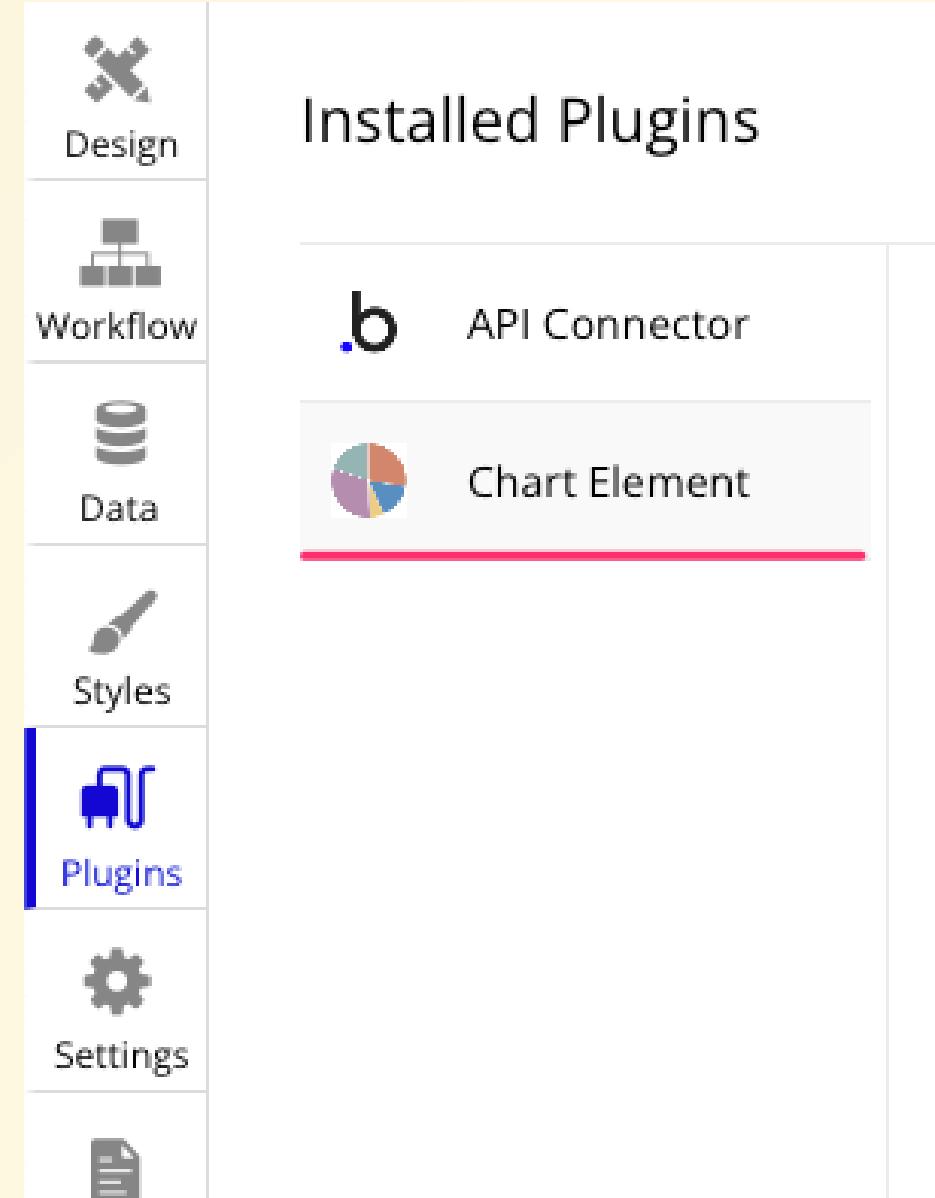
- Select Plugins from the left menu to display the plugins screen
- Then press **+ Add plugins** in the upper right corner to search for plugins to add.



- A pop-up window **Install New Plugins** will appear, and type **chart** in the search window on the left panel.
- Then, click the **Install** button for **Chart Element** (probably at the top).



- When the installation is complete, click DONE to close the pop-up window.
- If Chart Element is included in the Installed Plugins, it is OK.



- There are two types of Bubble plug-ins: free and paid.
- Plugins marked **Free** or **By Bubble** in the plugin list are free!
 - Especially, those marked **By Bubble** are literally the official plugins of Bubble, so you can trust them in some way.
- On the other hand, plug-ins that have a price listed are literally paid plug-ins.
- There are a variety of plug-ins available, so when introducing a plug-in, you should first look for a free plug-in that is sufficient for your needs, and if only paid plug-ins exist, you may consider installing a paid plug-in.



Simple LineChart from ApexCharts

Free

Easy to use beautiful line chart from the Apex JS library

By Avalan Labs

2,710 apps

All versions 4.0/5.0

Current version 4.0/5.0

Plugin page

Install



Chart JS - Charts and Graphs

\$190 once or \$9/month

24 highly-customizable and beautiful charts! These charts are built on the most popular Javascript chart library (Chart.js) as well as hundreds of hours of development for custom features and perf...

By TechBlocks

2,192 apps

All versions 4.8/5.0

Current version 4.8/5.0

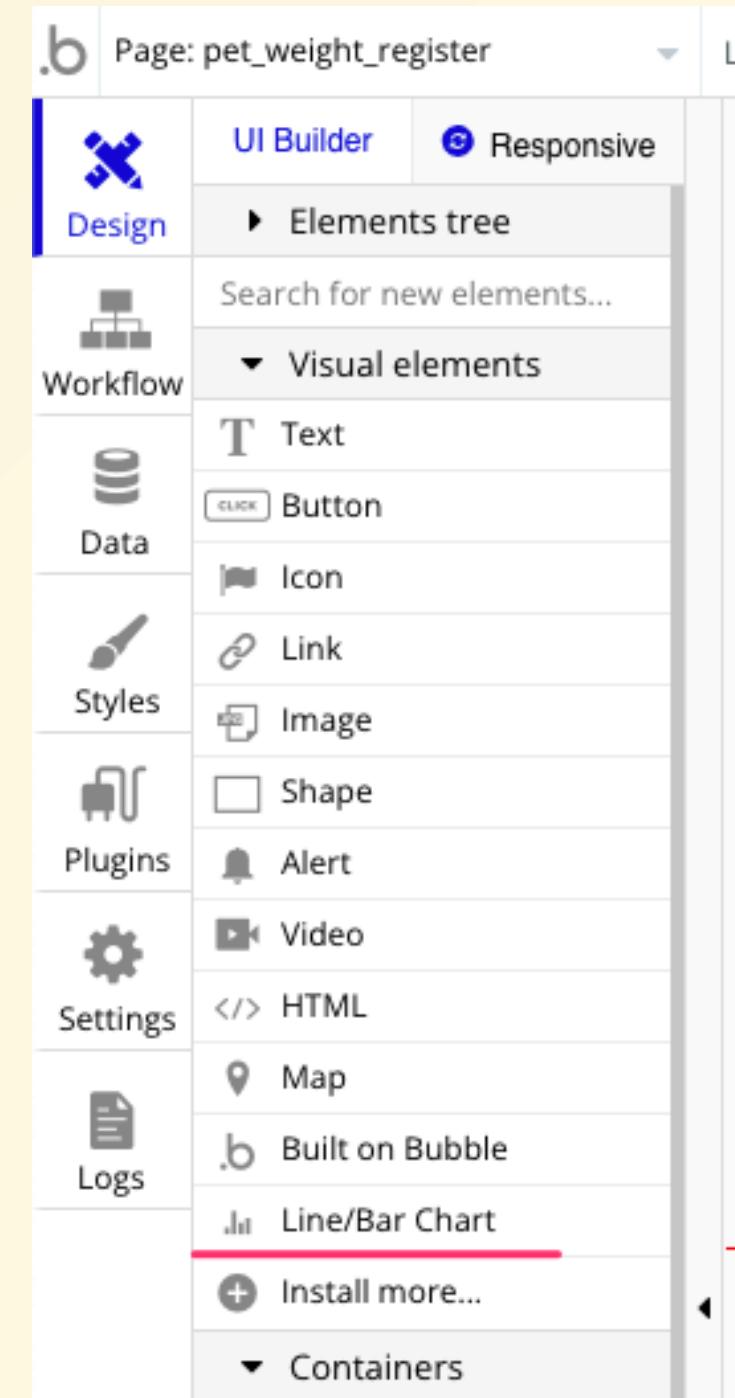
Plugin page

Buy \$190

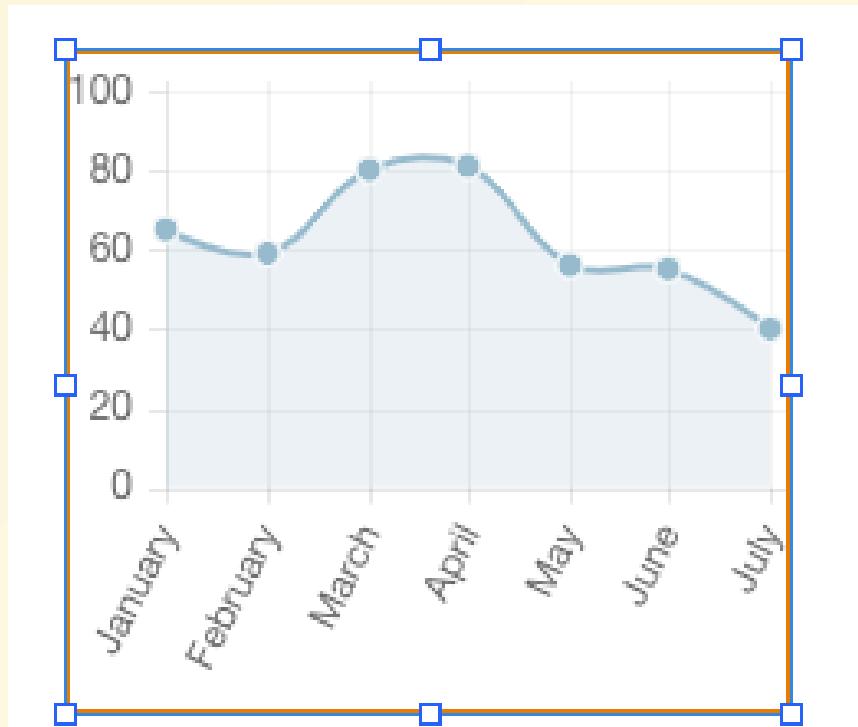
Subscribe

Now let's actually draw the graph!

- Open the Design section of the `pet_weight_register` page.
- Then you will see a new element `Line/Bar Chart` in the `Visual elements`, select this element and drag it to the top of the weight input.



- Adjust the size to this shape



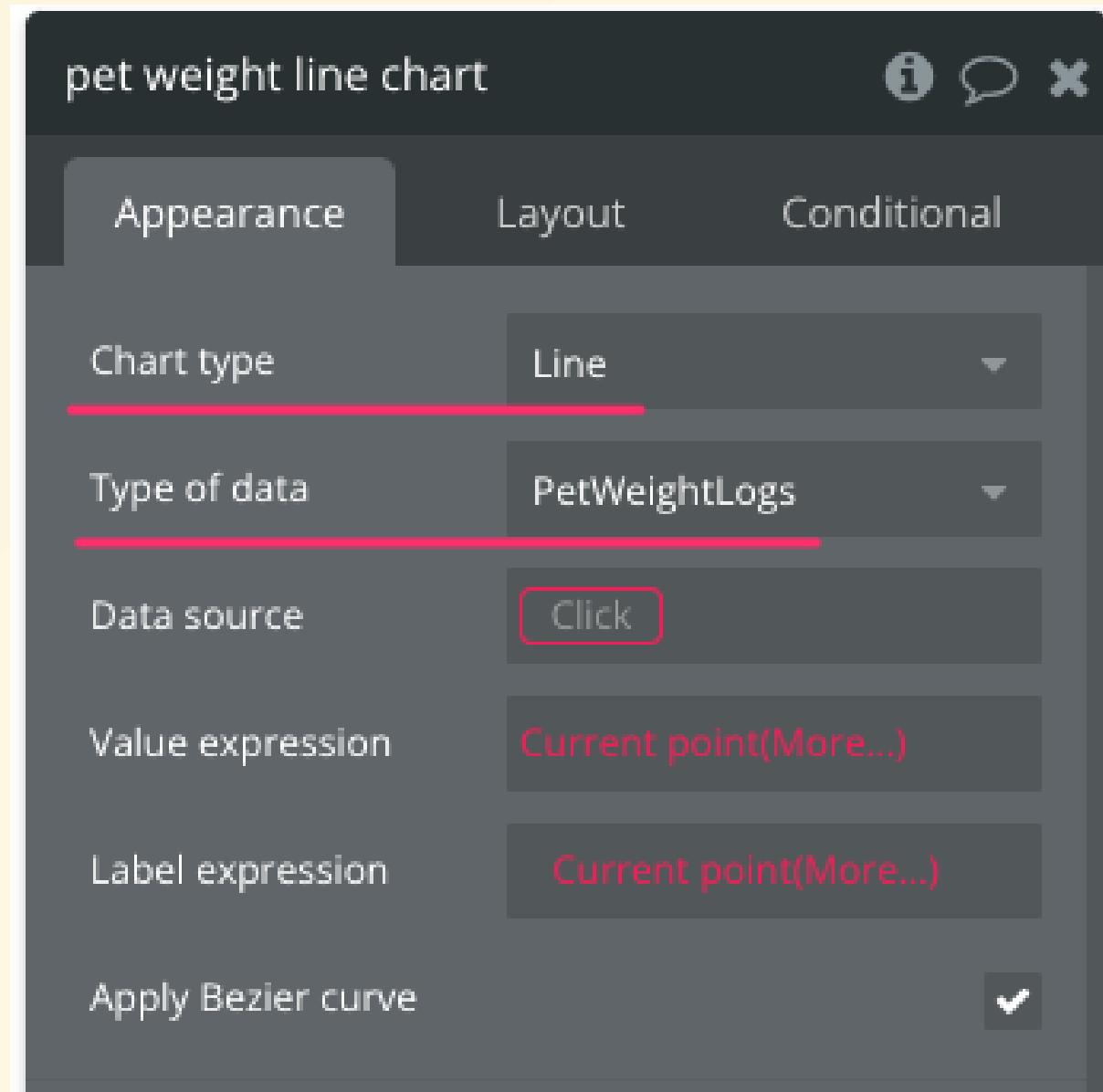
Weight

pet weight

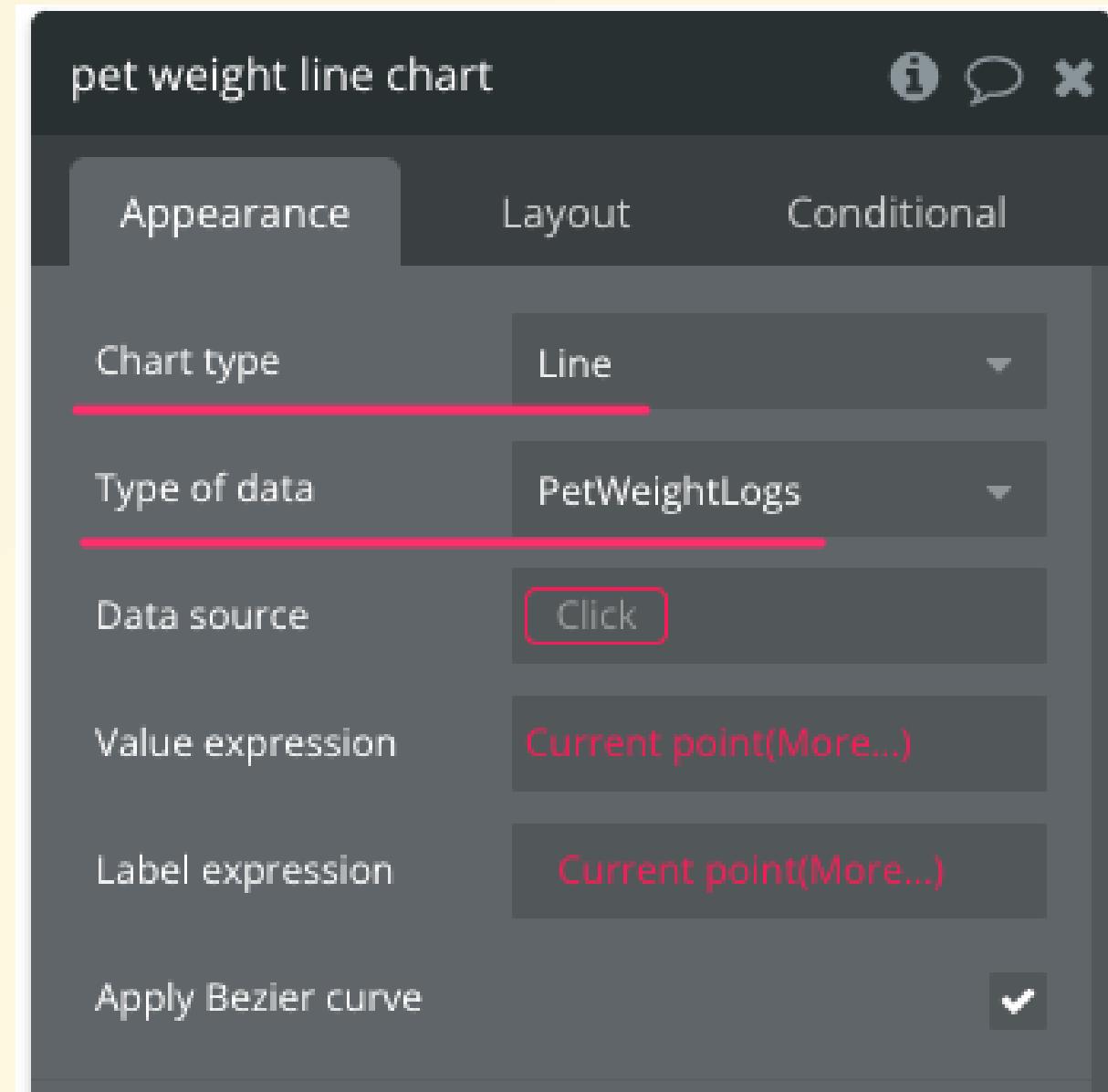
← Back to detail

ADD

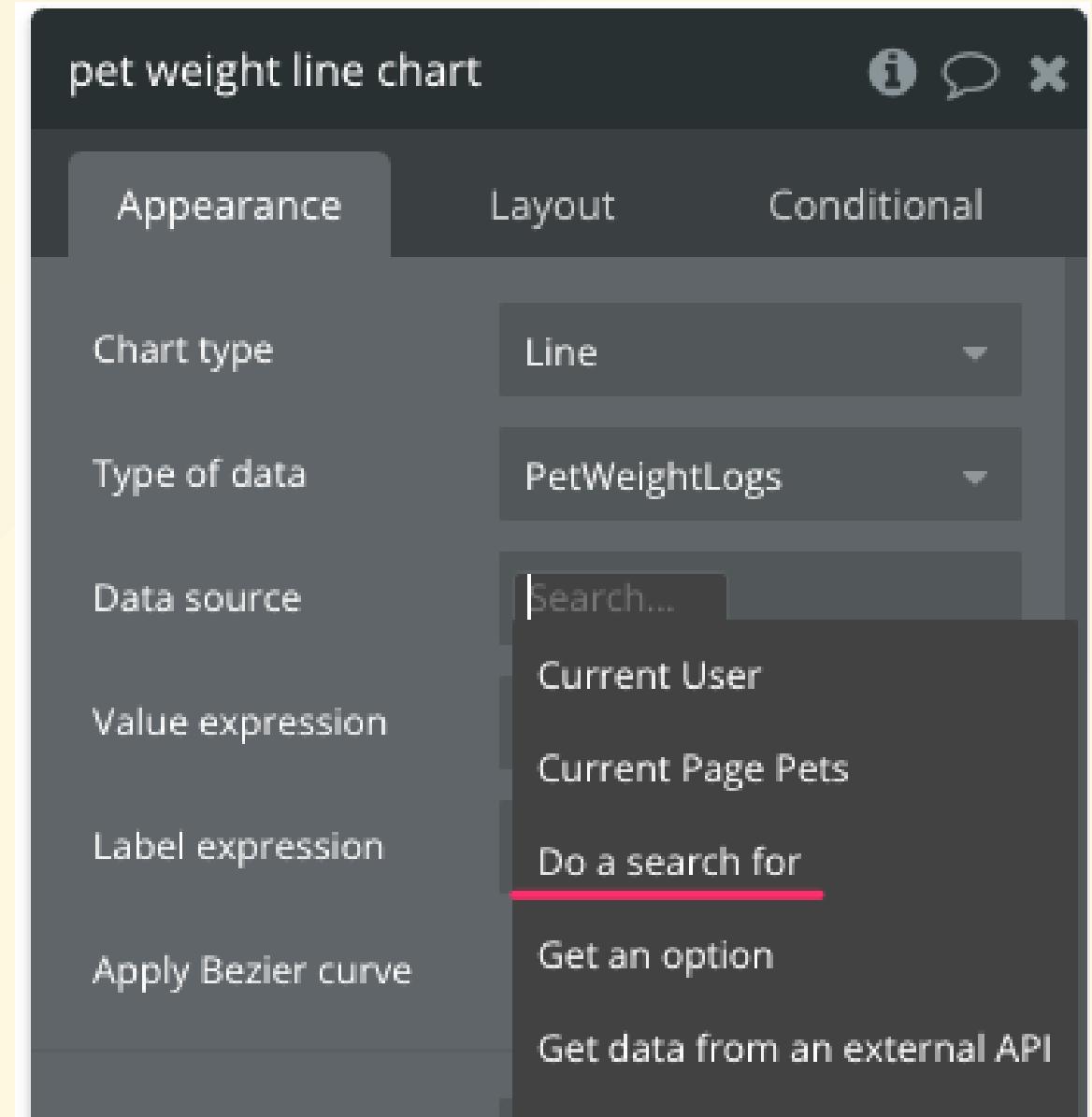
- After drawing, a dialog box for detailed settings will appear as before, so let's set each item!
- **Chart type** specifies the type of chart.
 - In this case, set it to **Line** since it is a line chart.



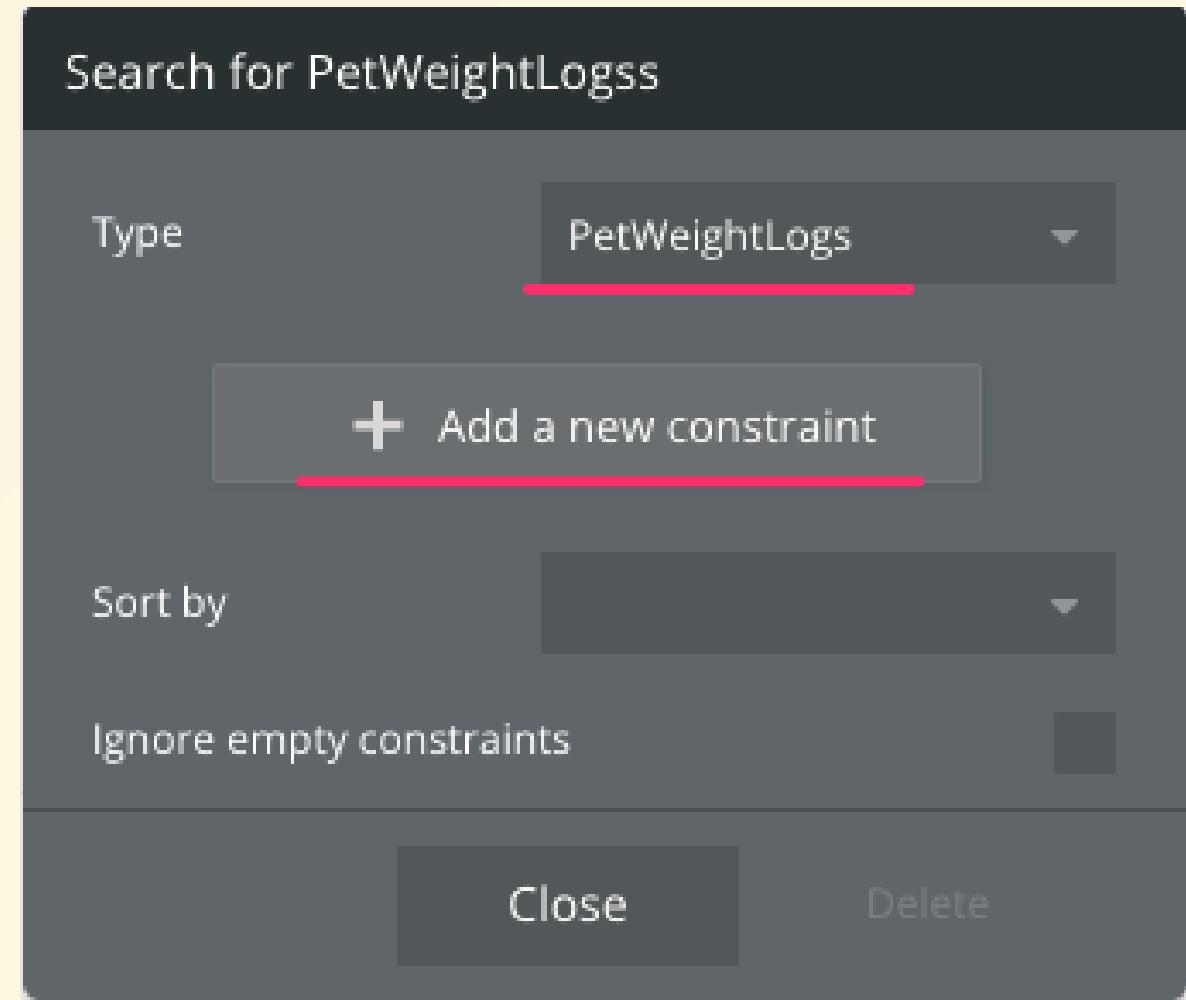
- Type of data specifies the type of data to be plotted in the graph.
 - In this case, PetWeightLogs is used to display the pet's weight.



- Continue to specify the conditions for the target data in the **Data source** field
 - Select **Do a search for** as you did in the list screen.



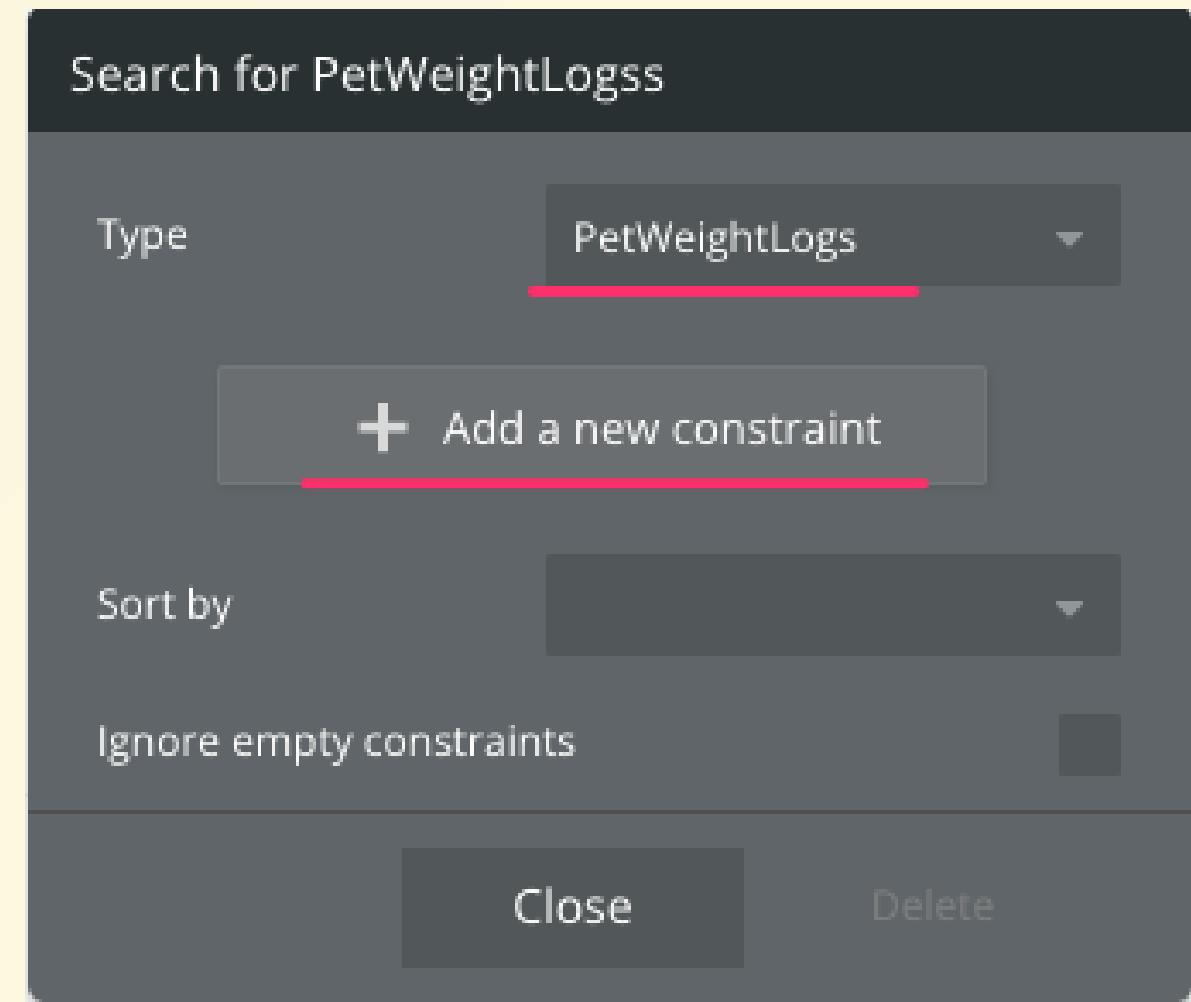
- A further 'Do a search for' dialog will appear.
 - Type is the type of the target data, so specify PetWeightLogs`.



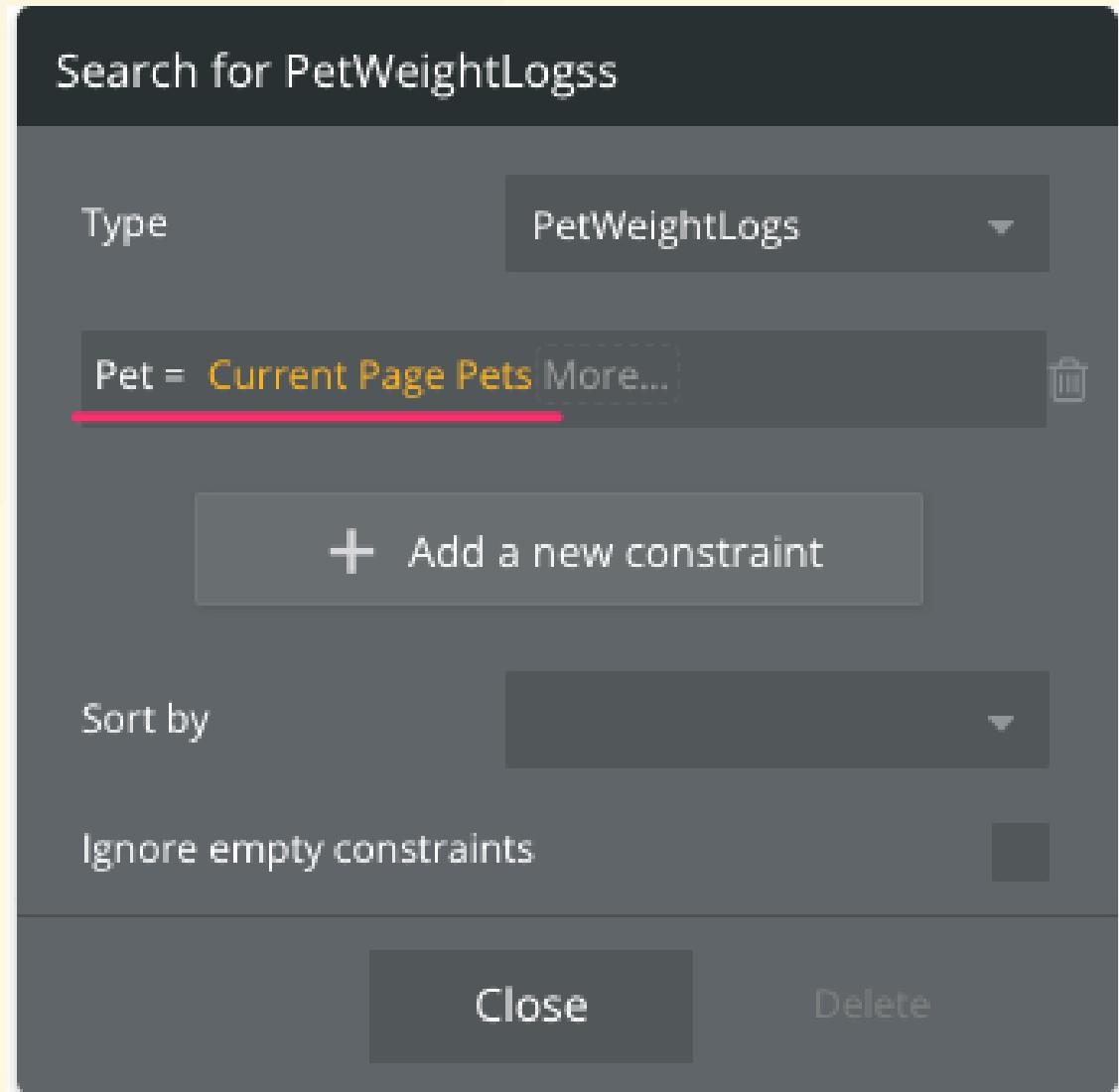
- This alone will display all the weight data for all pets, so we need to specify the criteria to narrow down the search.

- Click on

Add a new constraint to specify the criteria.

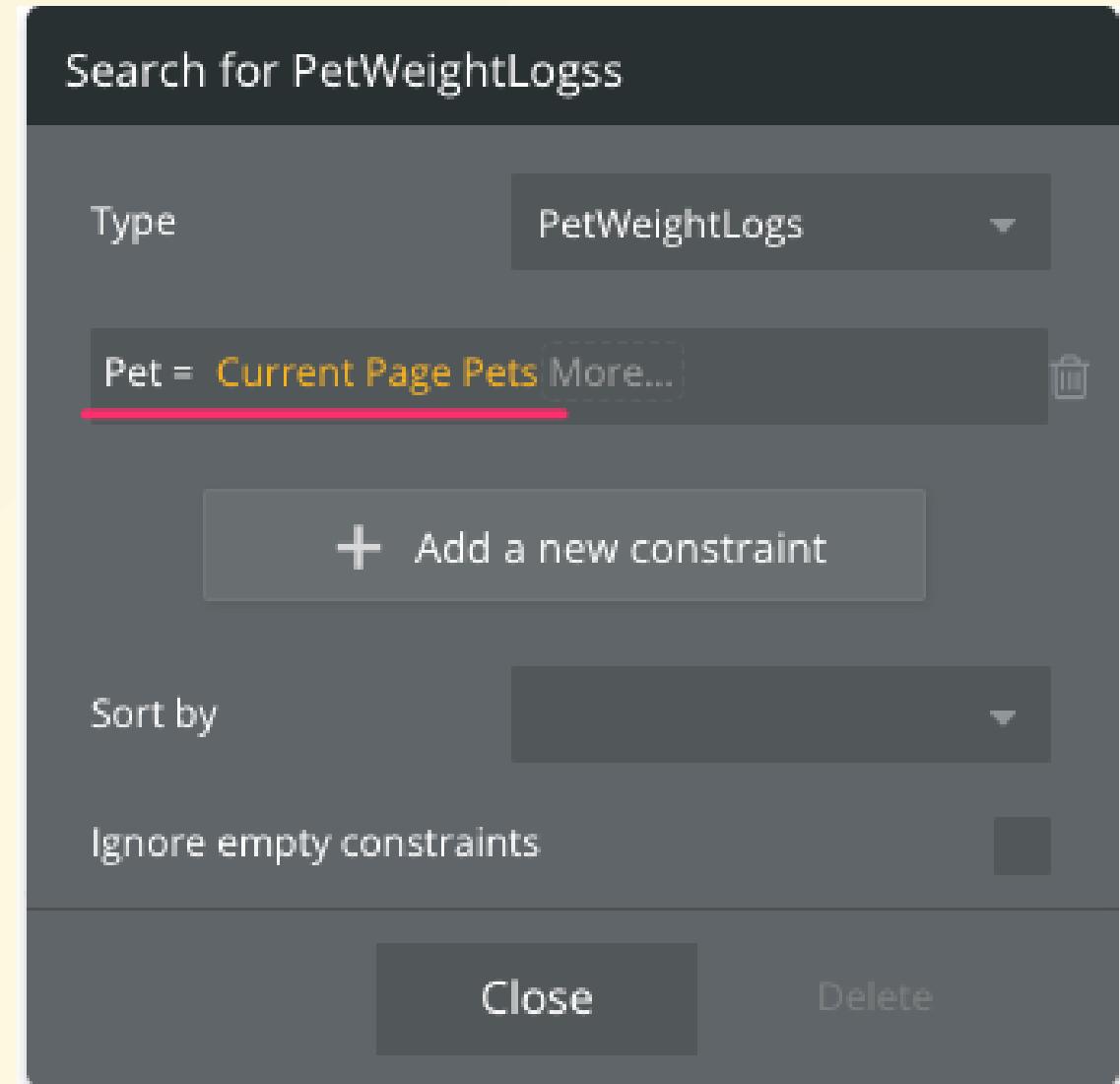


- First, click on "Click" to display the fields that the `PetWeightLogs` specified in Type has
- In this case, we will target the data of the `Pet` related to the `PetWeightLogs`, which is the same as the currently displayed pet.



- Let's express this as a condition, which looks like this, so let's set it up

Pet = Current Page Pets



- Finally, specify the order of the data
 - To show the graph, we would like to order the registered weights by the newest weight.



- Specify the `Created Date` in the `Sort by` field
 - `Sort by`` is now the key of the sort order.
- Specify `"yes"` for `Descending`.
 - This specifies the order of the keys in descending order.

Search for PetWeightLogs

Type	PetWeightLogs
Pet = Current Page Pets	
+ Add a new constraint	
Sort by	<u>Created Date</u>
Descending	<u>"yes"</u>
+ Add another sort field	
Ignore empty constraints	
Close	Delete

- If you want to sort the list by the oldest registration date, set **Descending** to "no" to sort the list by the oldest registration date (ascending order).
- After setting up, click **close** to close this sub-dialog.

- Finally, specify the X-axis and Y-axis to be plotted on the graph.
- In this case, the X-axis is the date and time of registration and the Y-axis is the weight.



- Set Label expression for the X axis.

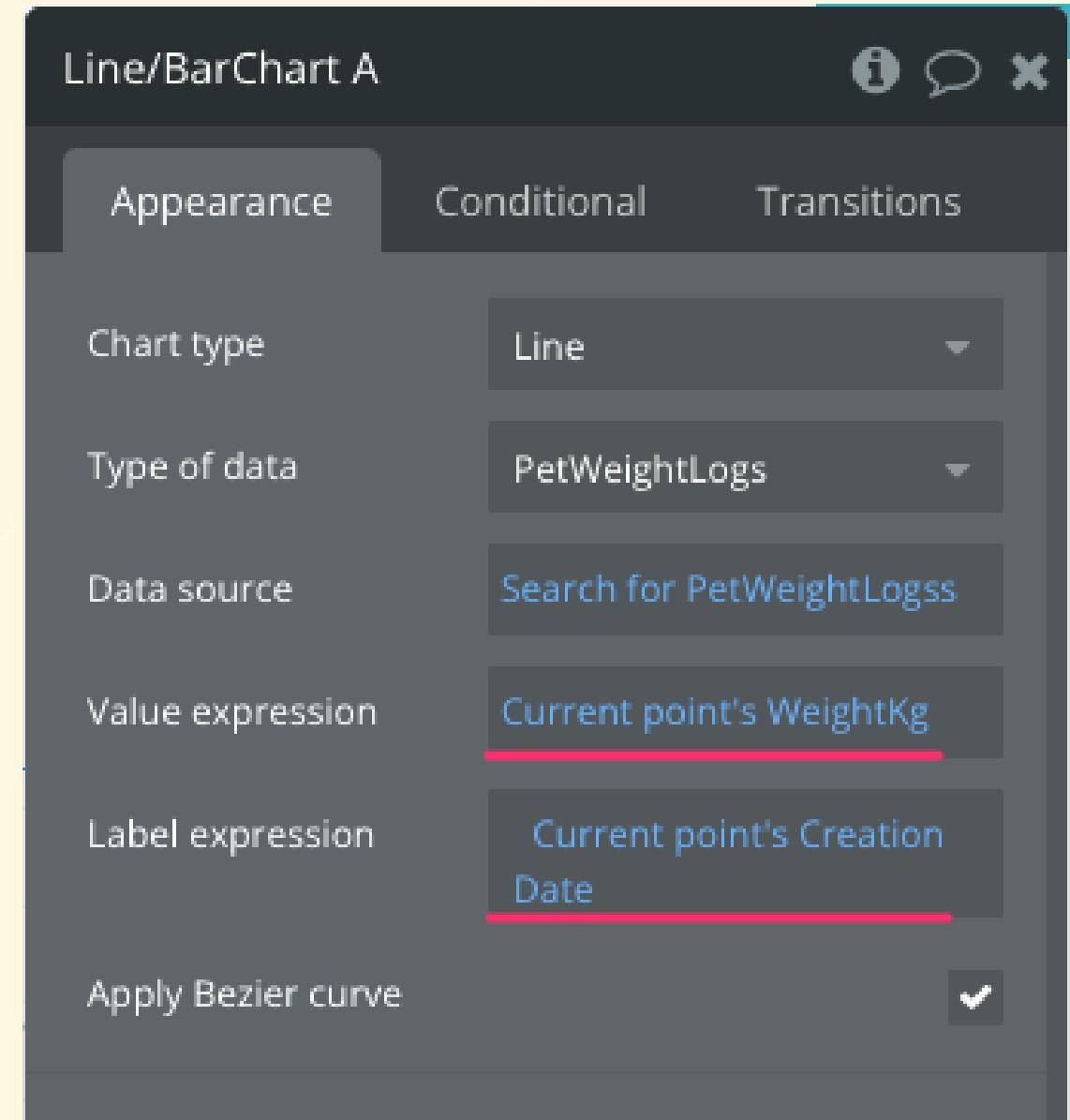
- This time,
Current point's
Creation Date

is set.

- The Y-axis is set to Value expression.

- The Y-axis is set to Value expression, this time

Current point's
WeightKg



- Current point refers to each individual data plot of each of the PetWeightLogs specified in the target data type

- Now let's go back to the list screen and preview!
- Do you see a graph showing the registered weight of the pet you selected in the list?
- Does the graph change as you add more and more weights on the same screen?



Exercise 6: Change the X-axis labels on the graph

- The current X-axis labels are in a long date/time format, so you can change them to whatever format you like.
- For example, here is what it looks like



Exercise 7: Send a message when there is no weight data

- With the current development, if no pet weight data is registered, the graph drawing area will be blank, and the page will not look good.

Weight

pet weight

← Back to detail

ADD

- So, let's improve usability by displaying a dedicated message when none of your pet's weights are registered!



- The corresponding image is a Shape with the same width and height as the graph element at the same position.
 - Place the message you want to display on it as Text, and group the Shape and Text.
 - By grouping the Shape and Text, they can be treated as a single element.
- Set the following for the Shape
 - Always hide the element itself
 - Show the element if the number of PetWeightLogs in Conditional is 0.
- In this way, if no weight is registered, the shape will be displayed, otherwise the graph will be displayed.

- Here is an image of the setting

The image consists of two side-by-side screenshots of a user interface for managing search filters.

Left Screenshot: Search for PetWeightLogss

- Type: PetWeightLogs
- pet = Current Page Pets
- + Add a new constraint
- Sort by
- Ignore empty constraints
- Close
- Delete

Right Screenshot: Group A

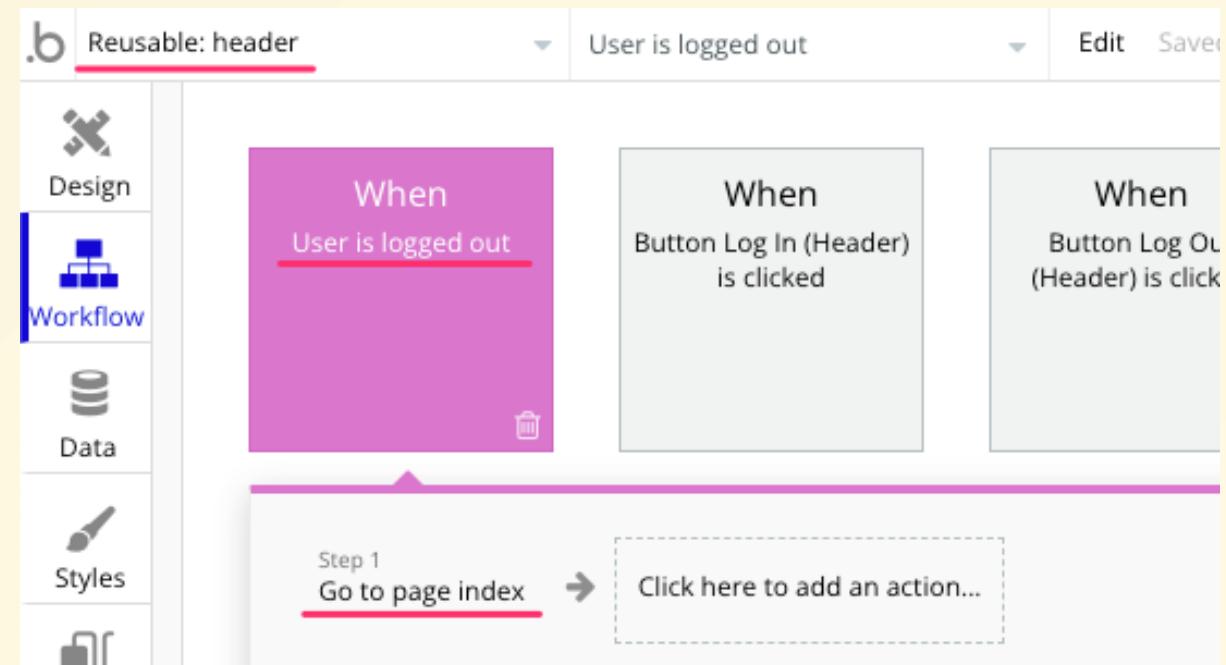
- Appearance
- Conditional (1)
- Transitions
- OFF remove condition
- When Search for PetWeightLogss:count is 0 More...
- This element is visible This element is visible
- Select a property to change when true
- + Define another condition

Exercise 8: Let's always transition to the login screen when the user is not logged in.

- We don't want all the screens where header components are placed to be used only when the user is logged in, do we?
- So, let's force a transition to the login screen (index) when the user is not logged in to the screen where the header component is placed.

- Hint 
- Instead of setting up a workflow for the screen where the header component is placed, the workflow is set up for the header component itself.
- The configuration is simple
- When: "Go to page index" if the user is currently not logged in
- The answer is on the next page!

- For the header component
 - When: User is logged out
 - Go to page index
- Unlike other workflows, this one does not trigger a button, so the color has been changed for clarity



Exercise 9: Let's create an update function with a lead from the detail screen to the update screen.

- The key point is the one on the next sheet! Please put them into practice!

Name

Image



Birthday

Gender

[← Back to detail](#)

UPDATE

- The update screen has almost the same screen elements as the registration screen, so let's clone `pet_register` and create `pet_update`.
- In the registration screen, each input element (name, birthday, etc.) was empty, but in the update screen, it is necessary to set the values of data already registered at the time of displaying the screen.
- In the update screen, we need to set the values of the already registered data at the time the screen is displayed.
 - Hint  Insert dynamic data
 - Hint  Initial content

- As for the behavior when the update button is pressed, it was **Create a new thing...** in **Data(Thing)** when a new registration was made, but this time it is an update, right? `, but this time it is an update, right?
 - Speaking of update, change... Oh, there seems to be an action like that 😊
- Finally, when we move from the detail screen to the update screen, we need to pass the information of the pet displayed in the detail screen. This is the part I learned in today's lecture!
 - Hint 💡 Data to send

Exercise 10

Feel free to add functionality using the features you learned today!
When you are done, share the URL of the login screen on Slack for everyone to see!

Presentation of the results of Exercise 10

(If there is time)

Introduce the screens and functions created in the exercise 😊 ?

Summary

- In this lecture, we started with how to use Bubble, and then moved on to the actual designing of the screen, and then to the integration with the database.
- While the basic flow of Bubble is similar to that of Adalo, I hope you could feel a little of why it is called a visual programming tool in terms of workflow.
- Next time, I would like to give a more in-depth lecture on Bubble, so stay tuned!

That's all!

Thank you for your hard work!