

Programming Boot Camp

Bubble Basics

Tokyo Institute of Technology 2023/11/04

Naotake KYOGOKU

目次

- What is Bubble? (P3 ~)
- Let's start by creating a screen with user registration and signup components. (P25 ~)
- Brief overview of application development with Bubble (P43 ~)
- Let's make it from pet registration to listing (P56 ~)
- Connecting to a database (P93 ~)
- Let's create a header component. (P203 ~)
- Keep track of your pet's weight! (P292 ~)
- Summary (P349 ~)

What is Bubble?

- Bubble 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 top page
 - <https://bubble.io/>
 - Click "Get started"

The screenshot shows the official website for Bubble. At the top, there is a navigation bar with the ".bubble" logo on the left and links for "Product", "Use cases", "Pricing", "Showcase", "Academy", "Contact sales", "Log in", and a prominent blue "Get started" button on the right. Below the navigation bar, a dark banner displays the text "BubbleCon2023: That's a wrap! See the BubbleCon recaps." with a close button (an 'X') on the right. The main content area features a large, bold headline: "The world's only full-stack, no-code platform".

- 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 the day, select "BUILD".

GET STARTED

What is your goal today?

Let us know how you plan to use Bubble so we can tailor your experience to meet your needs



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.

- You will be asked how comfortable you are with learning new technology, so everyone should choose at your own level.

GET STARTED

How comfortable are you learning new technology?

Let's customize your Bubble experience with the right level of support based on your technical skill level.



CHALLENGING

I find it difficult to learn new technologies.



DILIGENT

I am non-technical, but I work hard to learn new tech.



ADAPTABLE

I am non-technical, and I pick up new tech quickly and easily.



EXPERT

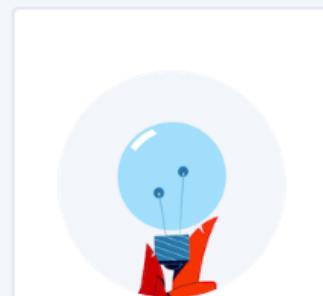
I have a technical background and pick up new tech very easily.

- You are asked at what stage you have an idea for the application you will create in Bubble, but in this case, I will select "EXECUTION" because it is clear what you will create in the lecture.

GET STARTED

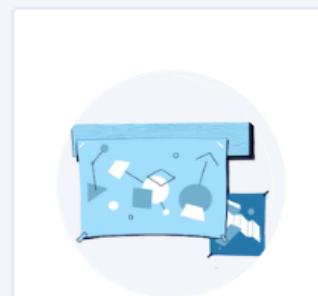
What best describes the stage of the idea you are looking to build on Bubble?

No matter where you are at in your ideation process, Bubble can be the right tool for you.



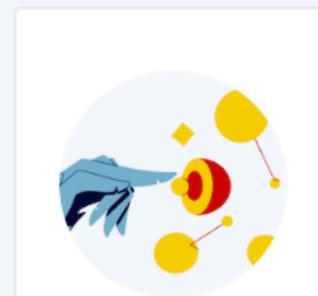
INSPIRATION

I'm still in the ideation stage and haven't settled on an idea.



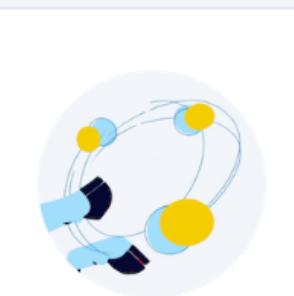
PLANNING

I have a rough idea and am starting to flesh out the details.



EXECUTION

I have a clear idea with wireframes and am ready to start building.



REBUILDING

I have a working product and want to rebuild it on Bubble.

- Finally, you will be asked where you learned about Bubble.

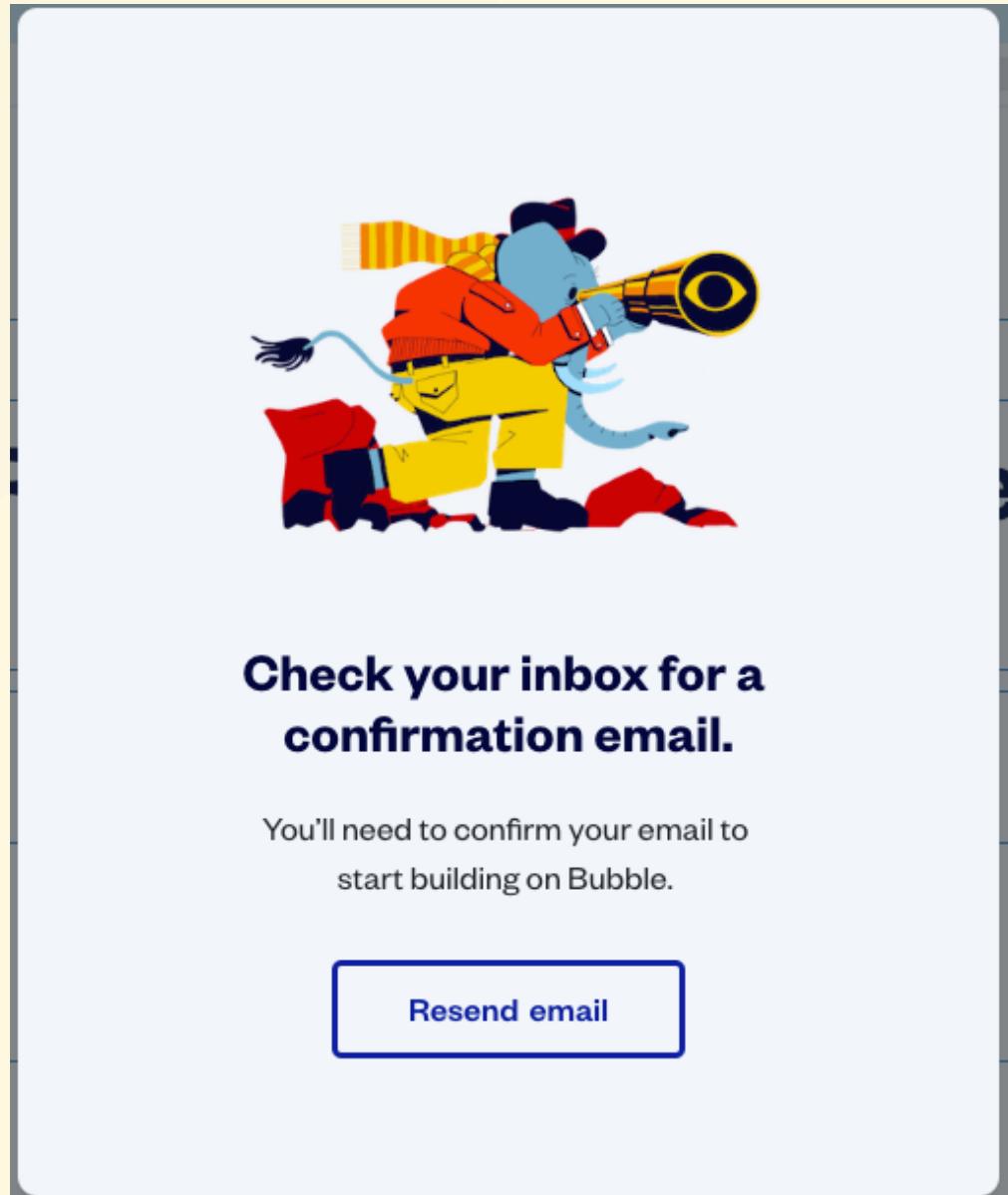
GET STARTED

Finally, where did you hear about Bubble?

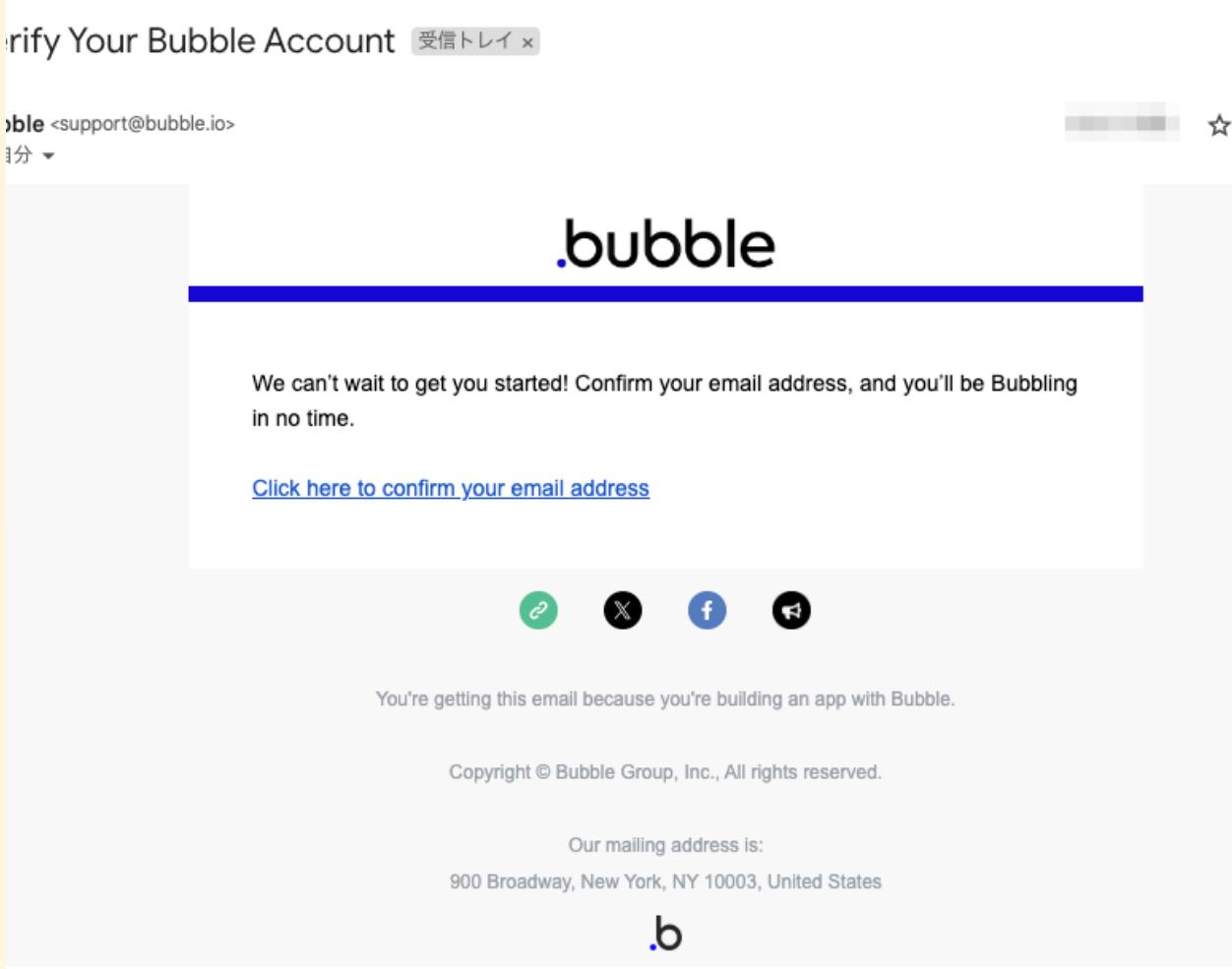
We'd love to know how you discovered us so we can continue to reach more people like you.

- Podcast
- Press
- Course
- Forums (Reddit, Hackernews...)
- Incubator
- Blog
- Other social networks
- Twitter
- Friends or colleagues
- Discovery platforms (PH, Quora...)
- LinkedIn
- Facebook/Instagram
- Search engines
- Review sites (G2, Capterra...)
- Other

- 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"

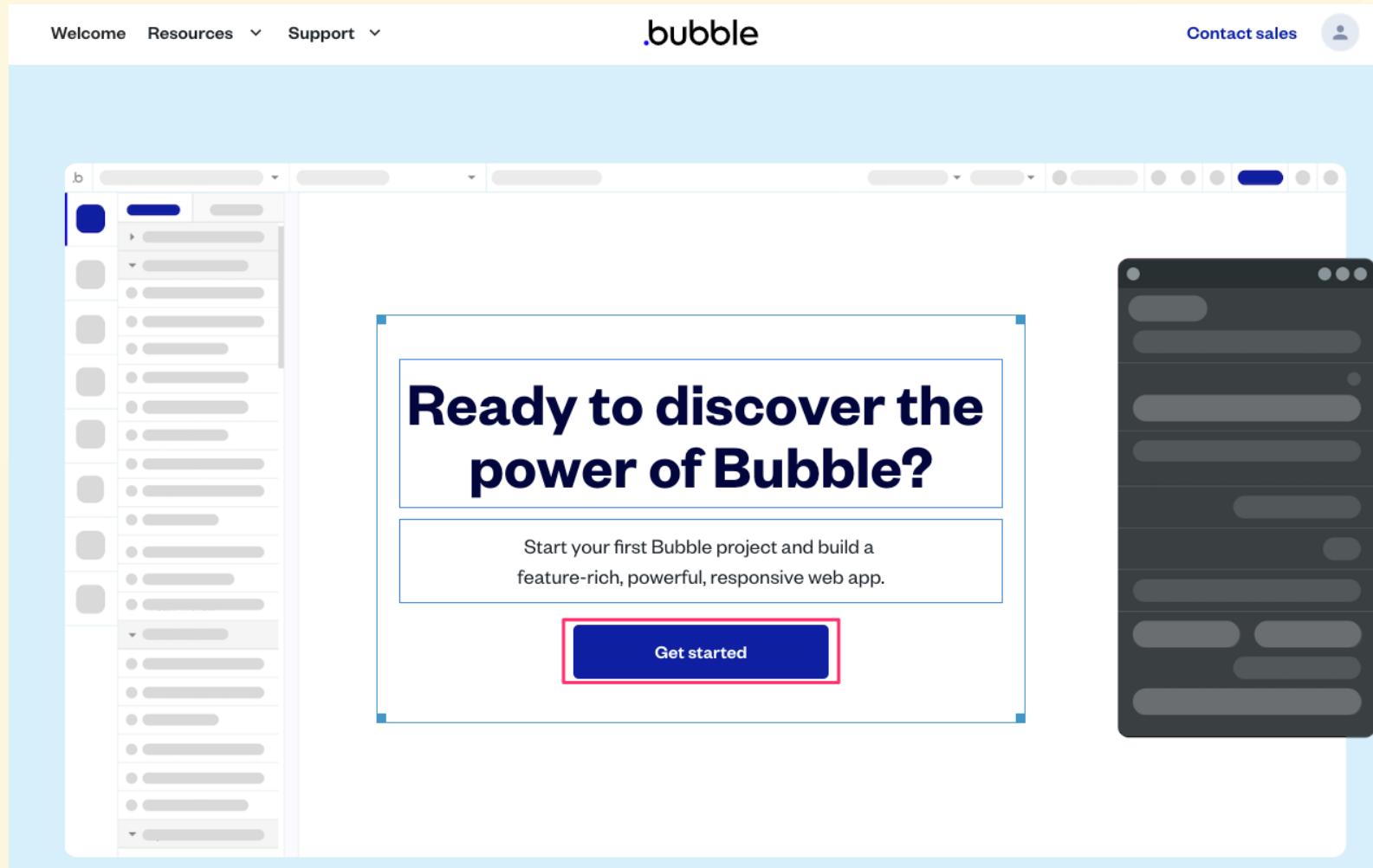
Thanks for confirming!

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

Start building



- Click "Get started" when the Welcome page appears.



Click "Start with basic features" under FREE PLAN on the left side when asked about the plan for the app you are about to create.

The screenshot shows a comparison of two plans: Starter Plan and Free Plan. The Starter Plan is recommended and offers a free trial for 14 days. It includes features like custom domains, publishing, extensibility, and automated actions. The Free Plan is \$0 and includes core features and hosting on a Bubble domain. A call-to-action button at the bottom of the Free Plan section says "Start with basic features".

Build your app with premium features
Upgrade or downgrade at any time.

RECOMMENDED

STARTER PLAN

Free for 14 days!

- 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 services like Stripe, Google Maps and more.
- Automated actions**
Develop recurring workflows to automate processes.

Activate free trial

Free for 14 days, then \$32 per month. Cancel anytime.

FREE PLAN

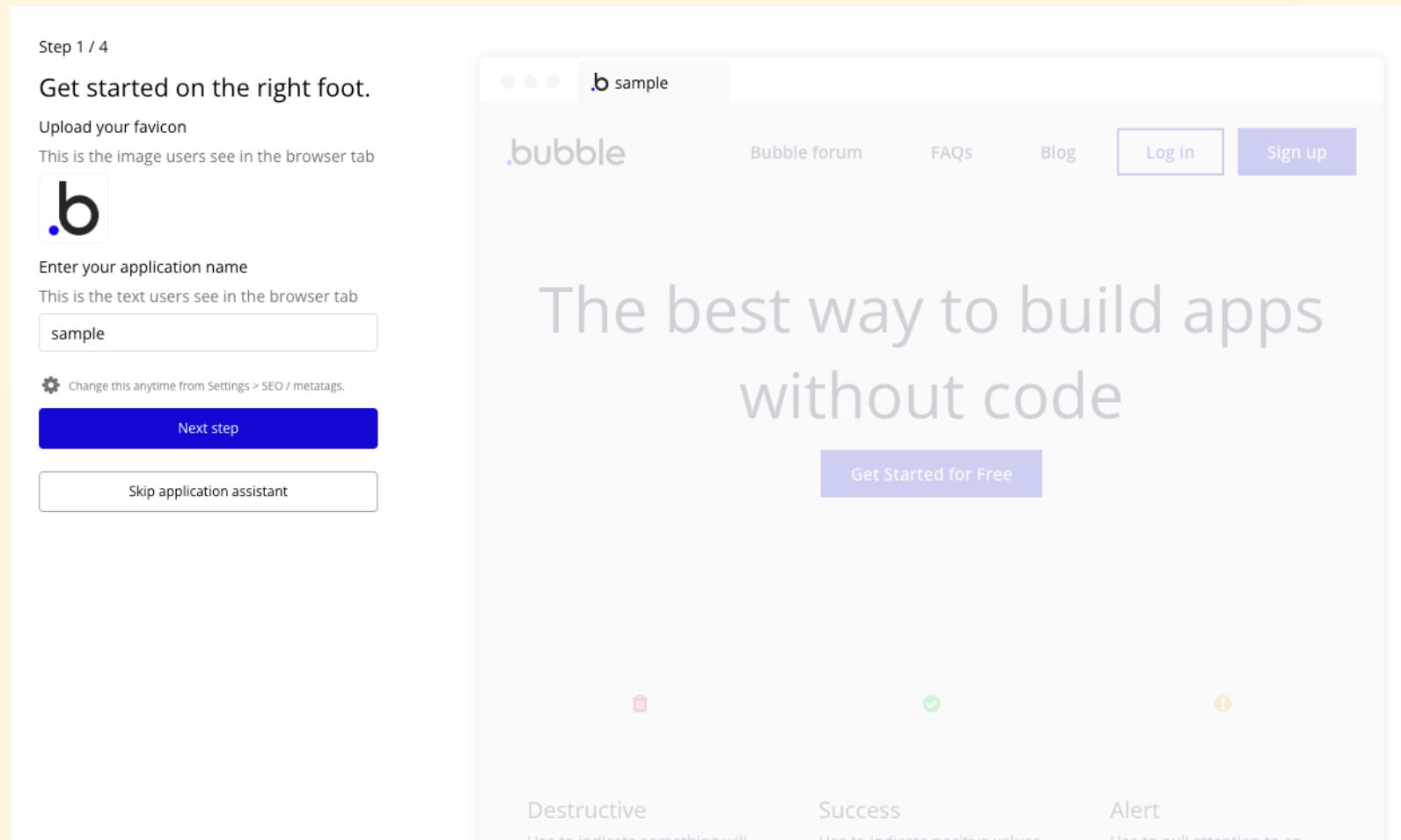
\$0

- Core features**
Access to basic features for testing & learning.
- Host on Bubble domain**
Preview your app on a bubbleapps.io domain.

Start with basic features

No credit card required. Upgrade anytime.

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



- Step 1/4 will ask 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 1 / 4

Get started on the right foot.

Upload your favicon

This is the image users see in the browser tab



Enter your application name

This is the text users see in the browser tab

sample



Change this anytime from Settings > SEO / metatags.

Next step

Skip application assistant

- Step 2 of 4 is to select the standard font of the application.
- You can change it later, so leave it as it is and click Next step.

Step 2 / 4

Set your default font

Your app font is applied to all elements as the default font-family in styles

App font

Open Sans



Change this anytime from Styles > Style variables.

Back

Next step

Skip application assistant

- Step 3/4 selects the colors for the various standard elements of the app.
- You can change it later, so leave it as it is and click Next step.

Step 3 / 4

Set your default colors

Your color variables will apply to your default elements and the component library

Primary ⓘ #0205D3 █

Primary contrast ⓘ #FFFFFF

Text ⓘ #091747 █

Surface ⓘ #FFFFFF

Background ⓘ #FFFFFF

Destructive ⓘ #FF0000 █

Success ⓘ #17DB4E █

Alert ⓘ #FAB515 █



Change this anytime from Styles > Style variables.

Back

Next step

Skip application assistant

- Step 4/4 is to choose whether to install plug-ins for the application.
- In the right panel, you can see some official Bubble plug-ins, but you can install them later.
- Click "Get started building

Step 4 / 4

Want to add plugins to your application?

Plugins allow you to define APIs, install elements, events and more



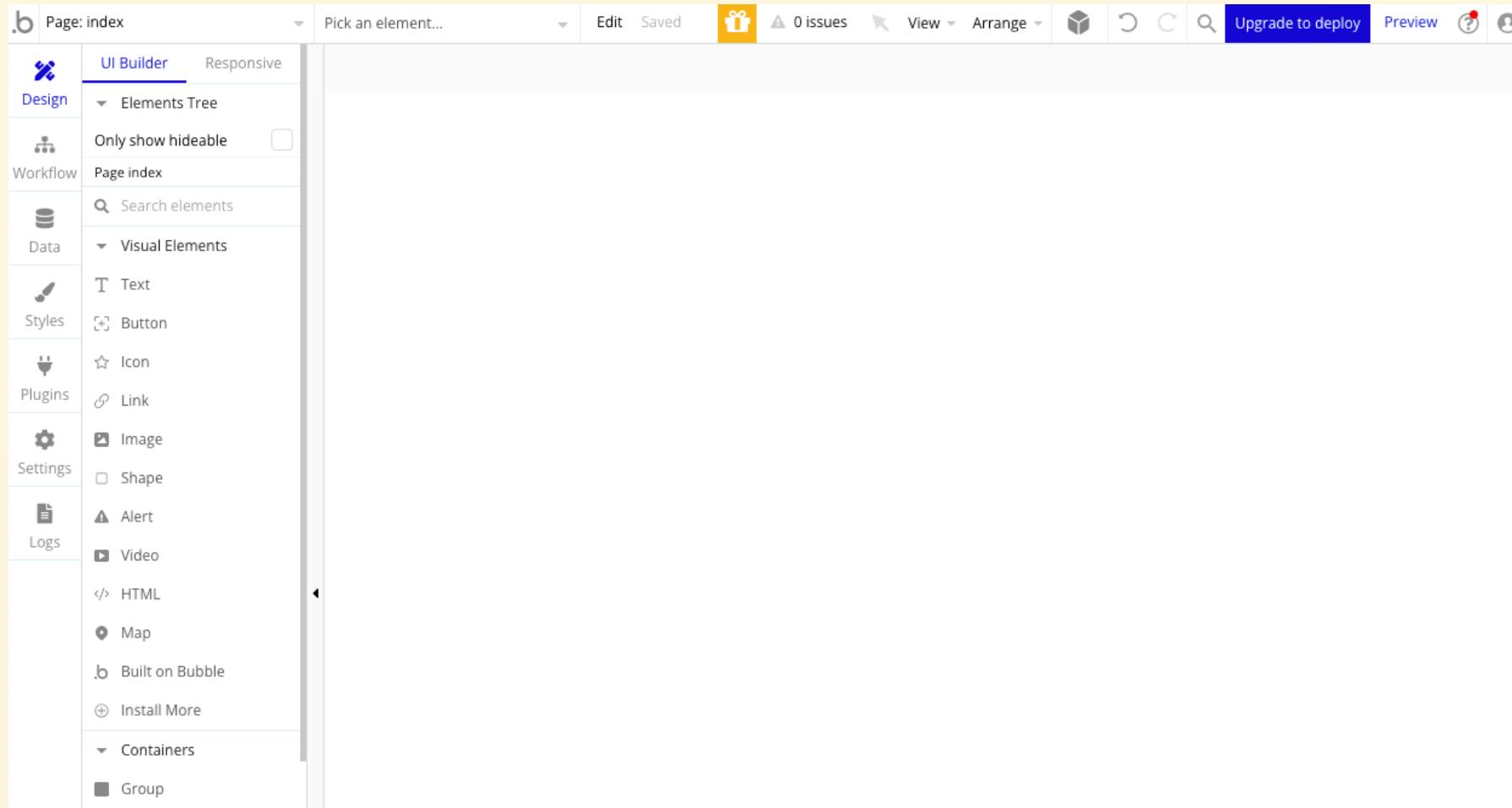
Change this anytime from Plugins > Add plugins.

Back

Get started building

Skip application assistant

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



Let's get the Bubble template working first!

- 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 is 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 these parts...
- ⚠ The actual sign-up and login functions will be developed later, so for now, it only works as a storyboard.

Let's start by creating a screen with user registration and signup components.

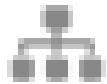
- warning: We will touch various parts of the screen, but we will explain each part later.
- Let's start with the minimum explanation of the parts and create a user registration screen.

Three main operations

- Bubble uses three main operations
 1. design
 2. Workflow
 3. Data
- You can go back and forth between each of them from the left menu



Design



Workflow

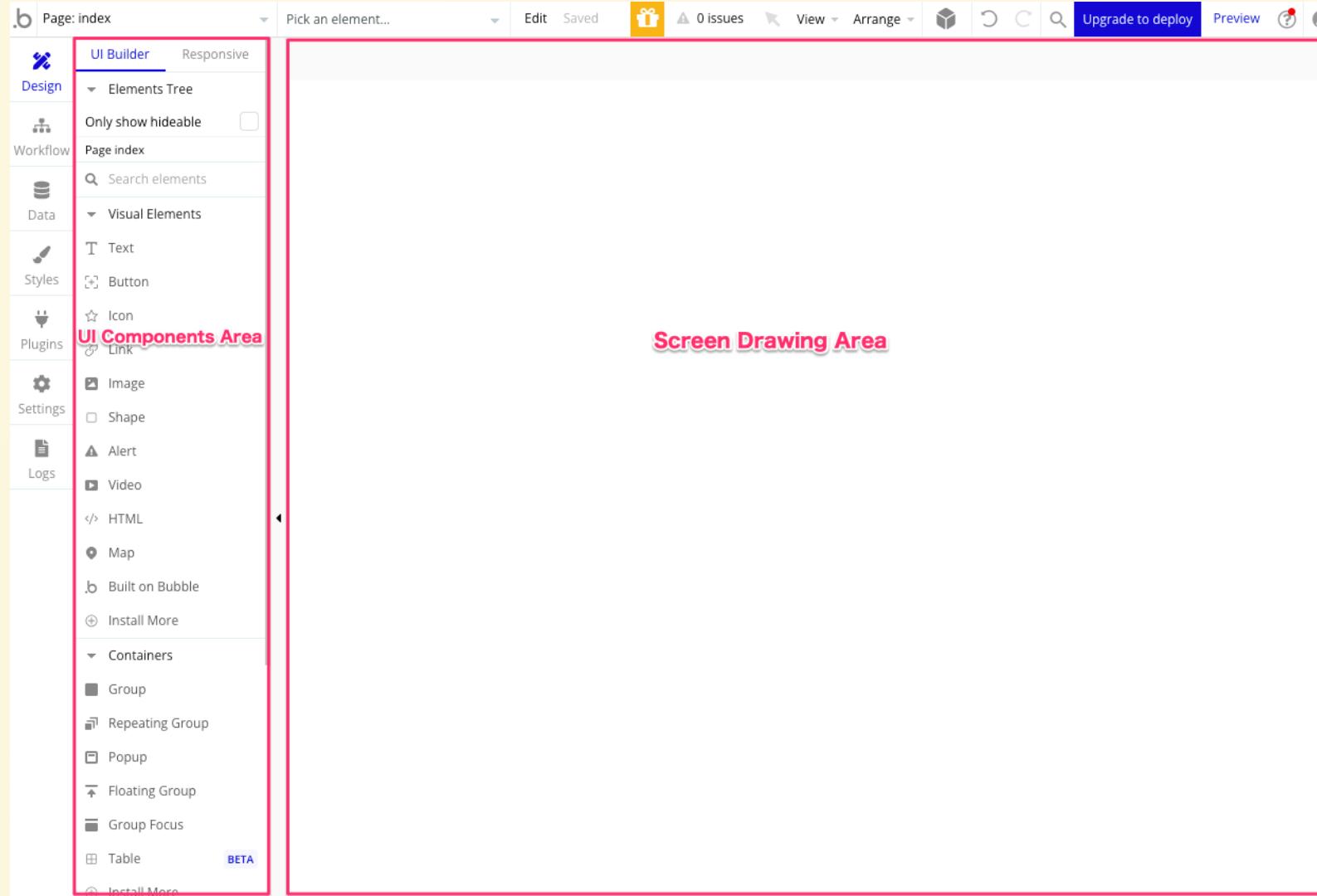


Data

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 a user registration!

.b

Welcome to Bubble

Name

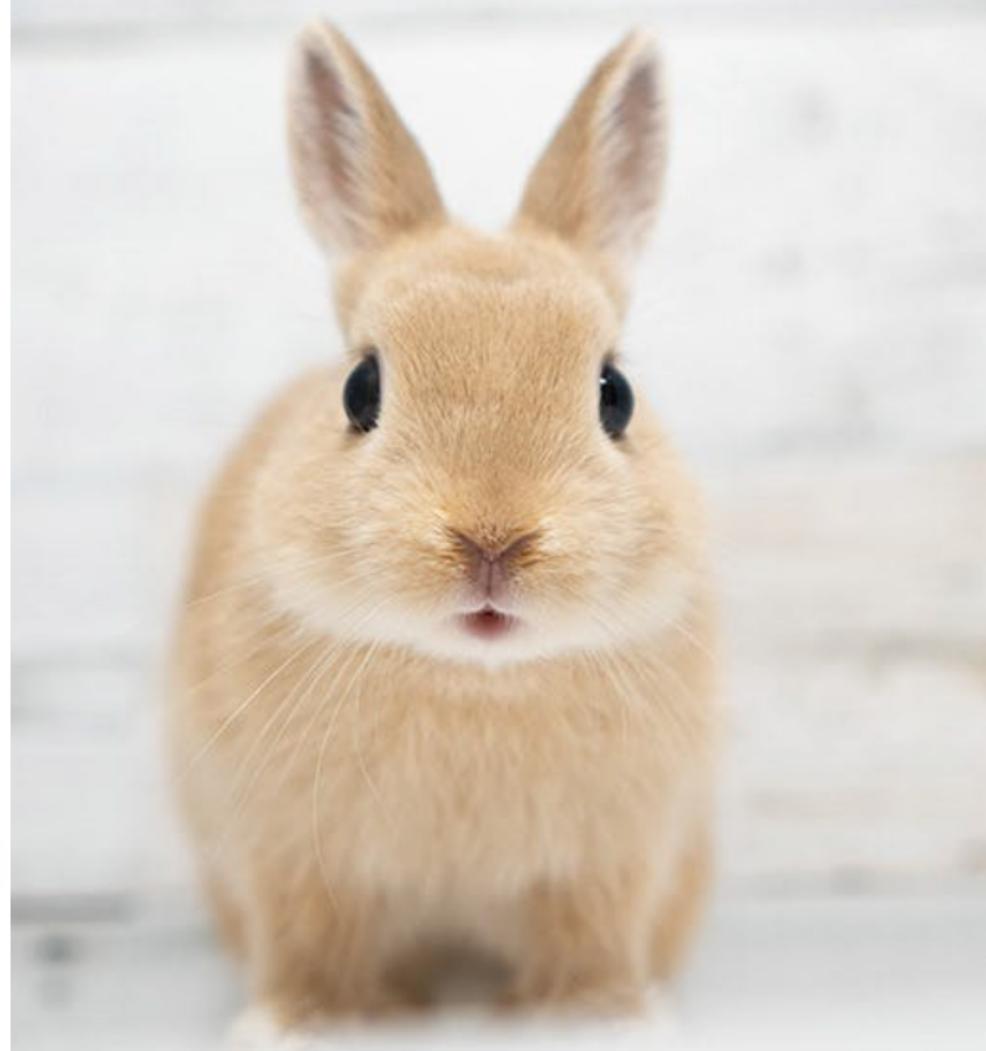
Email

Password

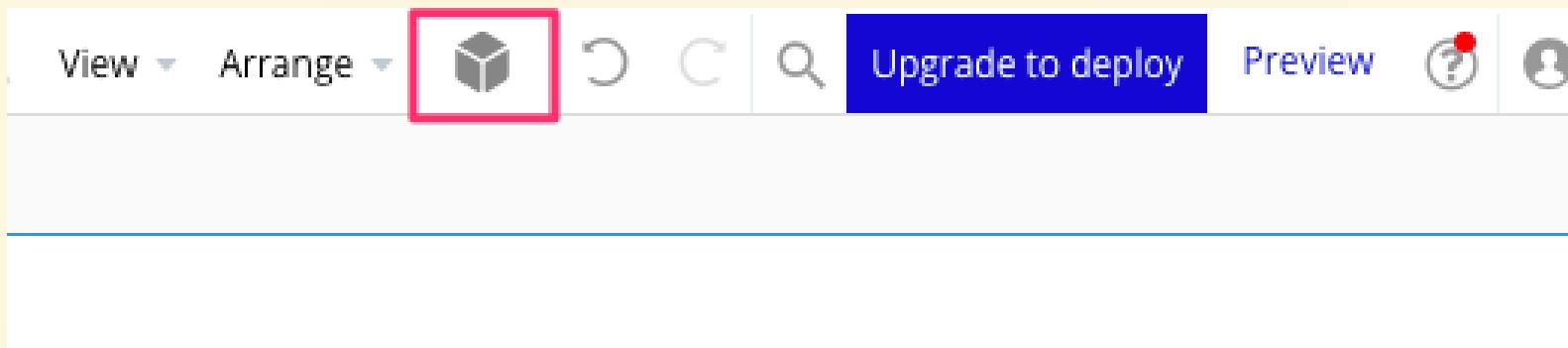
Re-enter Password

Sign Up

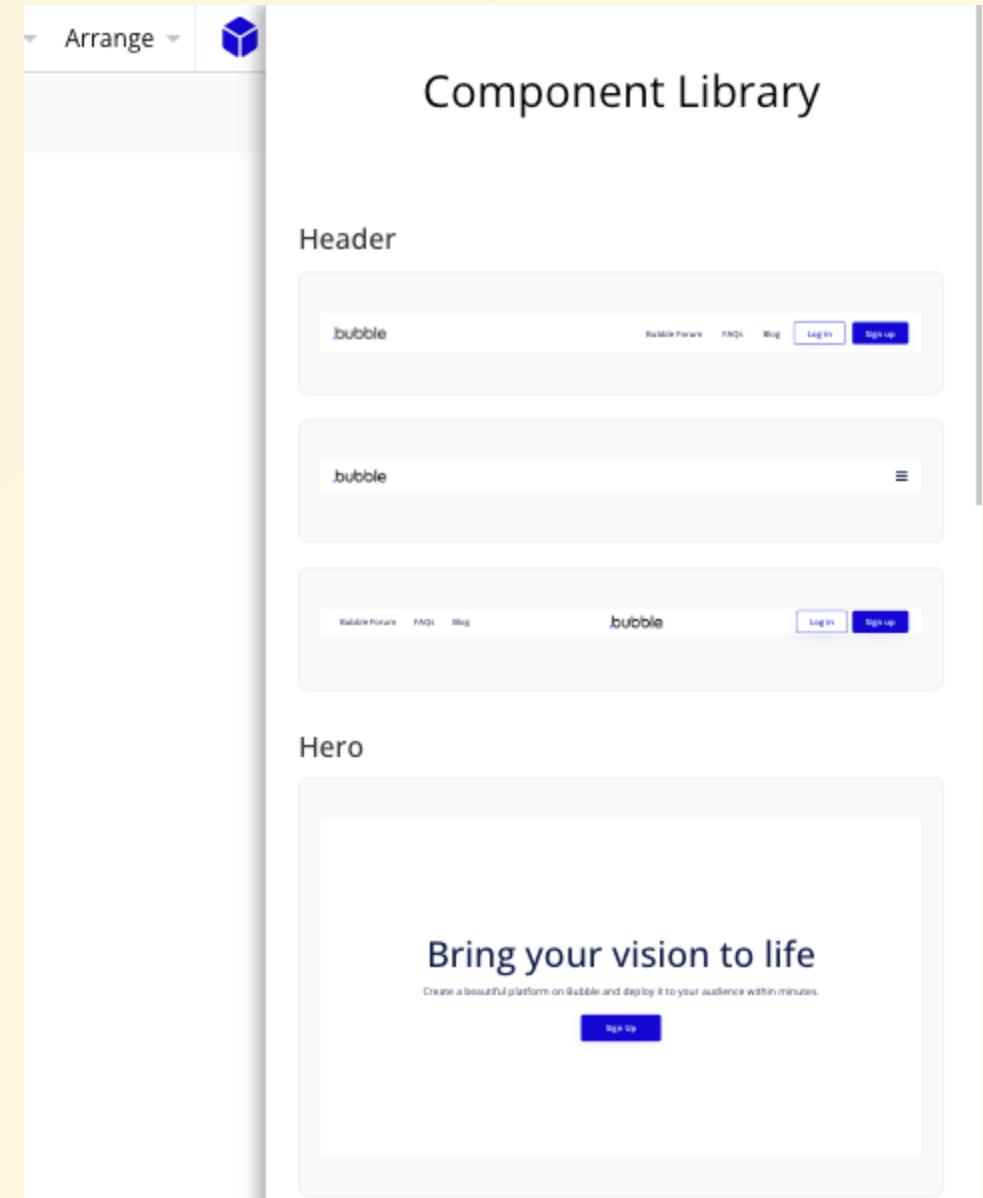
Already have an account? [Log In](#)



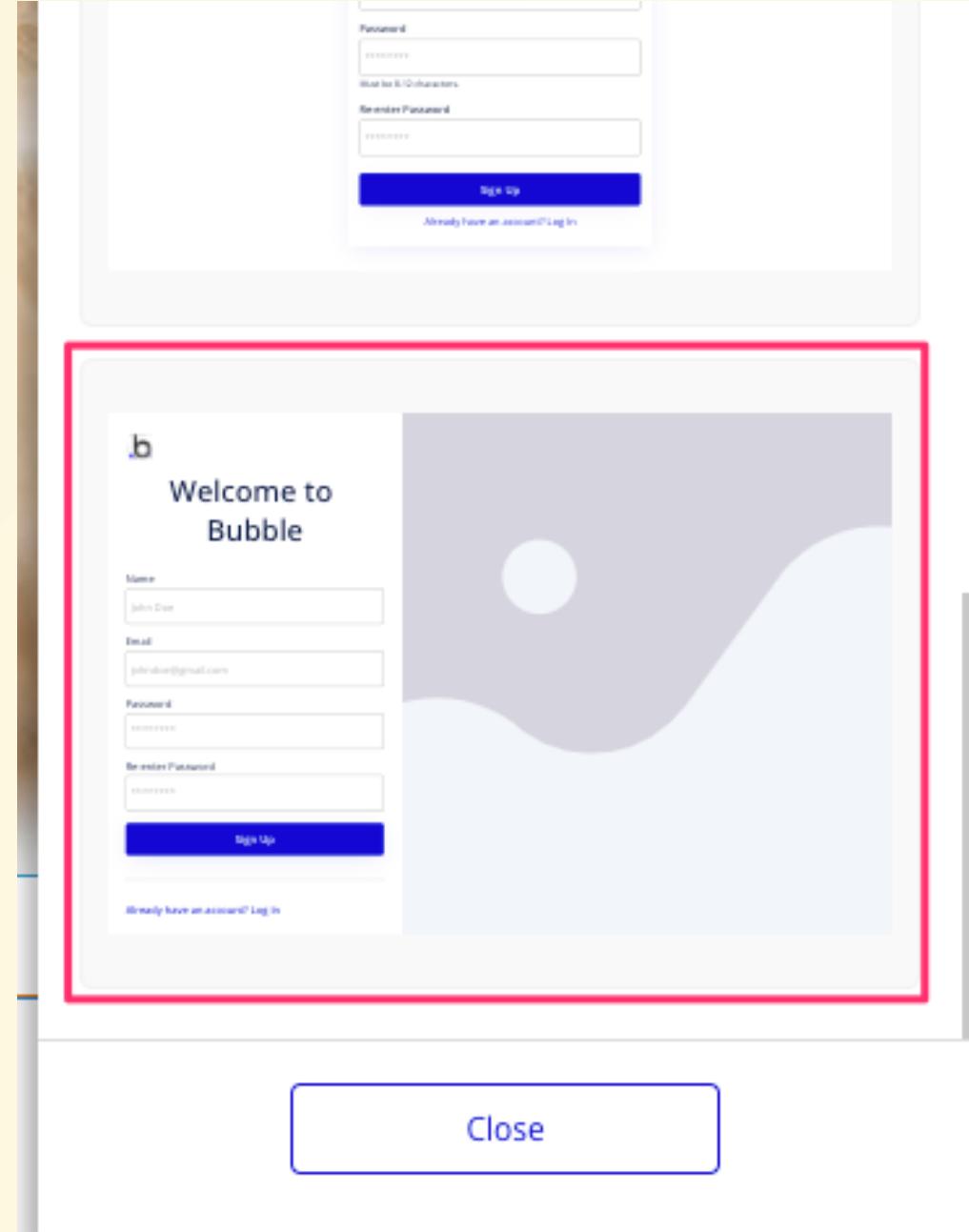
- First, let's place the signup and login components that come standard with Bubble.
- Click on the black square icon at the top of the screen.



- Then you will see an area called **Component Library** from the right side of the screen.
- This is literally a list of common components available in Bubble.



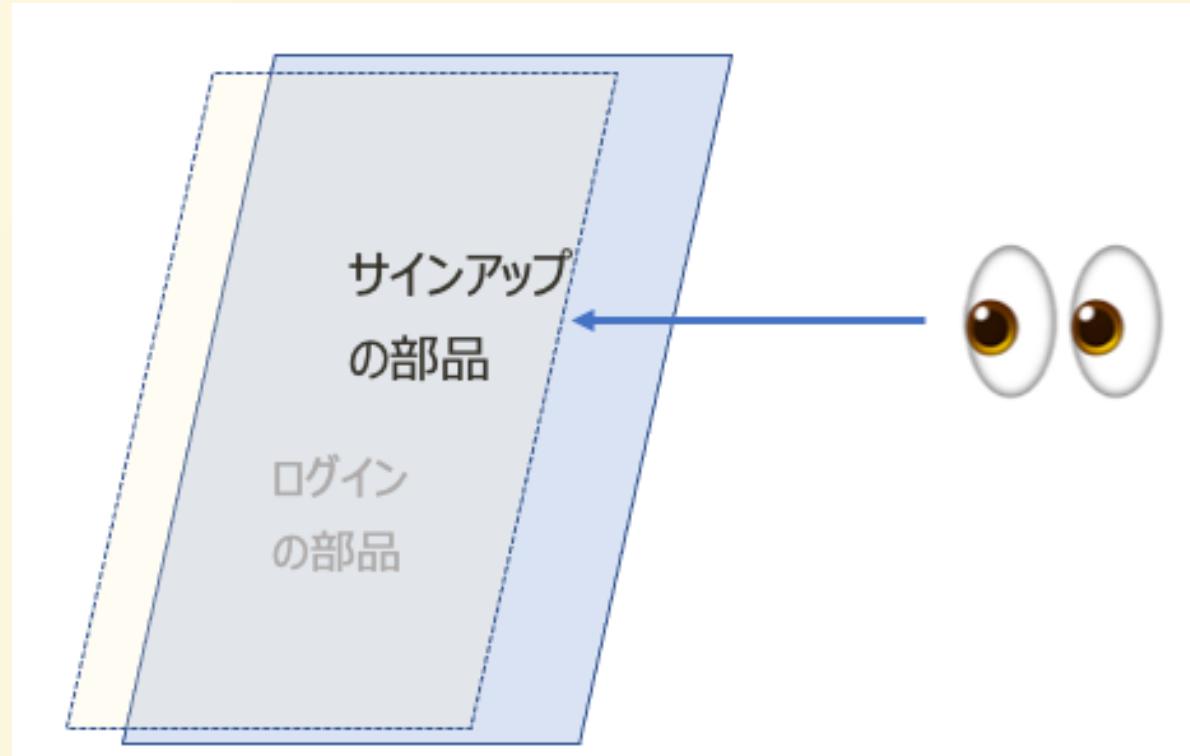
- At the bottom of this area, there is a common signup/login component with an image, drag & drop it to the Drawing area.



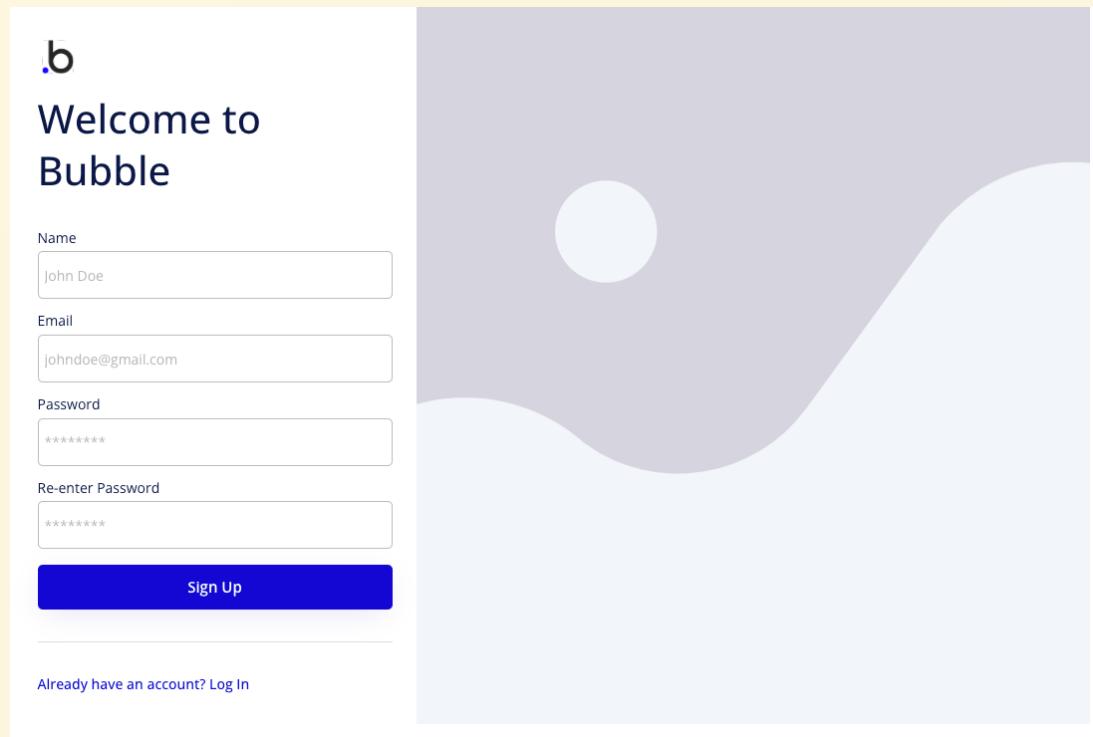
- Perhaps it could be positioned to fit perfectly at the top of the screen like this

The screenshot shows the Bubble UI Builder interface. On the left is the sidebar with various tools and components. The main area displays a two-column layout for a sign-up/login page. The left column contains a "Welcome to Bubble" header, input fields for Name (John Doe), Email (johndoe@gmail.com), Password, Re-enter Password, and a "Sign Up" button. The right column contains a "Sign Up / Log In" header, a "Sign Up" section with fields for Name, Email, Password, Re-enter password, and a "Sign Up" button, and a "Join a thriving community at Bubble." section with a "Log In" button. At the bottom of the left column is a link to "Already have an account? Log In".

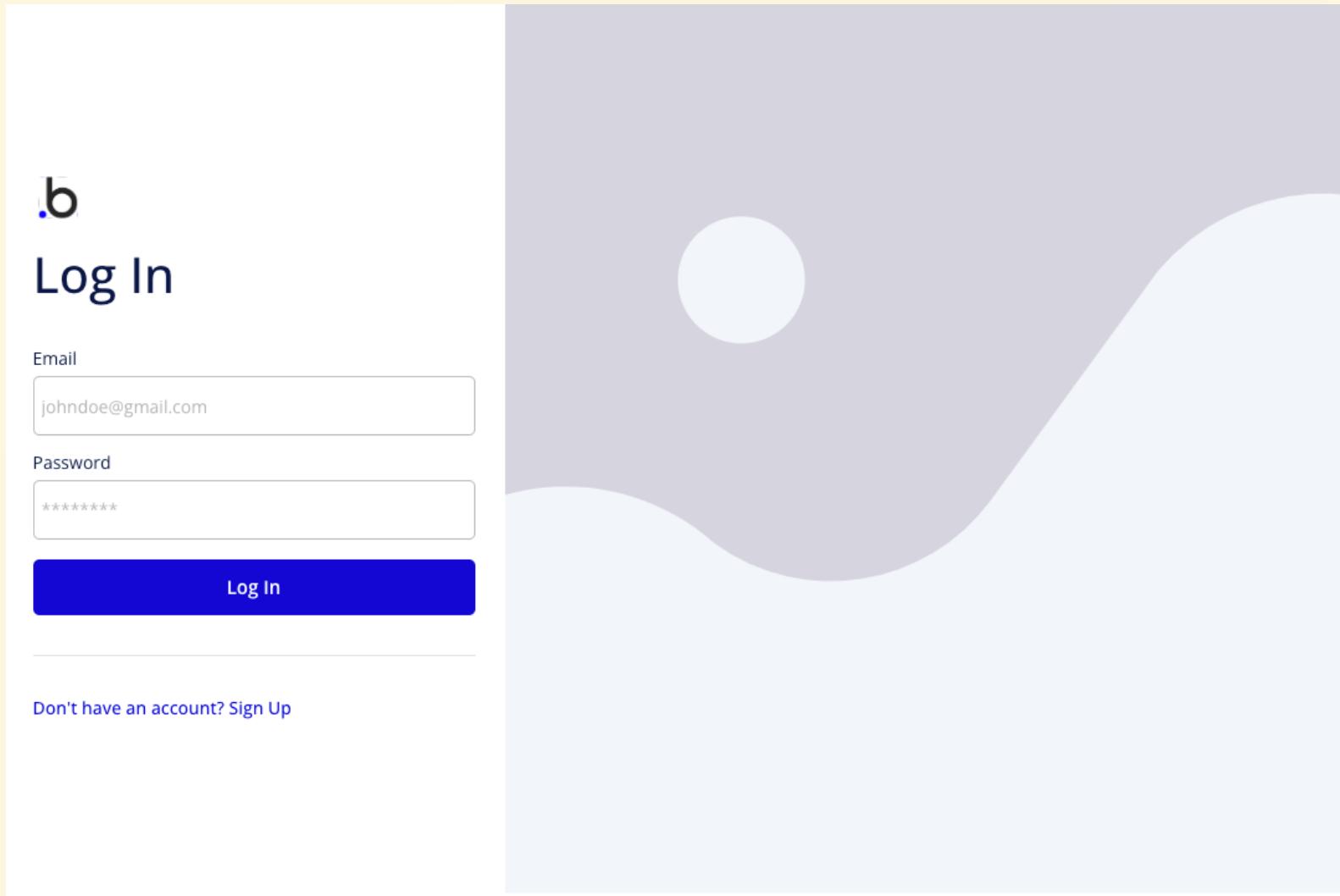
- ⚠ Be careful when moving this part, for example, because the "Sign Up" and "Login" parts are combined together.
- The way it works is that there is a "login" part on top of the "sign up" part that you can see now, but it is hidden.



- Let me run a preview here.
- At first you see the sign up component, but if you press the **Already have an account? Log In** link at the bottom, the content of the component switches to login.



- This is the login component



- As of yet, nothing happens when I press the "Sign Up" or "Log In" buttons.
- This is because we have not yet told Bubble what action we want it to take when the buttons are pressed.
- We will explain this setting later, so let's move on. 

About the 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 apps 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.

reference data

<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.
- We will discuss how to make Bubble applications responsive in the second Bubble lecture. 

Brief overview of application development with Bubble

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

3つのメイン操作

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



Design



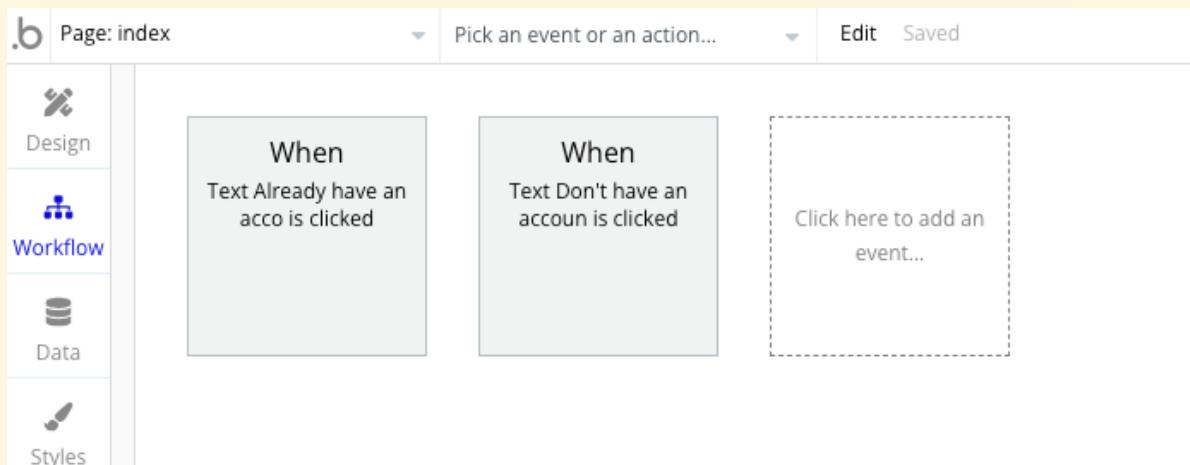
Workflow



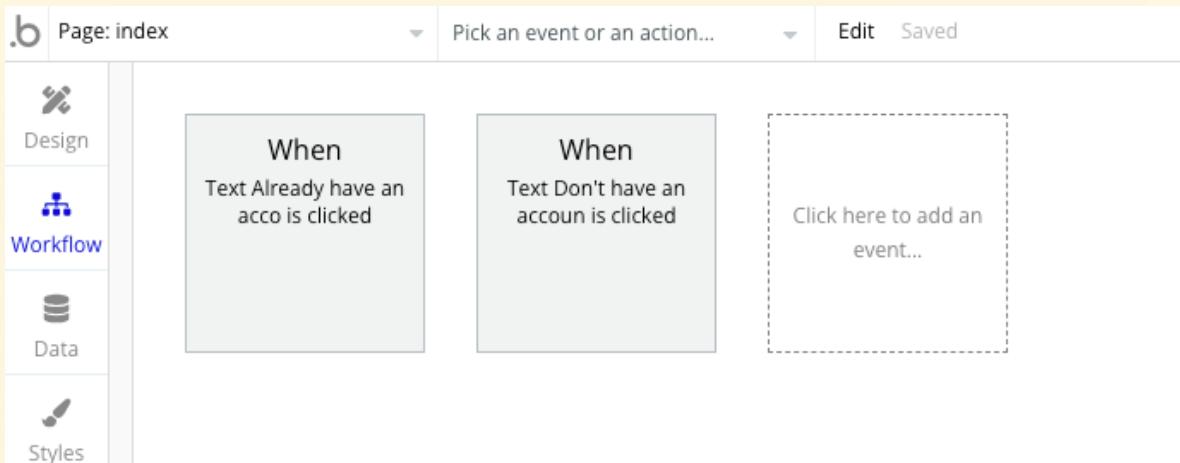
Data

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 workflow definition that was set up for the common sign-up / login component that we just used.
 - Specifically, this is the workflow that switches the display between the sign-up and login components.



Data

- Mode of defining and manipulating data

The screenshot shows a web-based application interface for managing data types. On the left, a vertical sidebar lists navigation options: Design, Workflow, Data (selected), Styles, Plugins, Settings, and Logs. The main content area has tabs for Data types, Privacy, App data, Option sets, and File manager, with 'Data types' selected. A search bar shows 'Data types'. Below it, a section for the 'User' data type is displayed, including fields for type name ('User'), email ('text'), Modified Date ('date'), Created Date ('date'), and Slug ('text'). A 'Create a new field' button is at the bottom. A note says 'Make this data type private by default' with a checkbox, and 'Things will be visible to everyone'.

- Select the "App data" tab and click on the "All Users" link
- You will then see the data of the list of registered users in the right panel, but since we have not yet created the sign-up function, I think there are still 0 users.

The screenshot shows the 'App data' section of a web application. On the left, a vertical sidebar lists navigation options: Design, Workflow, Data, Styles, Plugins, Settings, and Logs. The 'Data' option is currently selected. The main header includes a 'Page: index' dropdown, 'Edit' and 'Saved' buttons, a gift icon, a '0 issues' badge, and a 'Upgrade to deploy' button. Below the header, tabs for 'Data types', 'Privacy', 'App data' (which is active), 'Option sets', and 'File manager' are visible. The 'App data' section has a sub-header 'Application data - All Users - Development version' with links to 'Copy and restore database' and 'Switch to live database'. A row of buttons includes 'New view', 'Primary fields', 'Search', 'Data entries' (showing 0 entries), 'New entry', 'Delete (0)', 'Upload', 'Modify', 'Export', and 'Bulk'. A search bar labeled 'Views or data types' is present. The main area displays a table titled 'All Users' with columns: Email, Created Date, Modified Date, and Slug. The table is currently empty. A note indicates '2 additional fields'.

Other Operations

Other operations (menus) are also briefly explained

Styles

- Styles can be named for general use throughout the application (or individually for each component).

The screenshot shows the Bluebeam interface with the 'Styles' tab selected in the left sidebar. The main area displays a list of style definitions:

- 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 (Default)
- Dropdown - Standard Dropdown (Default)
- File Uploader - Standard Uploader (Default)
- Floating Group - Standard Floating Group (Default)

For the 'Standard Alert' style, there is a preview box labeled "...edit me..." and a 'Find all elements using this style' button. To the right, a detailed configuration panel is open, showing settings for Appearance, Layout, and Conditional styles. The Appearance section includes fields for Opacity (100%), Font (App Font: Open Sans), Size (600), and Text styling (B I U). The Layout section includes Word spacing (0), Line spacing (1), Letter spacing (0), and a checked checkbox for 'Center the text vertically'. The Conditional section includes a dropdown for 'Background style' set to 'Flat color' and a color swatch for 'Color' set to 'Primary (#0205D3)'. There is also a checkbox for 'Define each border independently'.

Plugins

- Just as Adalo added a line chart component, Bubble offers a variety of components to extend the application
- In Bubble, we call them "plugins".
- Bubble plugins can be free or paid for, so please check with us before using them!

Settings

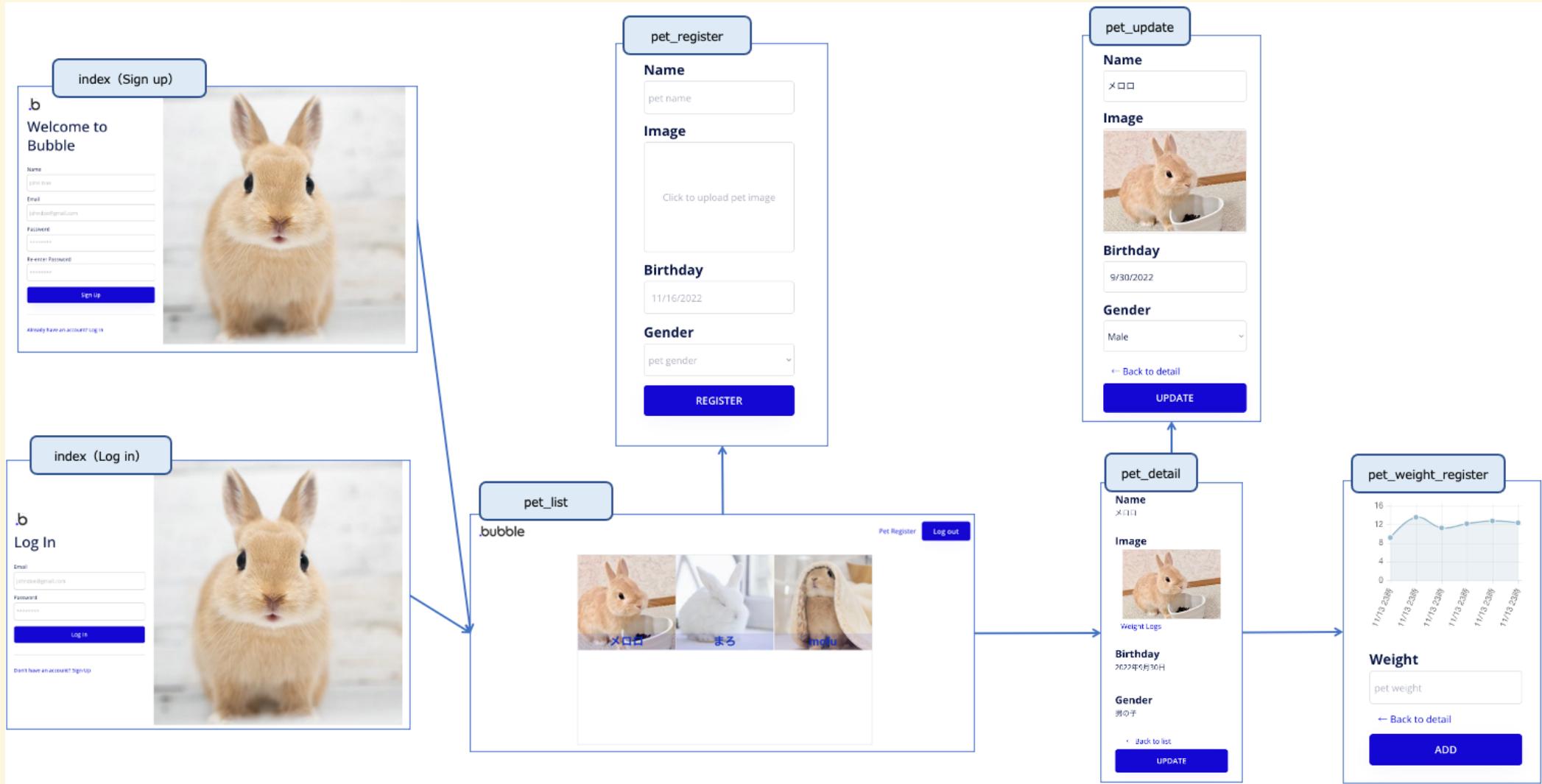
- Bubble's plan changes and account operations
- Not covered in this lecture

- 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.
- (This will not be 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.

Overall view of the screen we are going to make

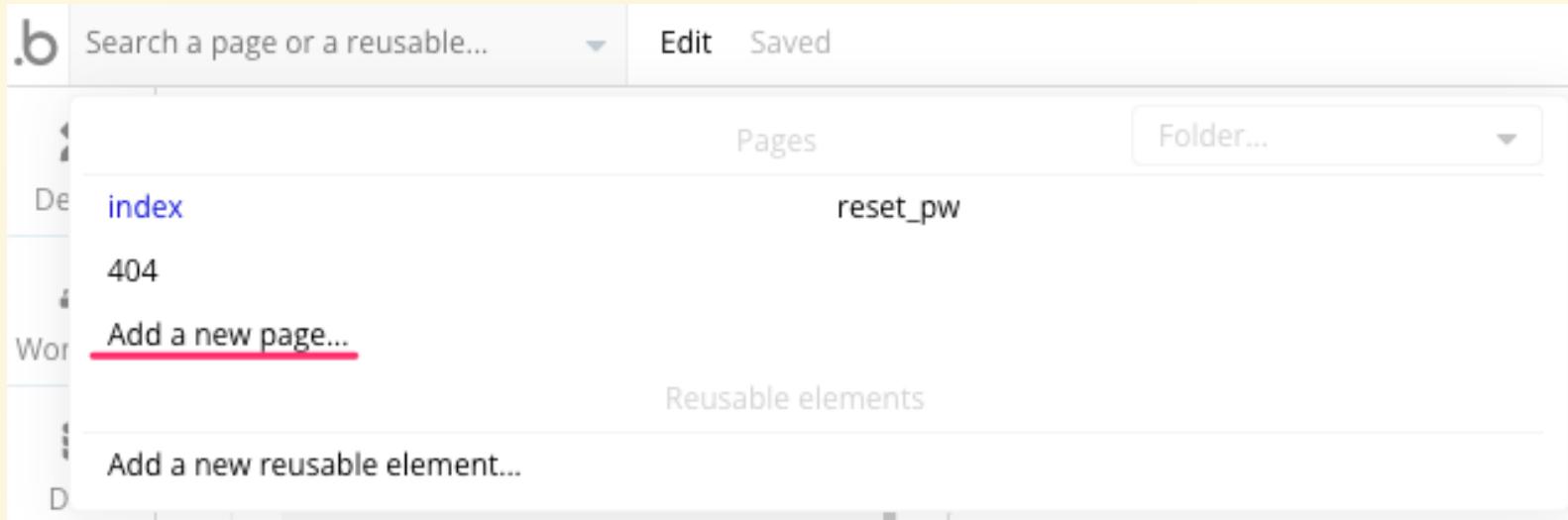


Let's make it from pet registration to listing

- 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 in the Bubble meeting ! !
- First, let's create a screen for pet registration and then let's make it to the point of displaying those pets in the list. !

We will create a pet registration screen

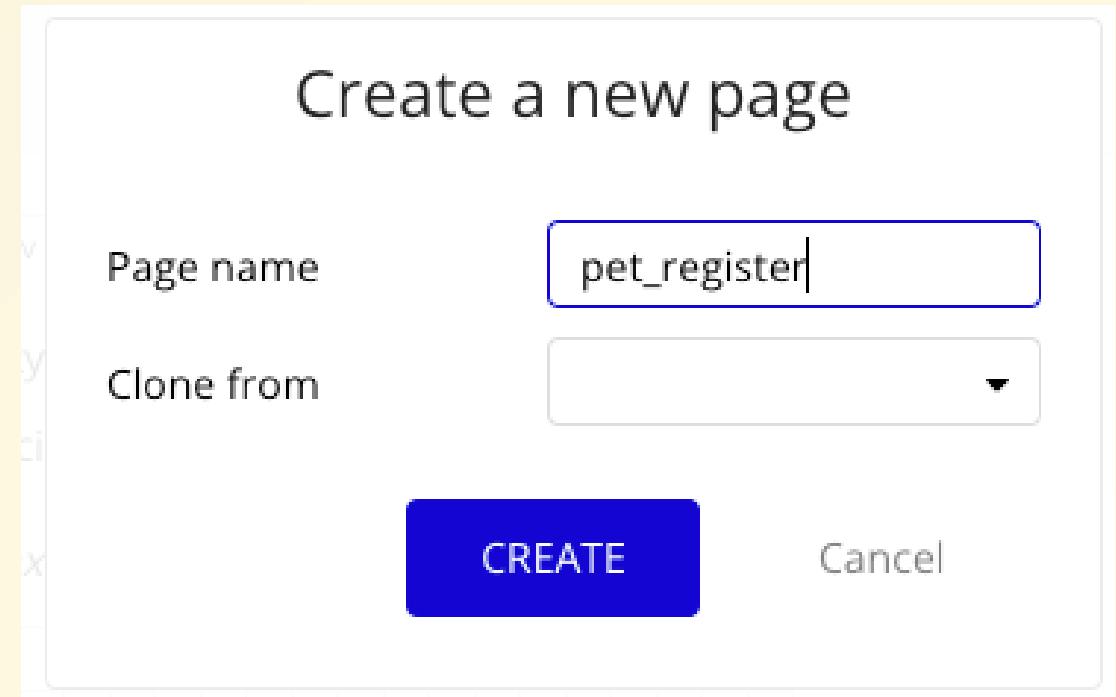
- Click **Page: index** in the upper left corner of the screen and a popup will appear.
- Click **Add a new page...** Click in the pop-up window.



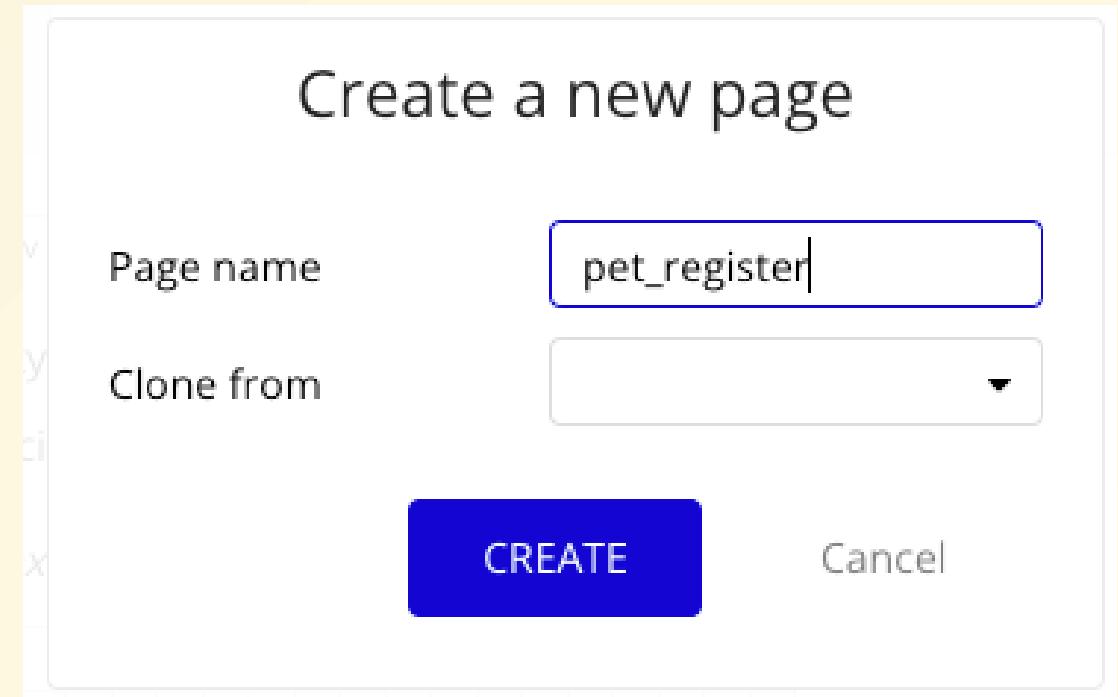
By the way...

- Here you can see a 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 will need to change the way you move from one screen to another from here .

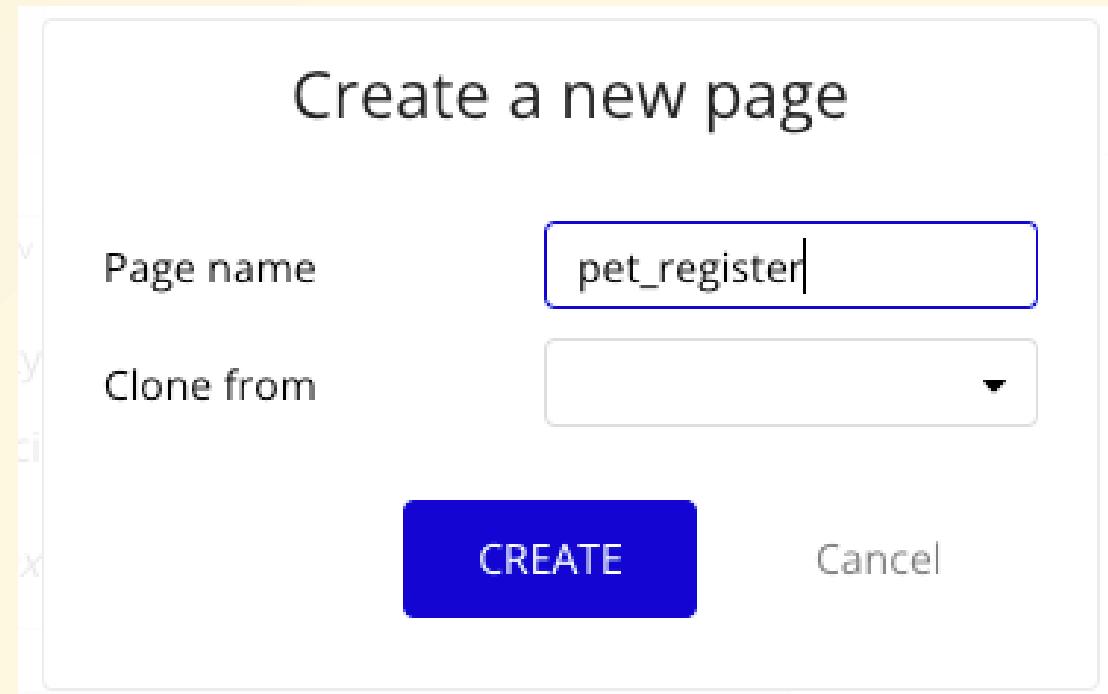
- Add a new page... to pop up a new page...
- The **Page name** is the name of the page, so enter only alphanumeric characters without spaces.
 - In this case, we will use **pet_register**.



- **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 is created.



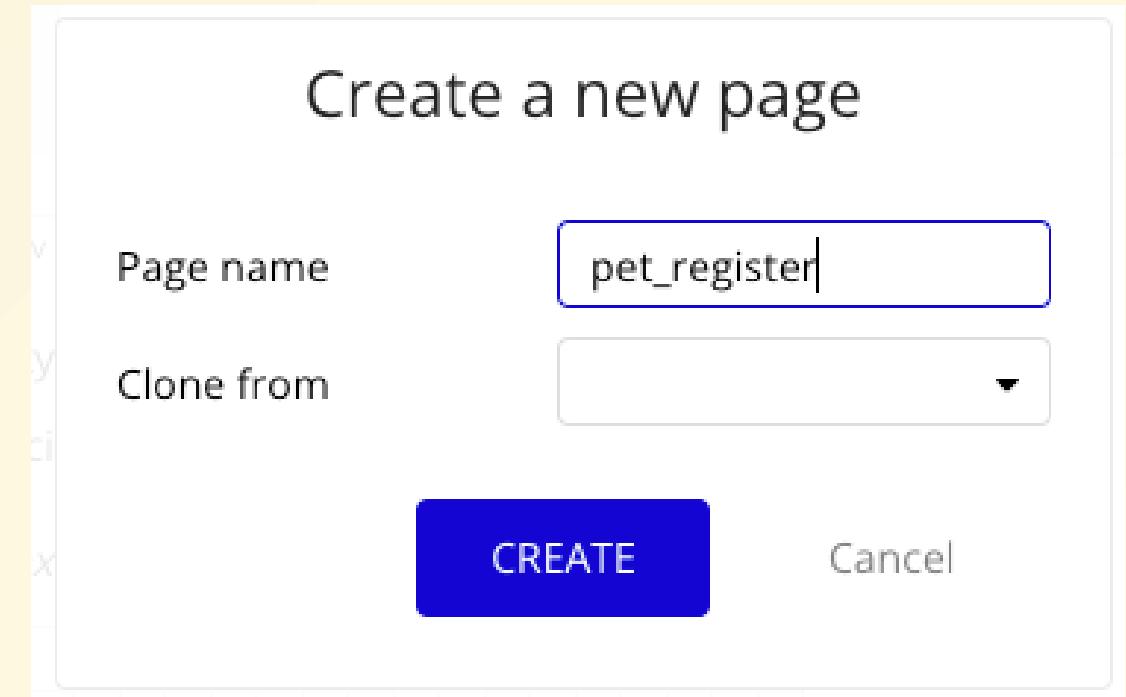
- Leave unselected for this time
- Click on the "CREATE" button

Create a new page

Page name

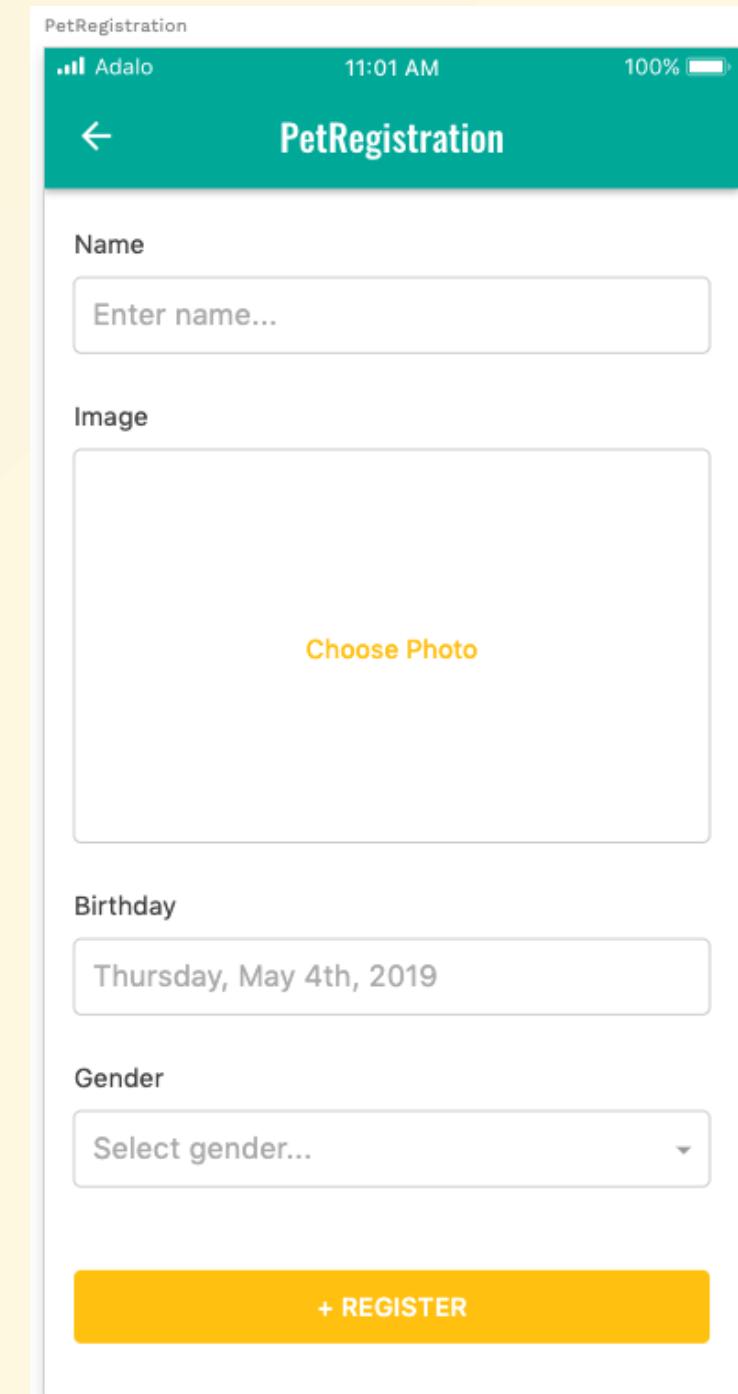
Clone from

CREATE Cancel



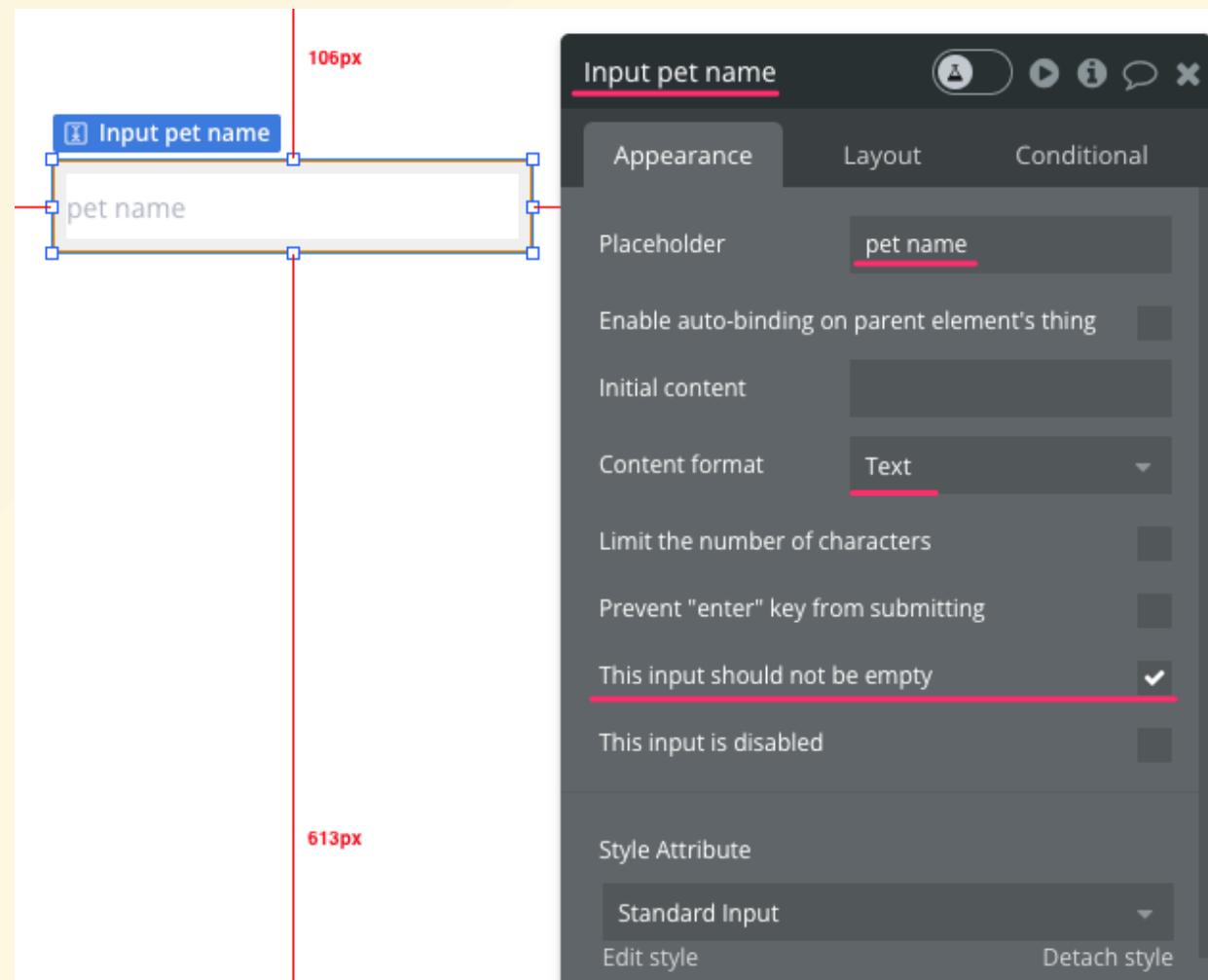
Assemble the pet registration screen

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

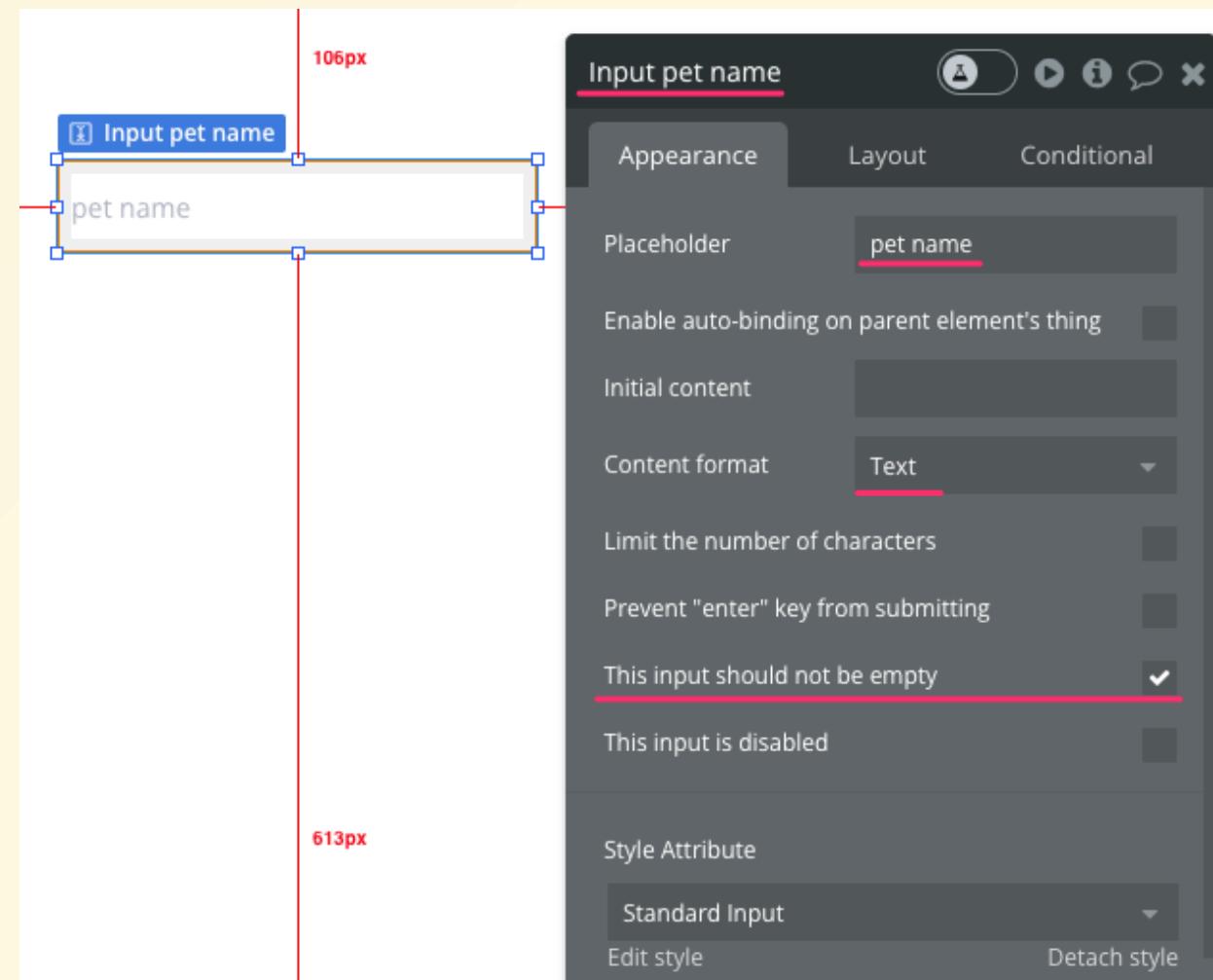


Pet Name

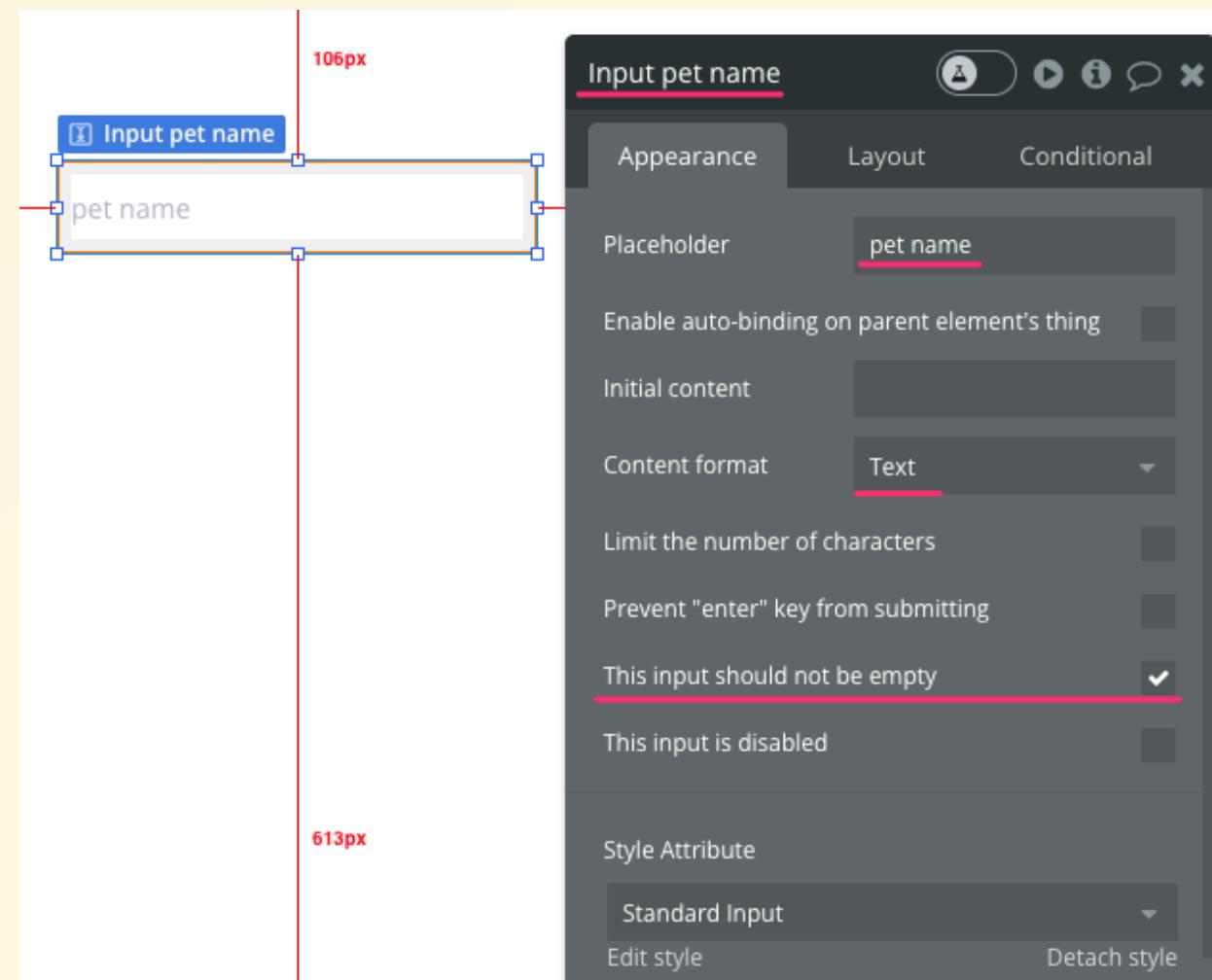
- First, place a text box for the pet's name
- Select **Input** in **Input forms** from the "UI Builder" in the left panel.
- Then, click on the place where you want to place the text box in the right panel, and the input box will appear.



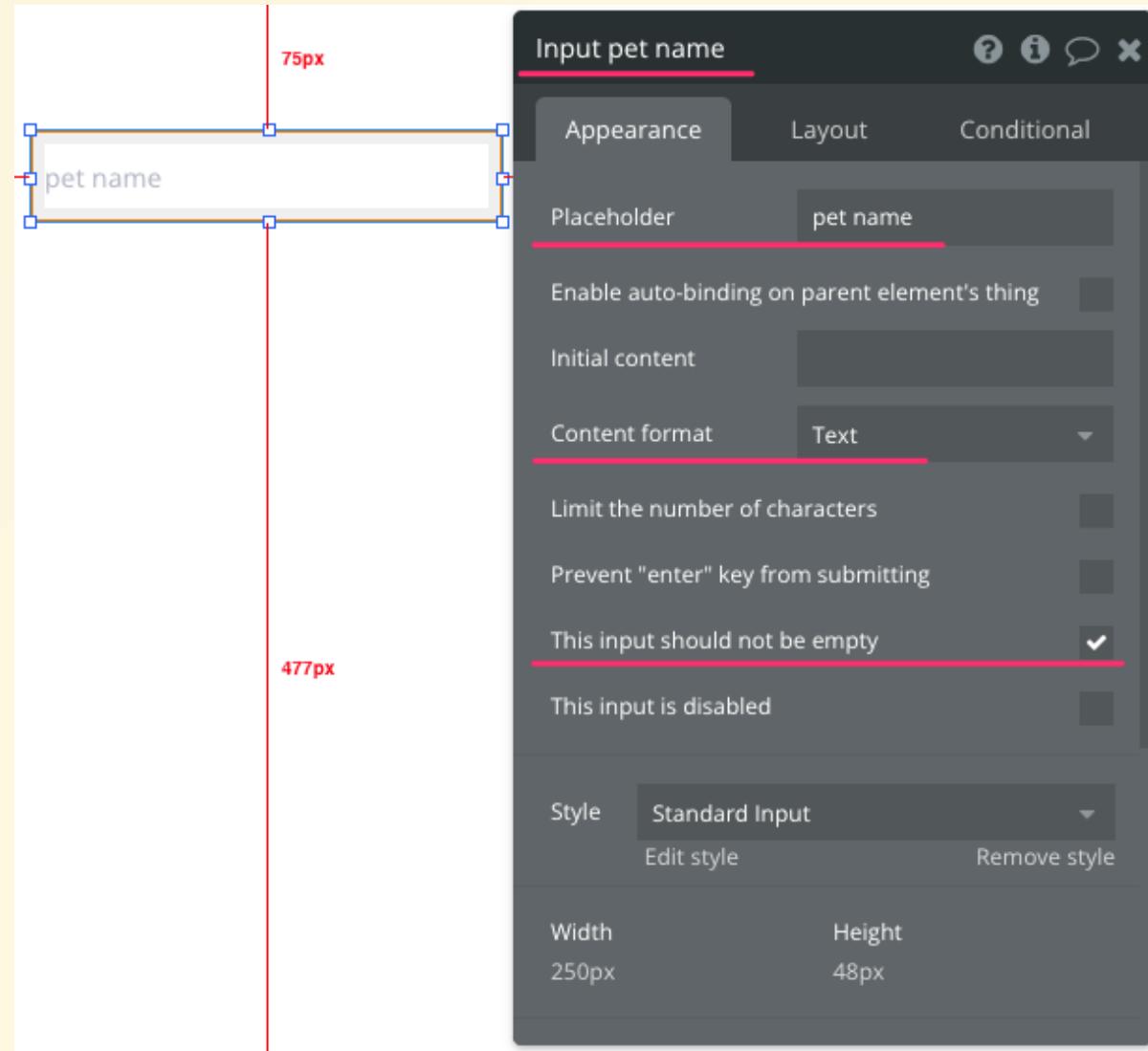
- Double-clicking on an element displays a dialog box where you can set various information about the element.
- Placeholder` is an auxiliary text that will be displayed when the text box is not filled in.
 - In this case, we will use "pet name".



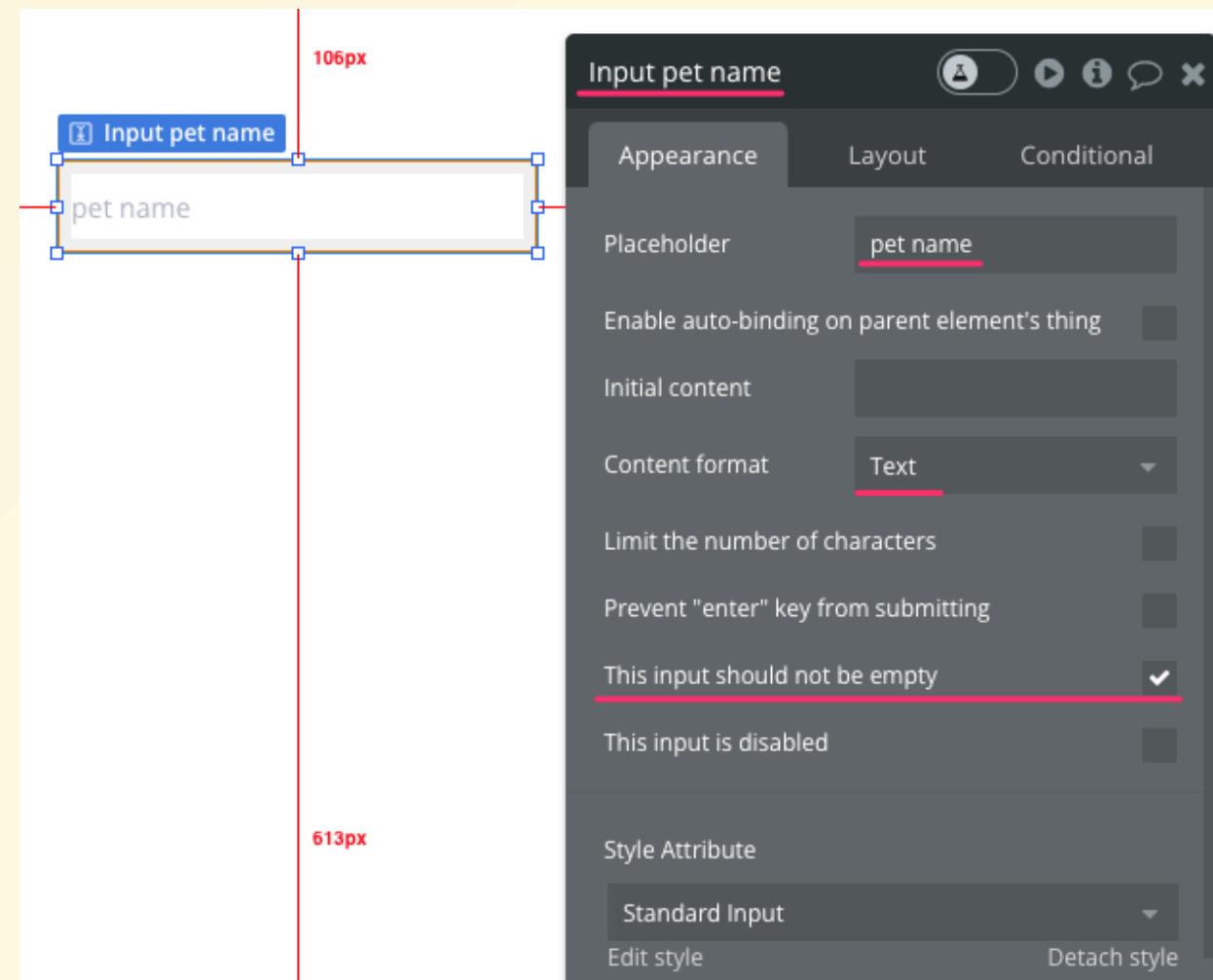
- The **Content format** allows you to specify the format of the value that can be entered into the text box
 - In this case, we will leave it as "Text" since we will be entering a string.



- 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.



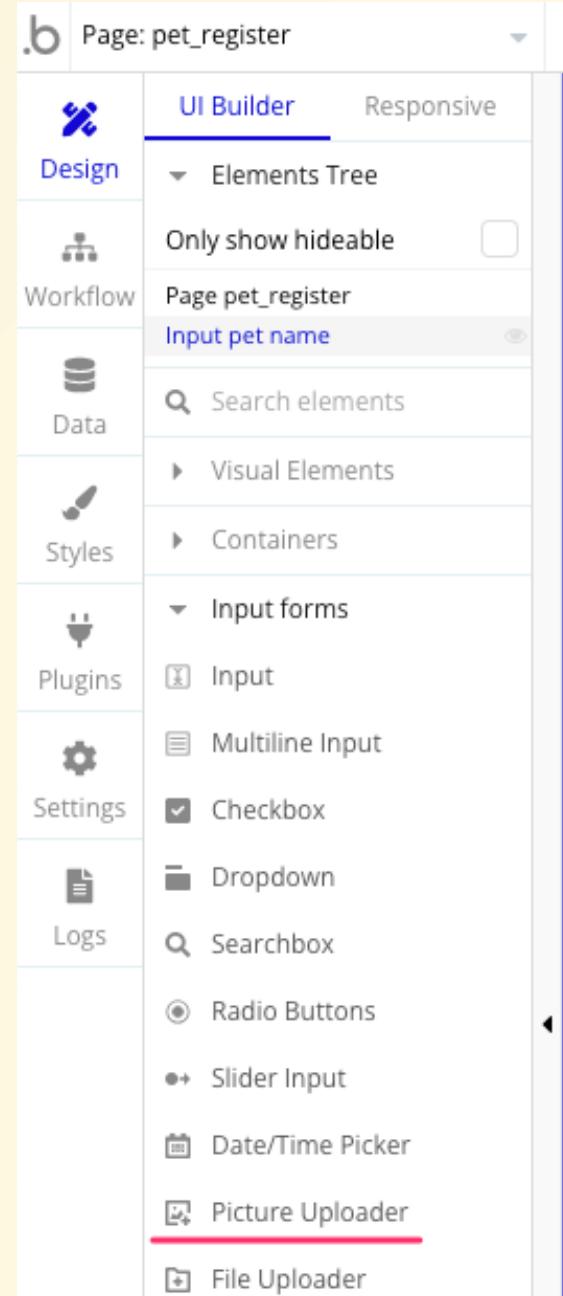
- If you click on the top of the dialog, you can name the elements in this text box
- This will come in handy later when defining the workflow, so let's specify "Input pet name" here.



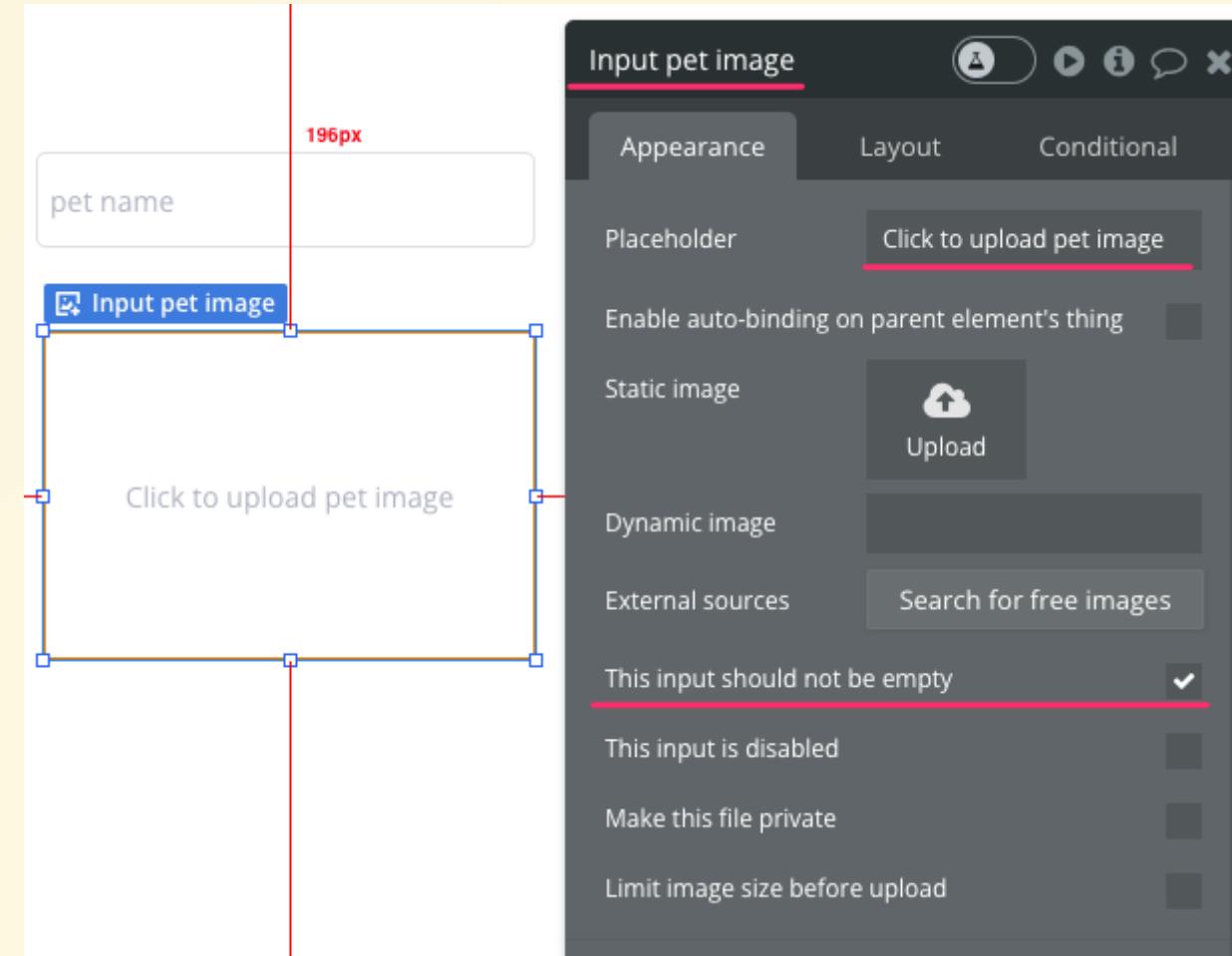
- After setting up to this point, you are ready to go.
 - You can also specify other details, but I won't explain them here.
 - If you are interested, please take a look .

Pet Images

- 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



- 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

.b Page: pet_register

UI Builder Responsive

Design Elements Tree

Workflow Only show hideable

Data Page pet_register

Input pet name

Input pet image

Styles Search elements

Visual Elements

Containers

Plugins Input forms

Input

Multiline Input

Checkbox

Dropdown

Searchbox

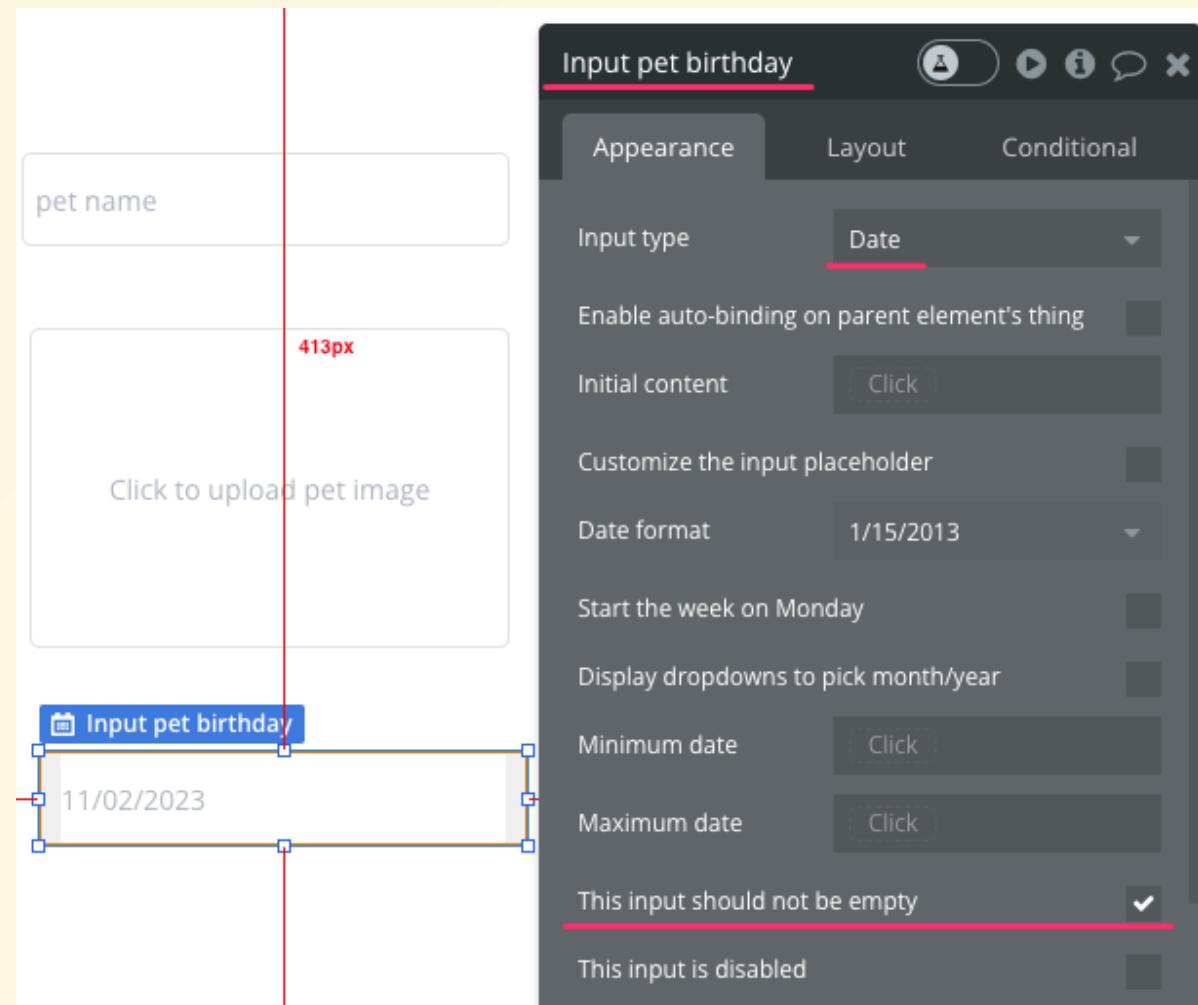
Radio Buttons

Slider Input

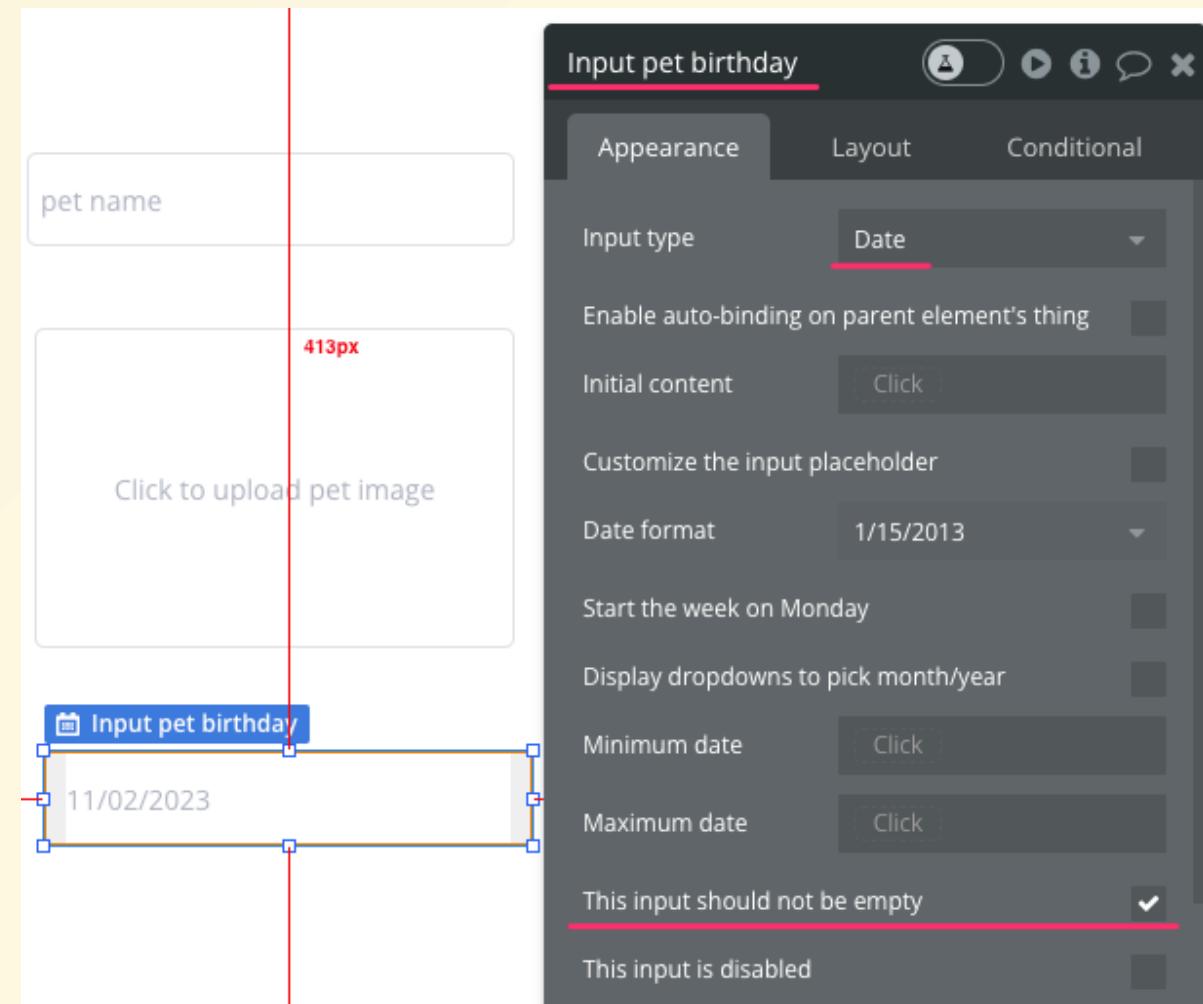
Date/Time Picker

Picture Uploader

- 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 will 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

.b Page: pet_register

UI Builder Responsive

Design

Workflow

Data

Styles

Plugins

Settings

Logs

Elements Tree

Only show hideable

Page pet_register

Input pet name

Input pet image

Input pet birthday

Search elements

Visual Elements

Containers

Input forms

Input

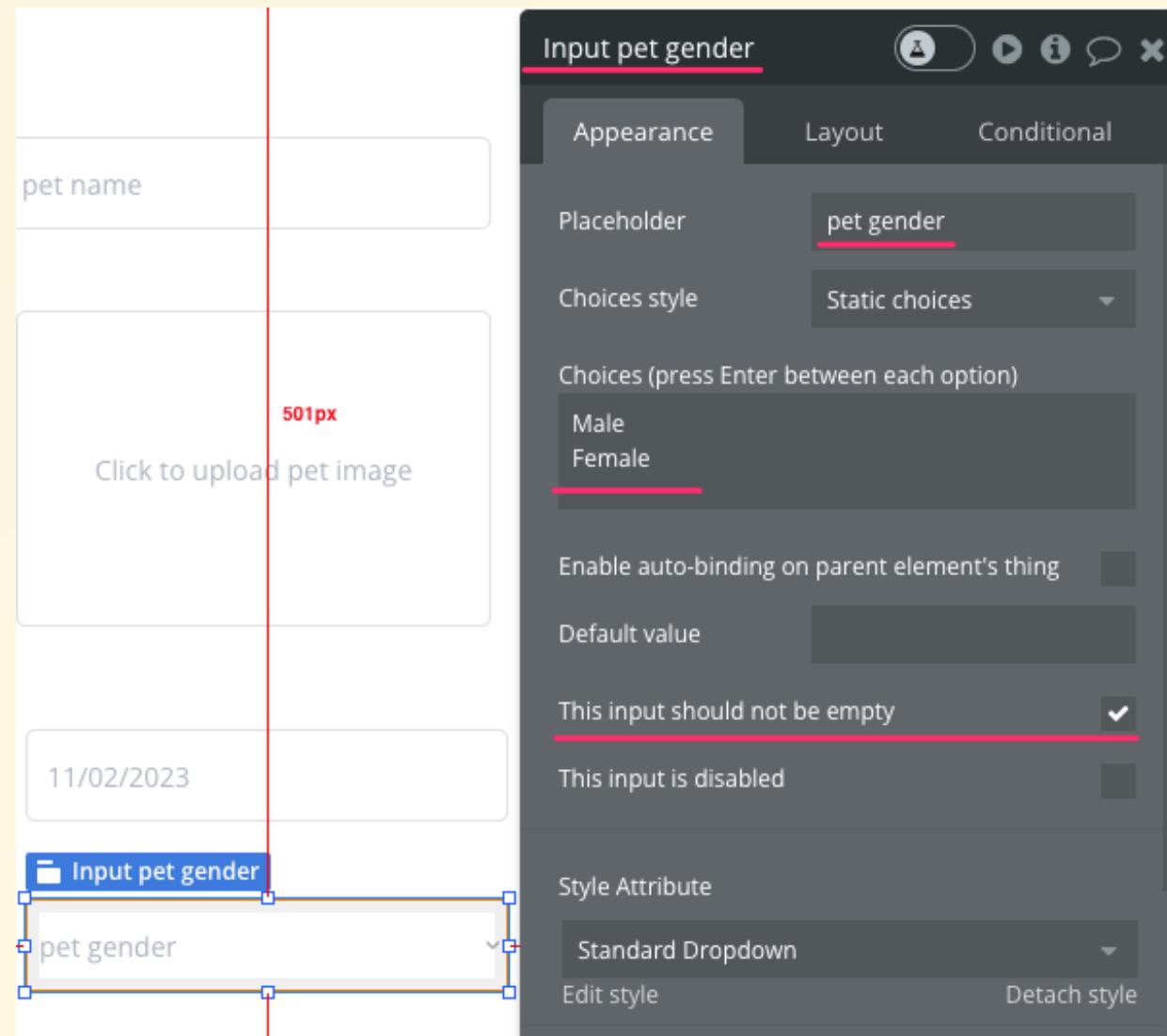
Multiline Input

Checkbox

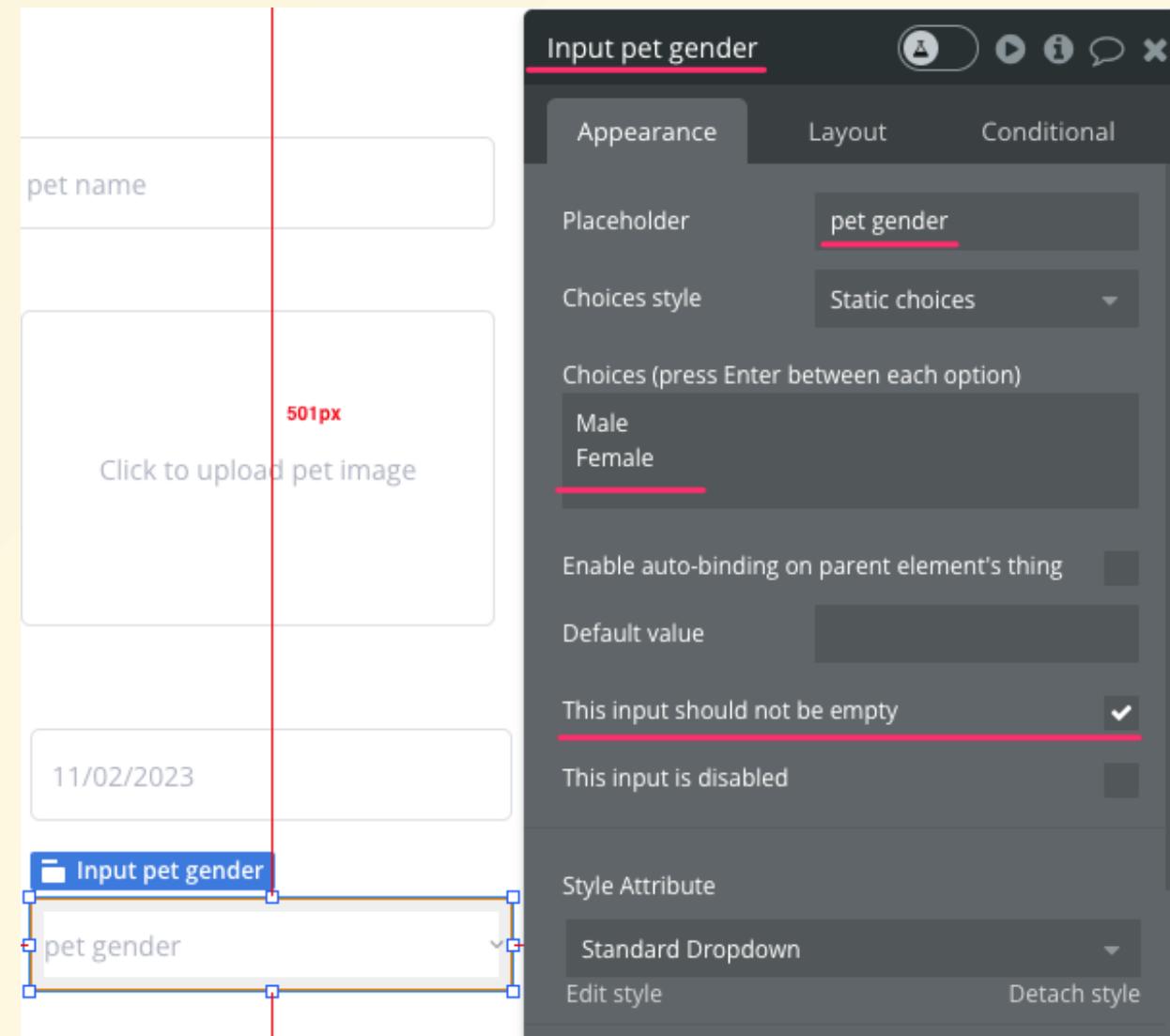
Dropdown

Searchbox

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

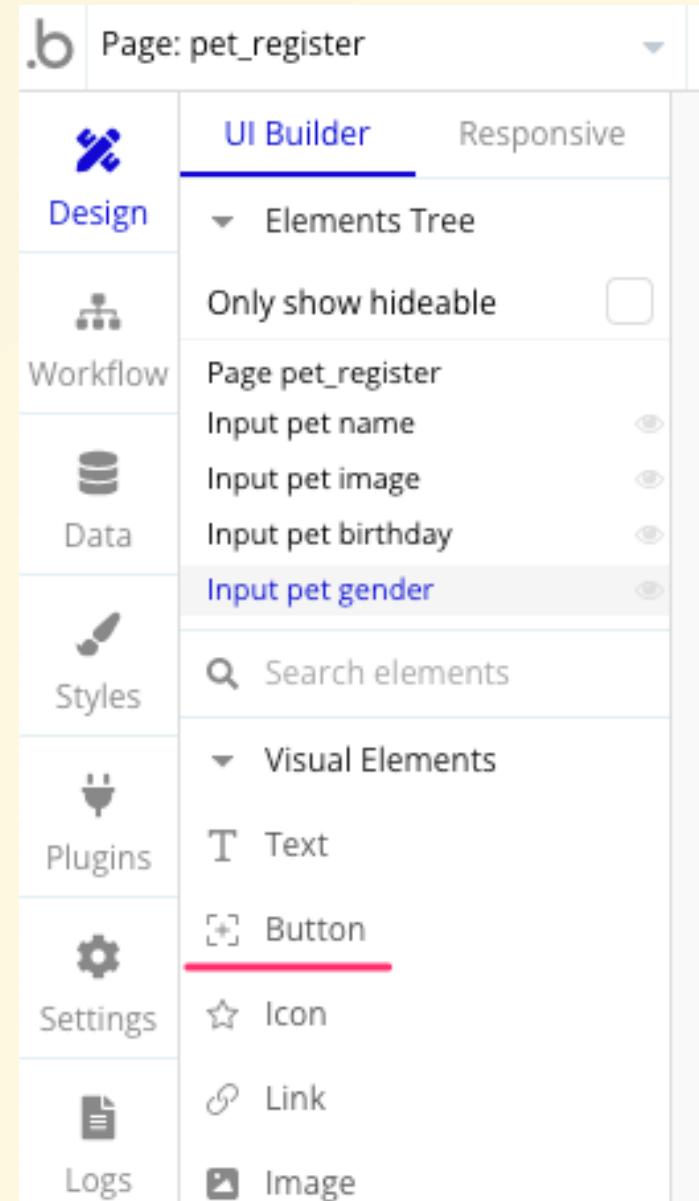


- Check This input should not be empty to make it mandatory.
- Specify Input pet gender as the name of the element.

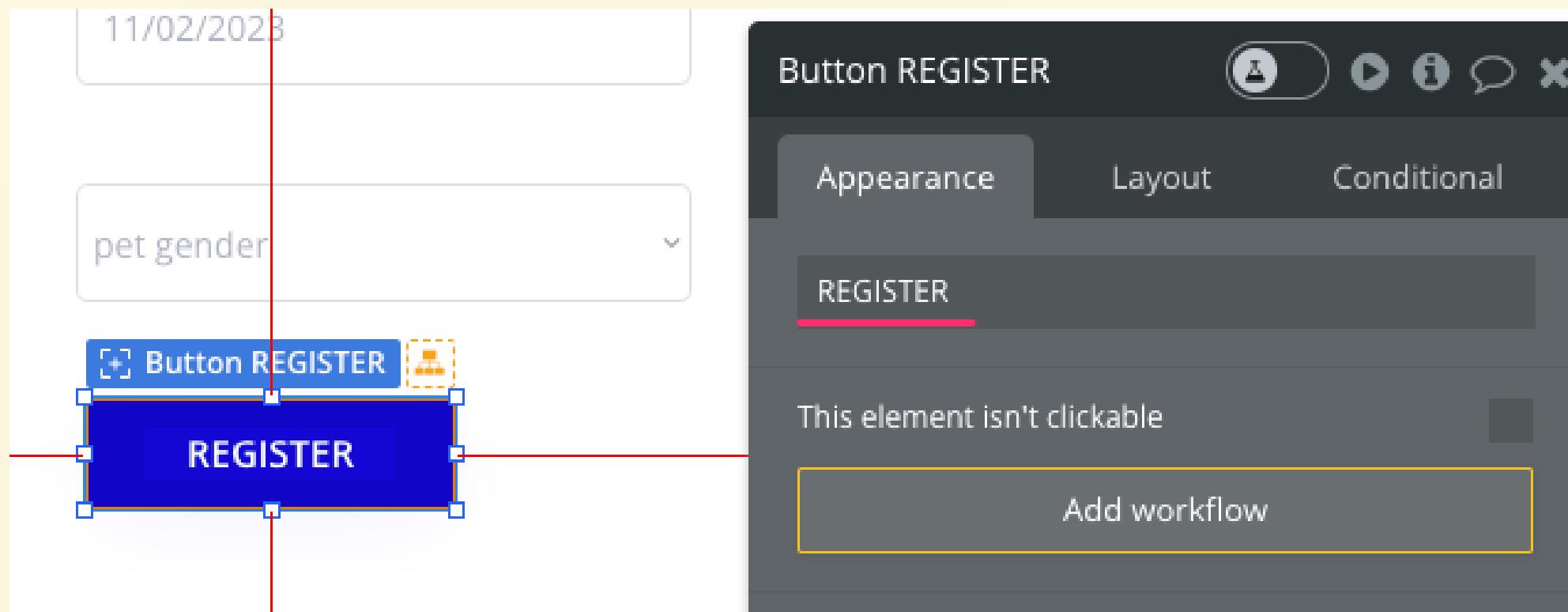


Registration Button

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

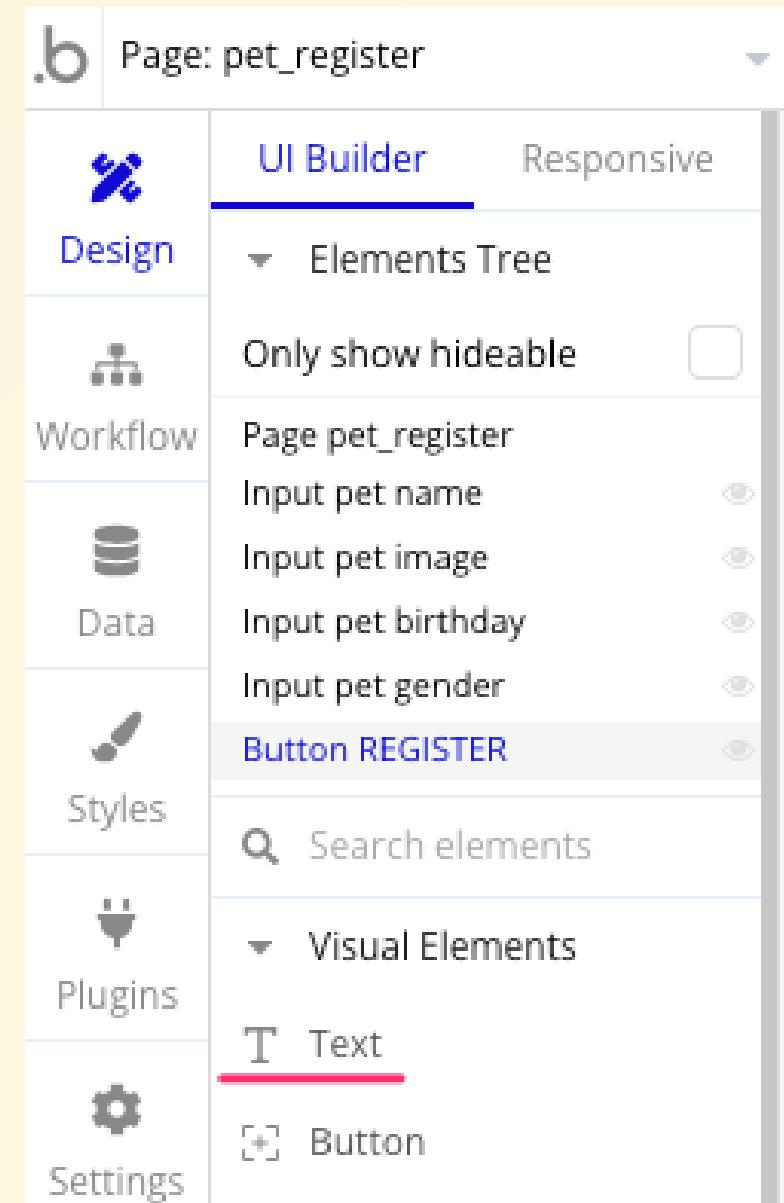


- Double-click on the element to set the details of the element
 - At the top of the Appearance tab, click ".... edit me..." at the top of the Appearance tab. at the top of the Appearance tab and type "REGISTER"

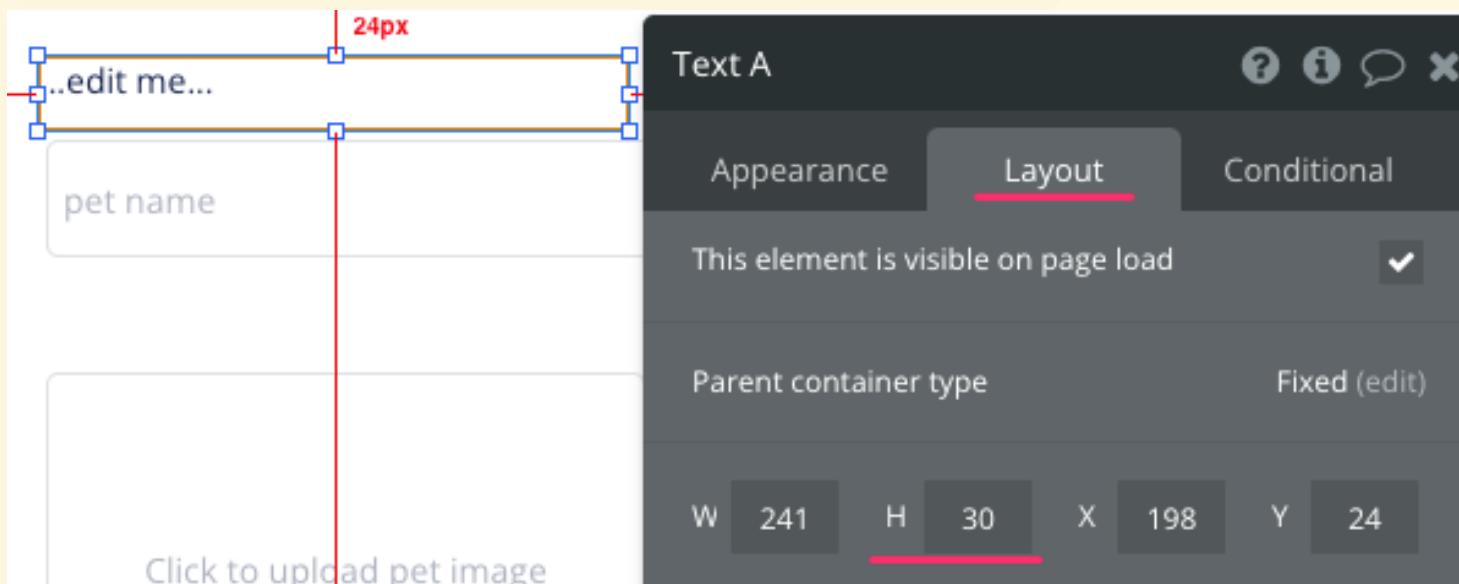


Let's label each input element

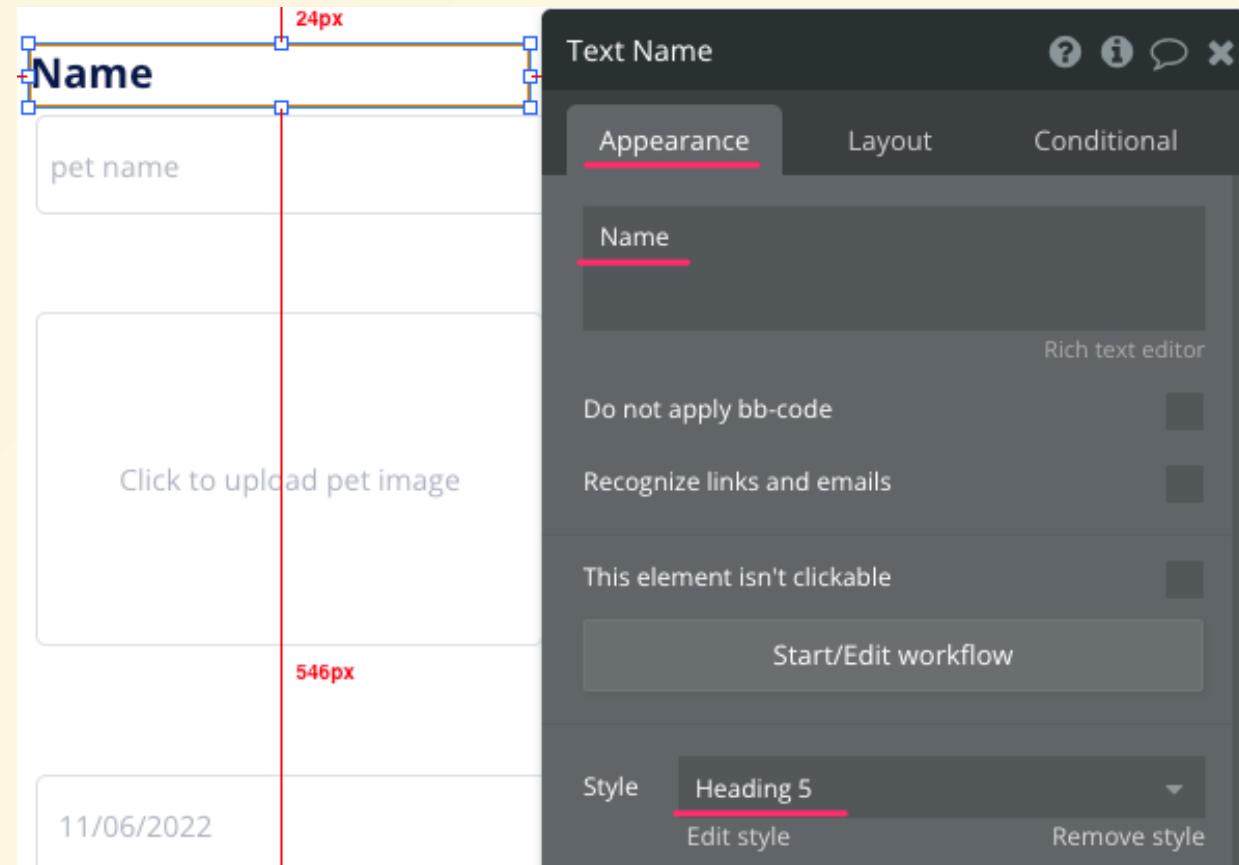
- Let's prepare a label in the upper left corner of each element, as shown in the Adalo screen
- Select **Text** in **Visual elements** in the left panel and drag the element in the right panel



- 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:

- Name**: A text input field containing the placeholder "pet name".
- Image**: A file upload input field with the placeholder "Click to upload pet image".
- Birthday**: A date input field showing the date "11/02/2023".
- Gender**: A dropdown menu with the placeholder "pet gender".

At the bottom is a large blue button labeled "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)

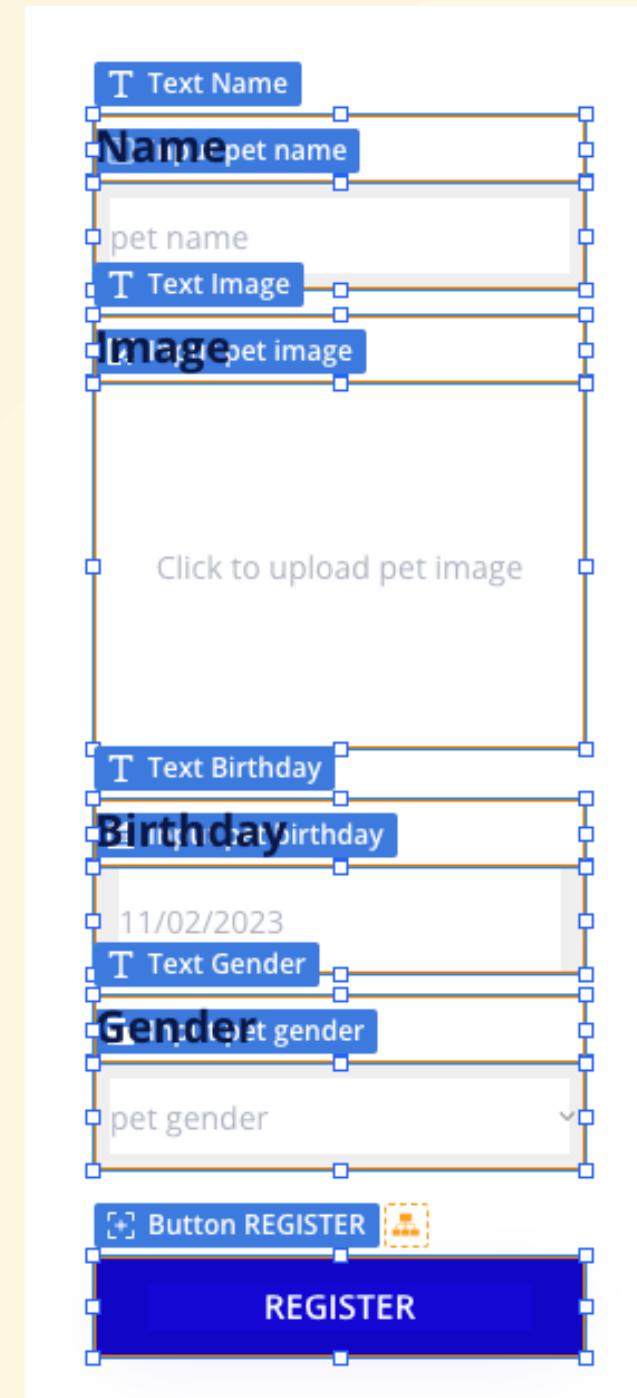
The image shows a user interface for registering a pet. It consists of several input fields and a final submission button. The fields are labeled with underlined text: "Name", "Image", "Birthday", and "Gender". The "Name" field contains the placeholder "pet name". The "Image" field is a box with the placeholder "Click to upload pet image". The "Birthday" field contains the date "11/02/2023". The "Gender" field is a dropdown menu with the placeholder "pet gender". At the bottom is a large blue button with the word "REGISTER" in white capital letters.

| | |
|----------|---------------------------|
| Name | pet name |
| Image | Click to upload pet image |
| Birthday | 11/02/2023 |
| Gender | pet gender |

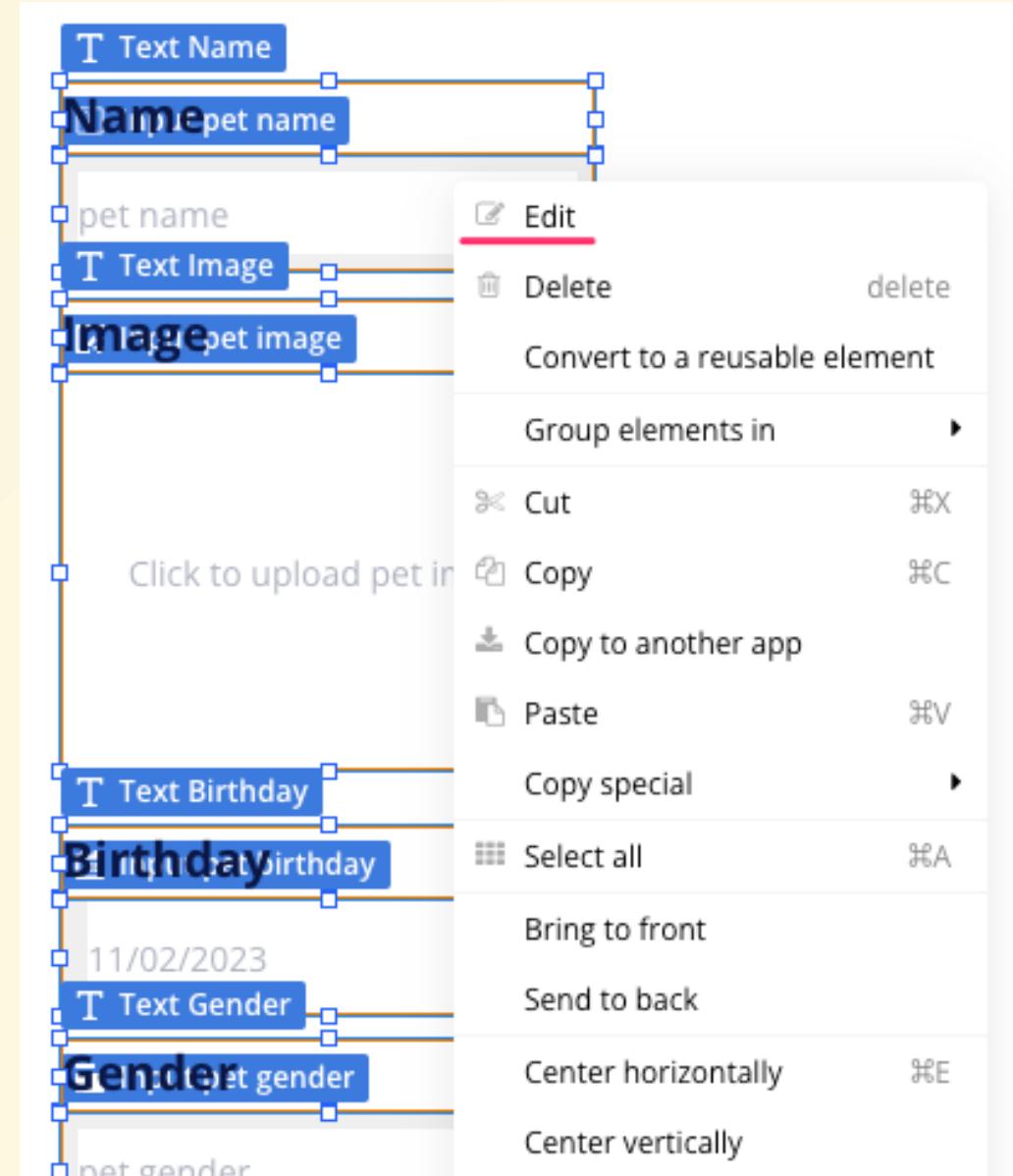
REGISTER

- 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.

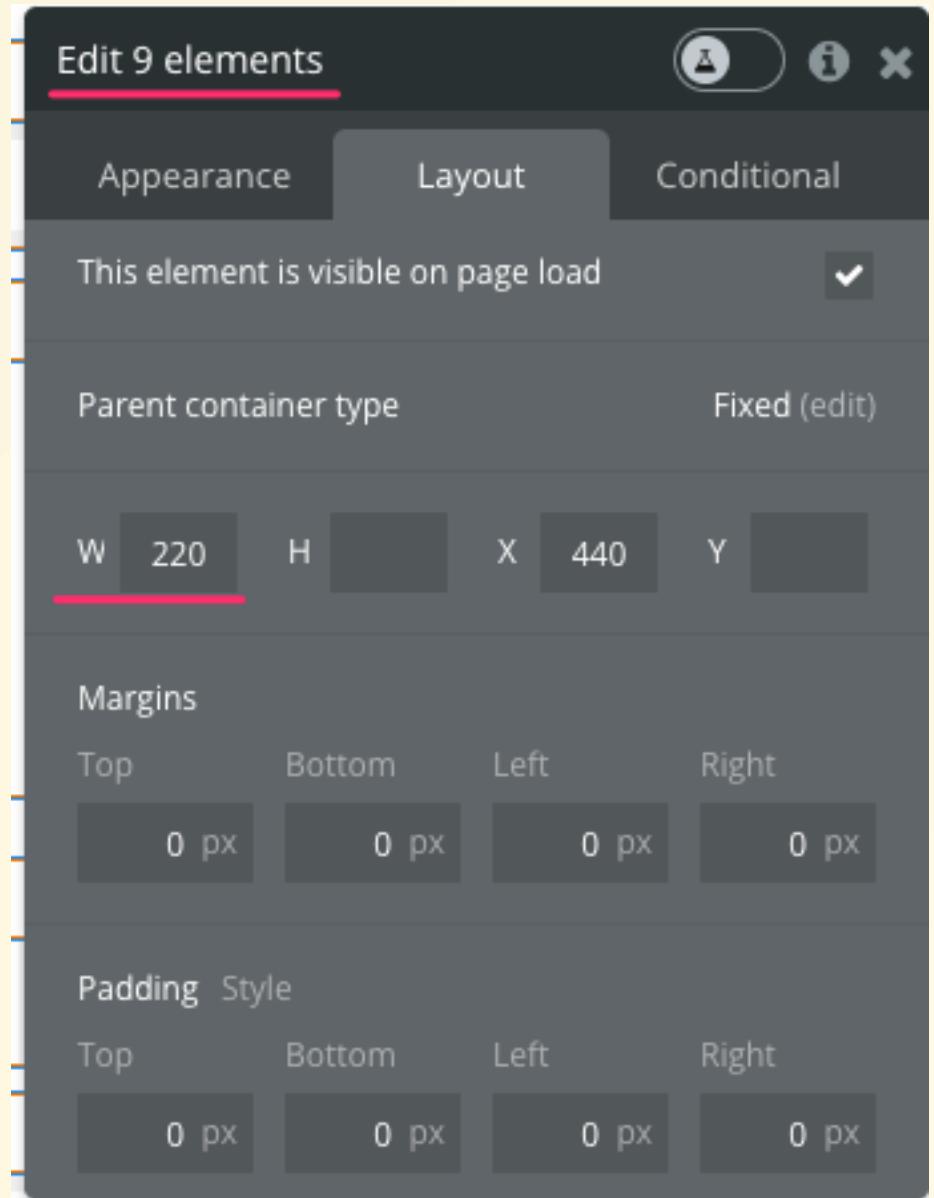
- You can change them one by one, but since there are so many elements, let's change them all at once!
- With all elements selected, right-click and select Edit
 - Select all elements by dragging



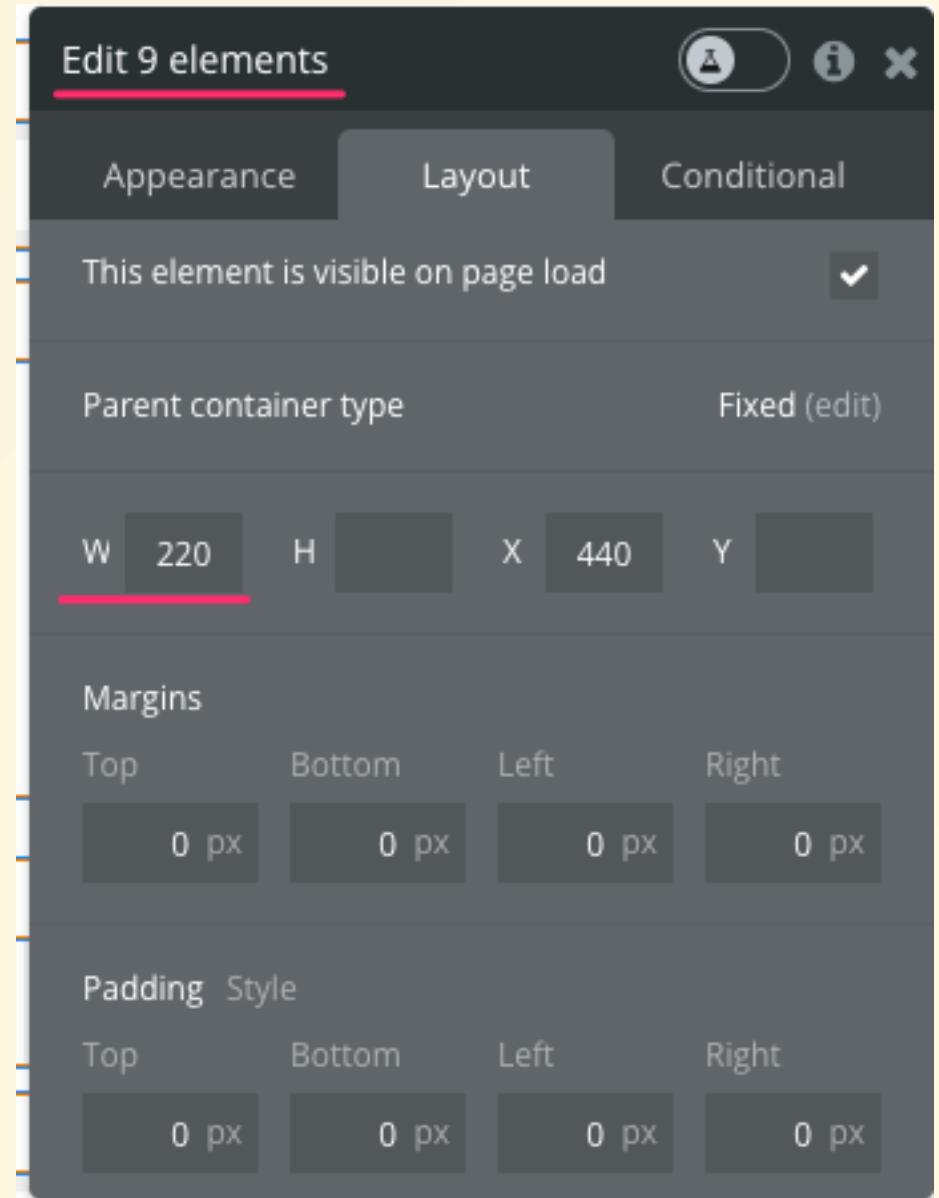
- With all of the elements selected, right-click on any one of them.
- Select Edit from the submenu that appears.



- Then a dialog box will appear, confirming that the number of selected elements is 9
- If all is well, open the Layout tab in the dialog and click on the W (Width) part.



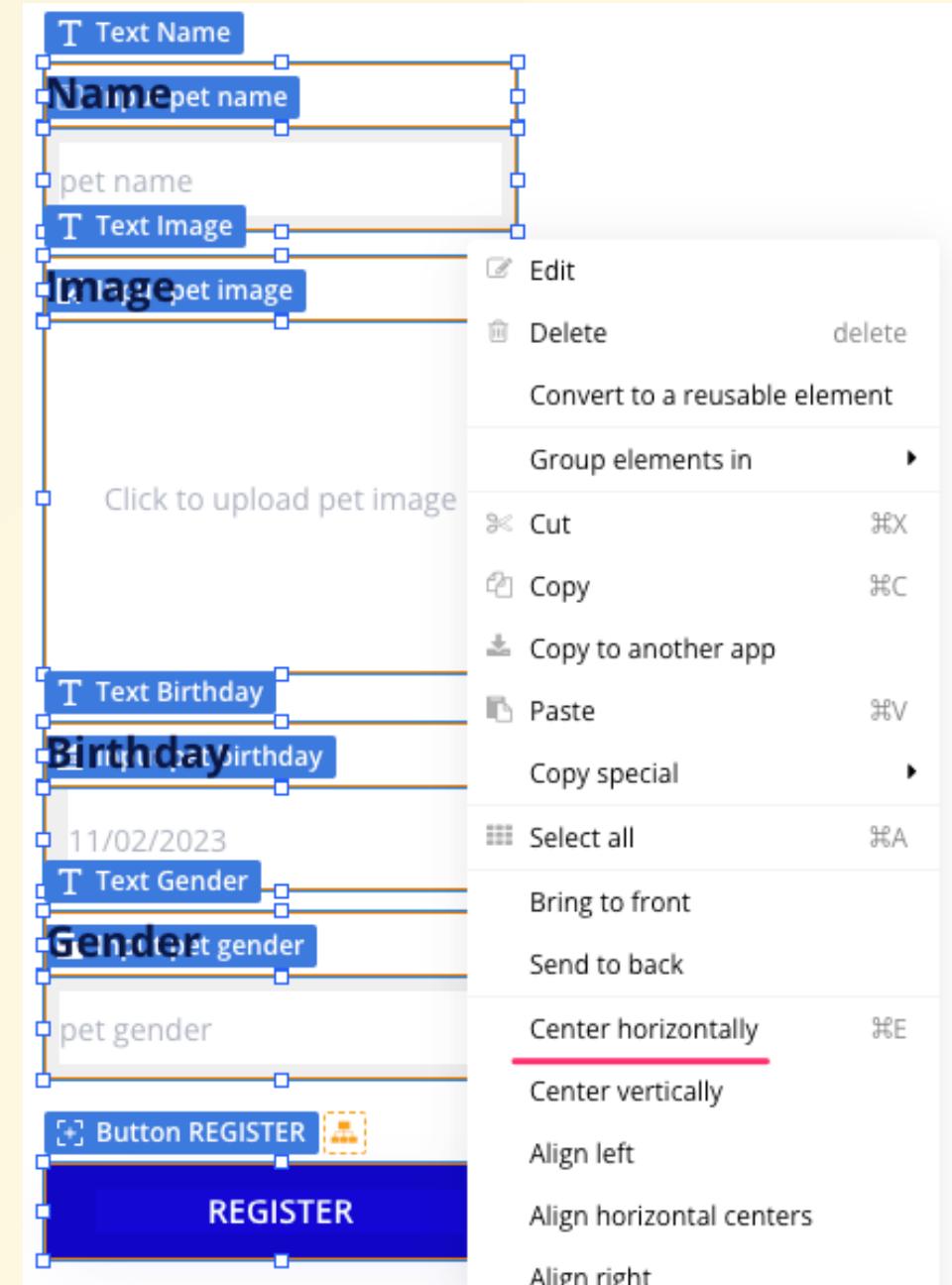
- Then you can set the width (W) and height uniformly for the selected elements, so enter 220 for W and you are done.
- Click on "Apply changes to elements".



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.



- Once you have done this, click **Preview** in the upper right corner of the screen to preview 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

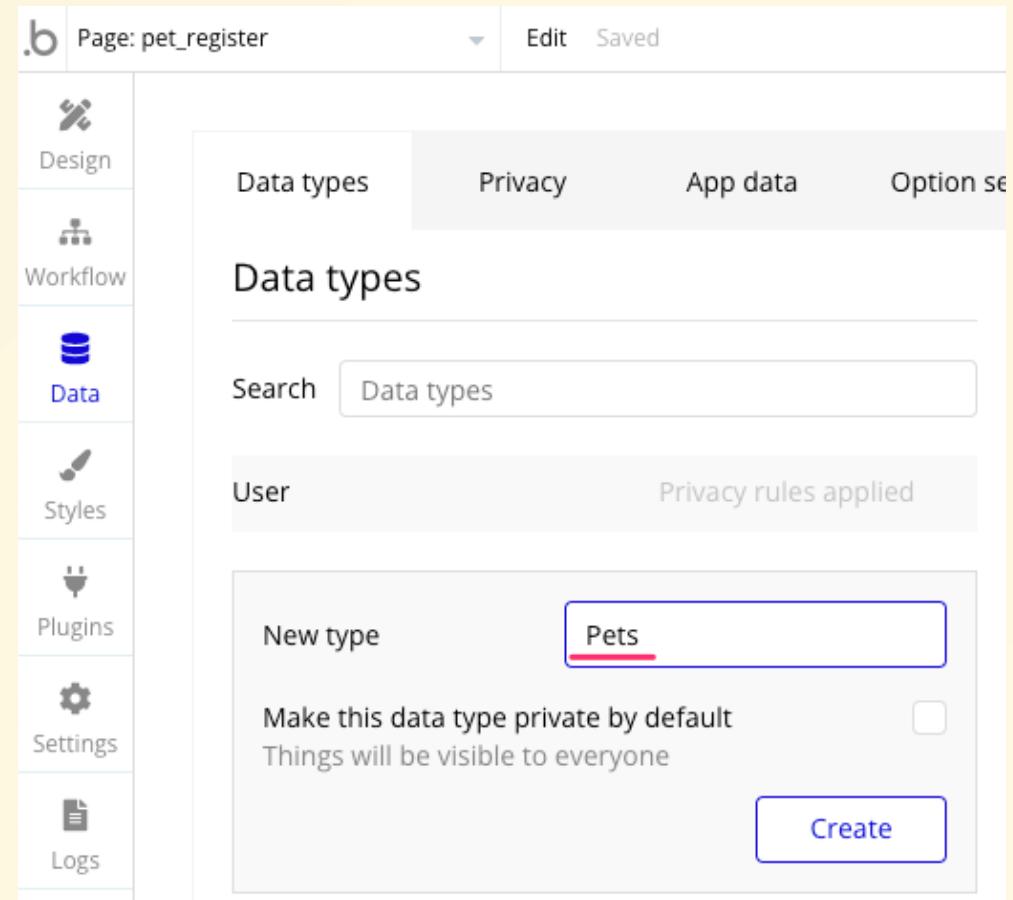
REGISTER

Connecting to a database

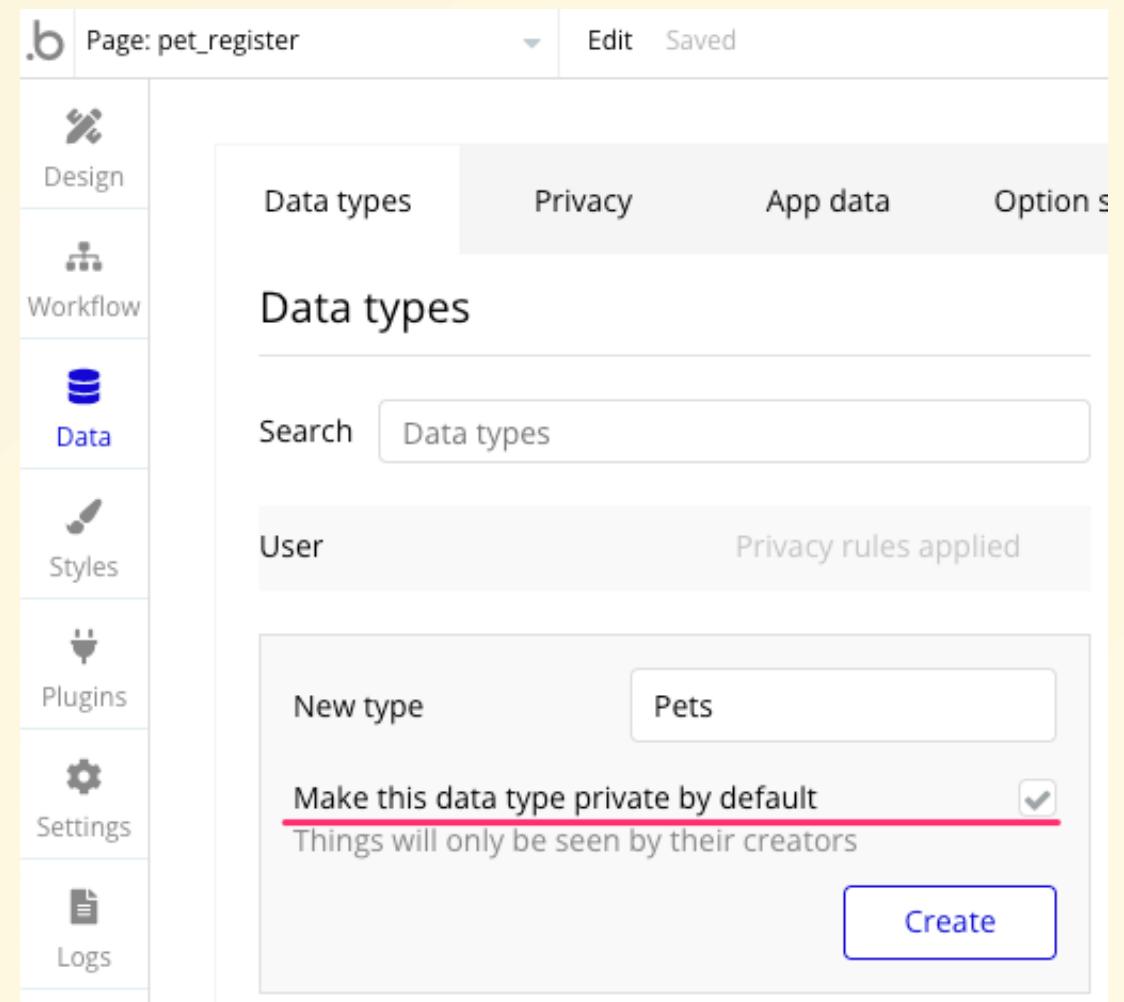
- Now that the screen layout is ready, it's time to connect to the database!

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

- Select Data from the tabs on the left menu
- Then select the Data types tab, 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 call each element of the database
- Type: Defines the type (box) of the data.
 - Collection in Adalo
- Field: Element for expressing the data type
 - Property in Adalo

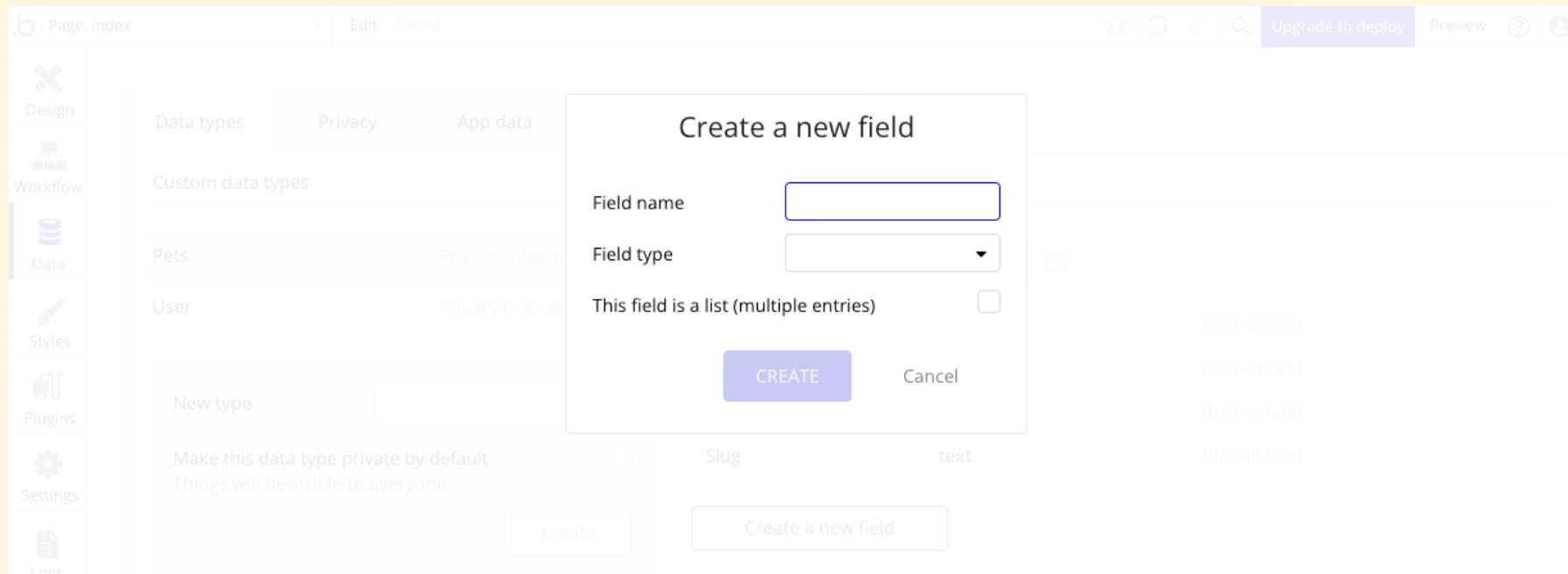
- Add elements (fields) to the created Pets type
- Select **Pets** from **Data types**.
- Click **Create a new field** under **Fields for type Pets** on the right side

The screenshot shows the Strapi admin interface for a 'pet_register' page. On the left, there's a sidebar with icons for Design, Workflow, Data, Styles, Plugins, Settings, and Logs. The main area has tabs for Data types, Privacy, App data, Option sets, and File manager, with 'Data types' selected. In the 'Data types' section, there's a search bar and a table with rows for 'Pets' and 'User'. A 'New type' input field is also present. On the right, under 'Fields for type Pets', there's a table with columns for Type name (set to 'Pets') and Field name. The table includes fields like 'Creator' (User), 'Modified Date' (date), 'Created Date' (date), and 'Slug' (text). A red box highlights the 'Create a new field' button at the bottom of this section.

| Type name | Field name |
|-----------|---------------|
| Pets | Creator |
| | User |
| | Modified Date |
| | Created Date |
| | Slug |

Create a new field

- 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 can specify a relationship between types from the **Relationship** field.
- In Bubble, the registered types are also displayed in the **Field type** field, so you can select the type you want to associate with it, as in Adalo.

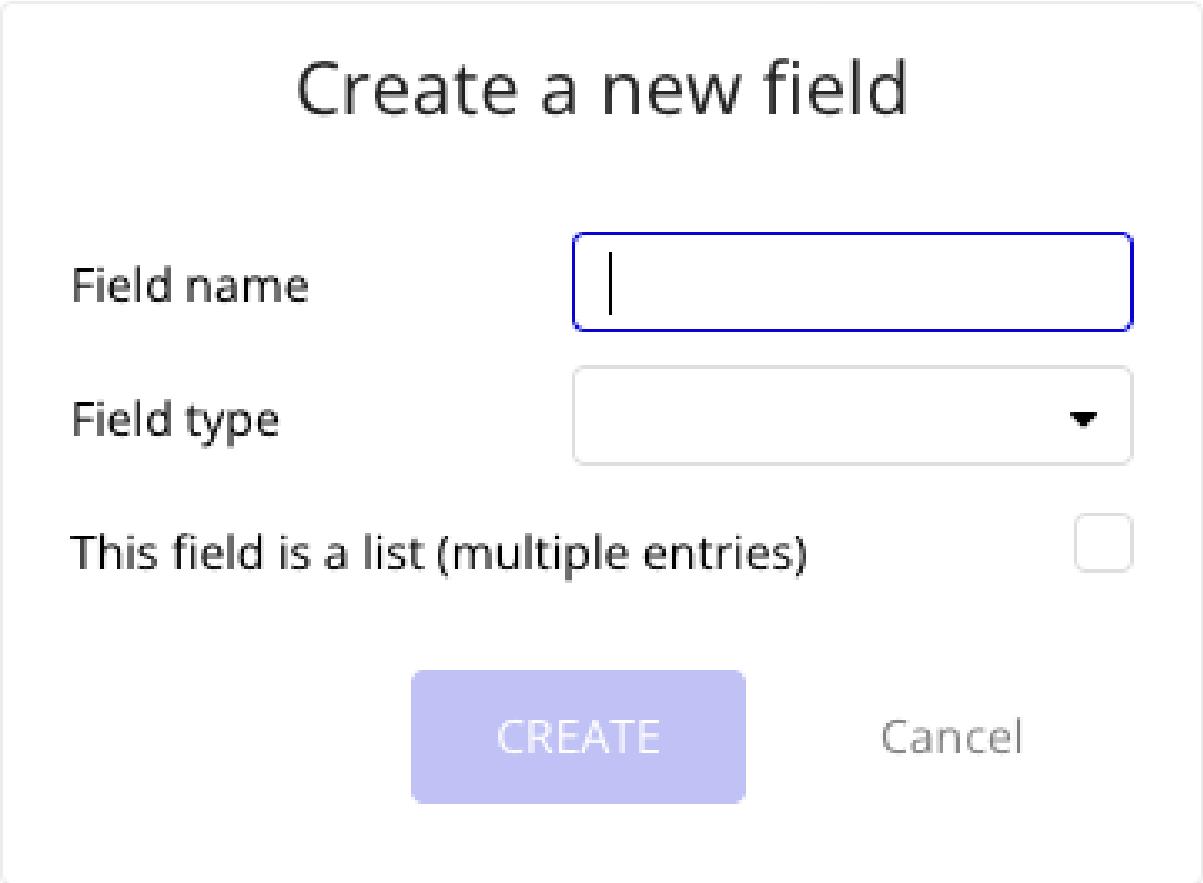
Create a new field

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

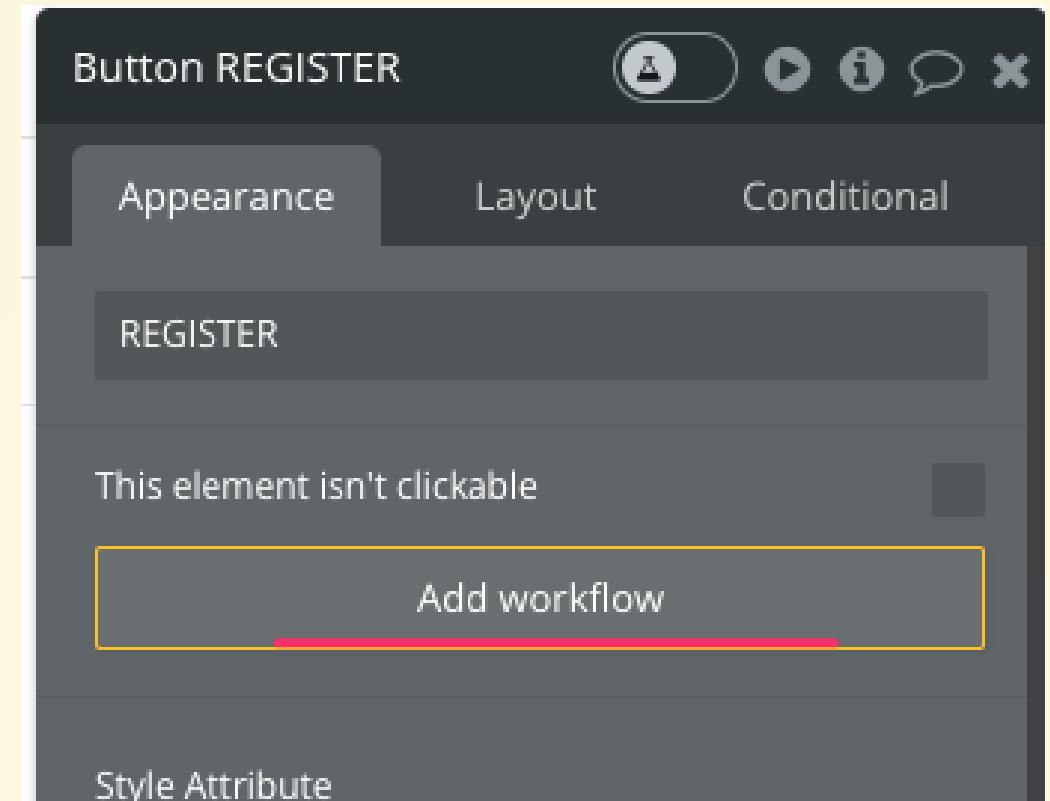
| | | |
|---------------|-------|---|
| Type name | Pets |  |
| Birthday | date | |
| Gender | text | |
| Image | image | |
| Name | text | |
| Creator | User | |
| Modified Date | date | |
| Created Date | date | |
| Slug | text | |

Create a new field

Make it possible to register pets when the registration button is clicked

- Now that the box for storing the pet's information is ready, it is time to add a movement to save the pet's 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.

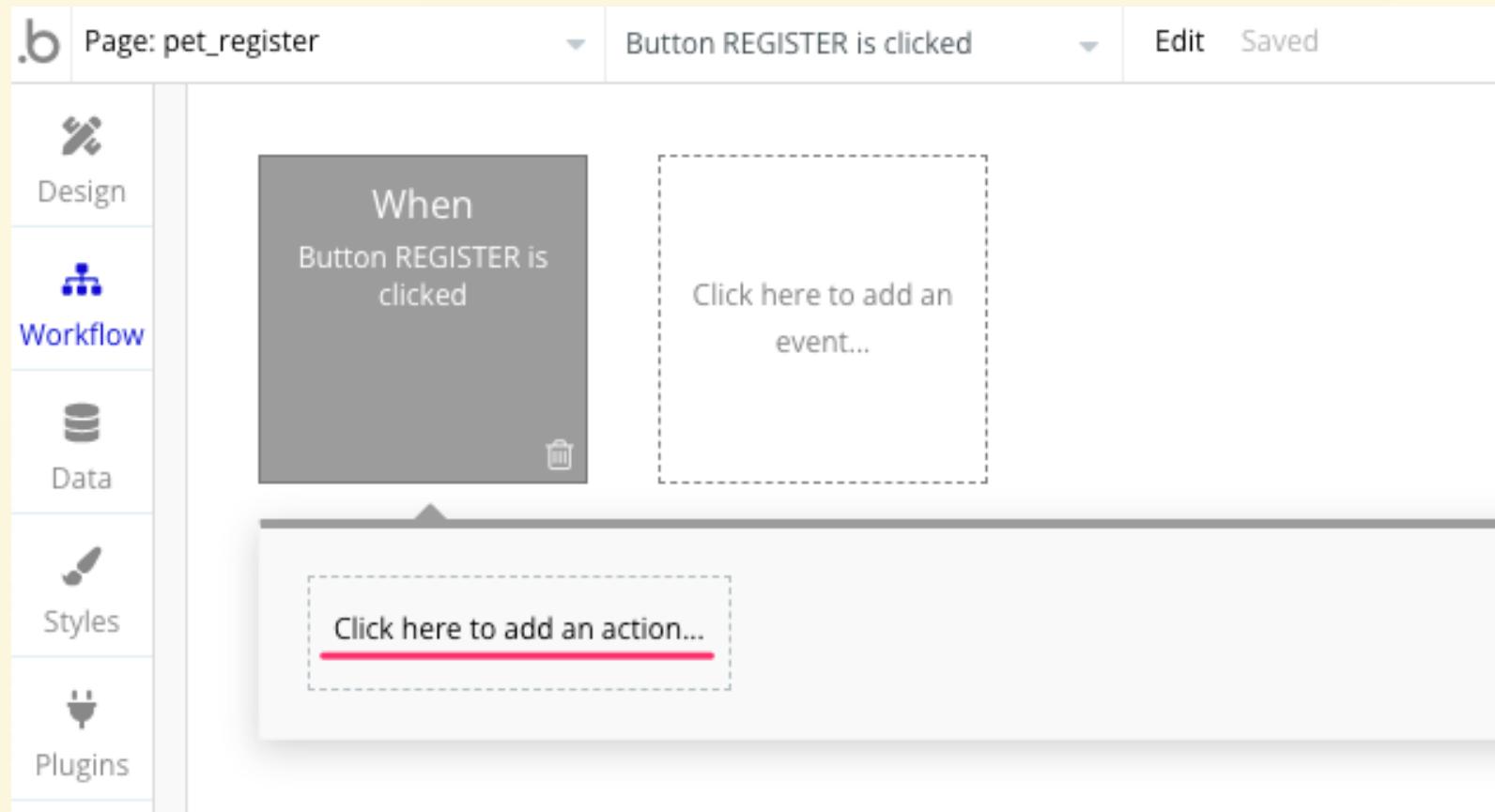
- Select the Design tab from the left menu
- Select the "pet_register" screen from the upper left panel
- Double-click the "REGISTER" button on the "pet_register" screen from the right panel
- Click **Add workflow** in the "Appearance" tab.



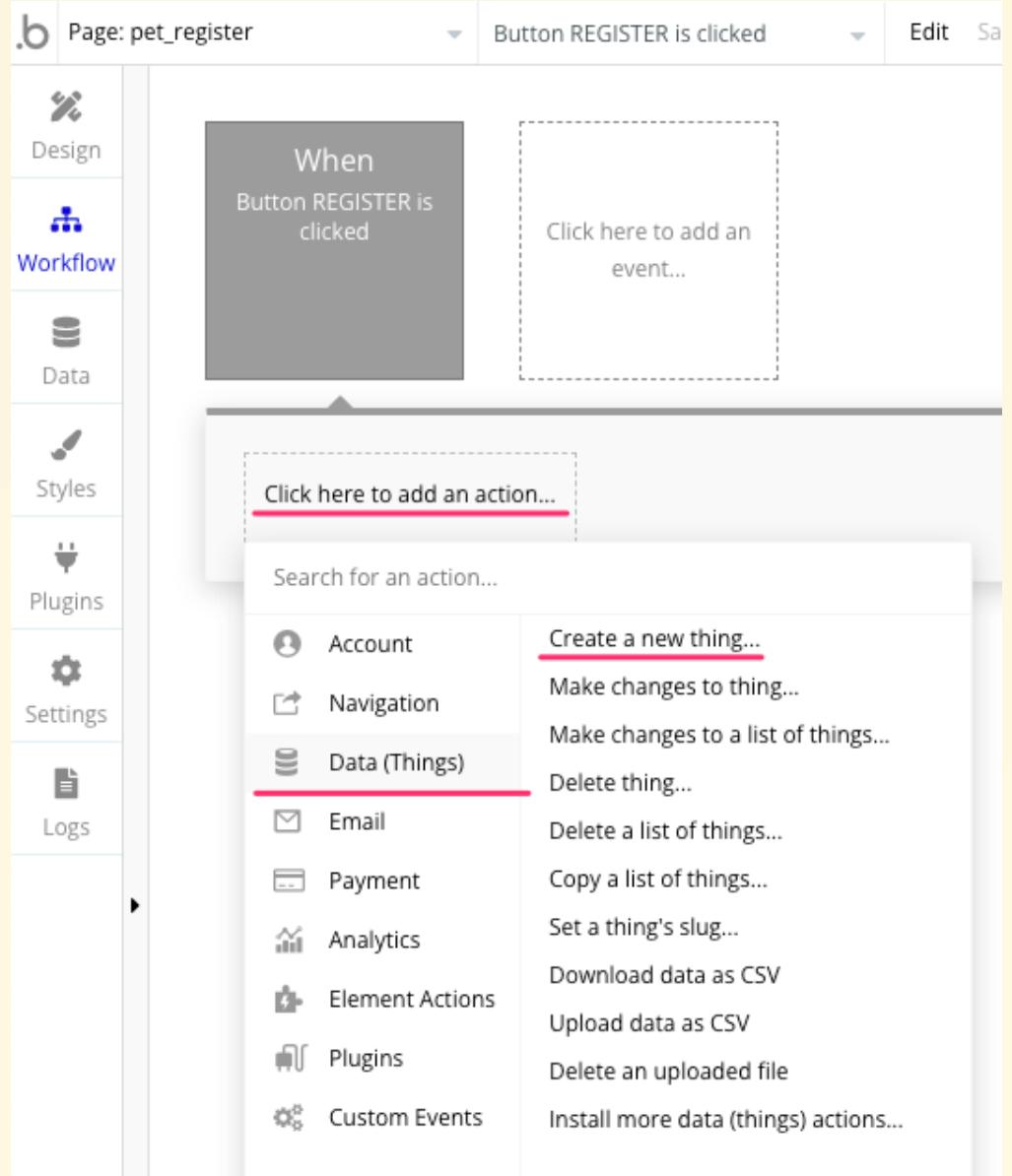
- You will then switch to the Workflow tab, where you will see **Button REGISTER is clicked** in the When section.
 - This is literally the workflow definition when the "REGISTER" button is clicked.

The screenshot shows a workflow editor interface. At the top, there's a header with a logo, the page title "Page: pet_register", a dropdown menu, and buttons for "Edit" and "Saved". On the left, a vertical sidebar has five tabs: "Design" (selected), "Workflow" (highlighted in blue), "Data", "Styles", and "Plugins". The main workspace displays a workflow step. It starts with a "When" condition box containing "Button REGISTER is clicked". To the right is a dashed box with the placeholder "Click here to add an event...". Below this is a horizontal bar with a downward-pointing arrow. Underneath the bar is another dashed box with the placeholder "Click here to add an action...". A red horizontal line highlights the "Click here to add an action..." box.

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



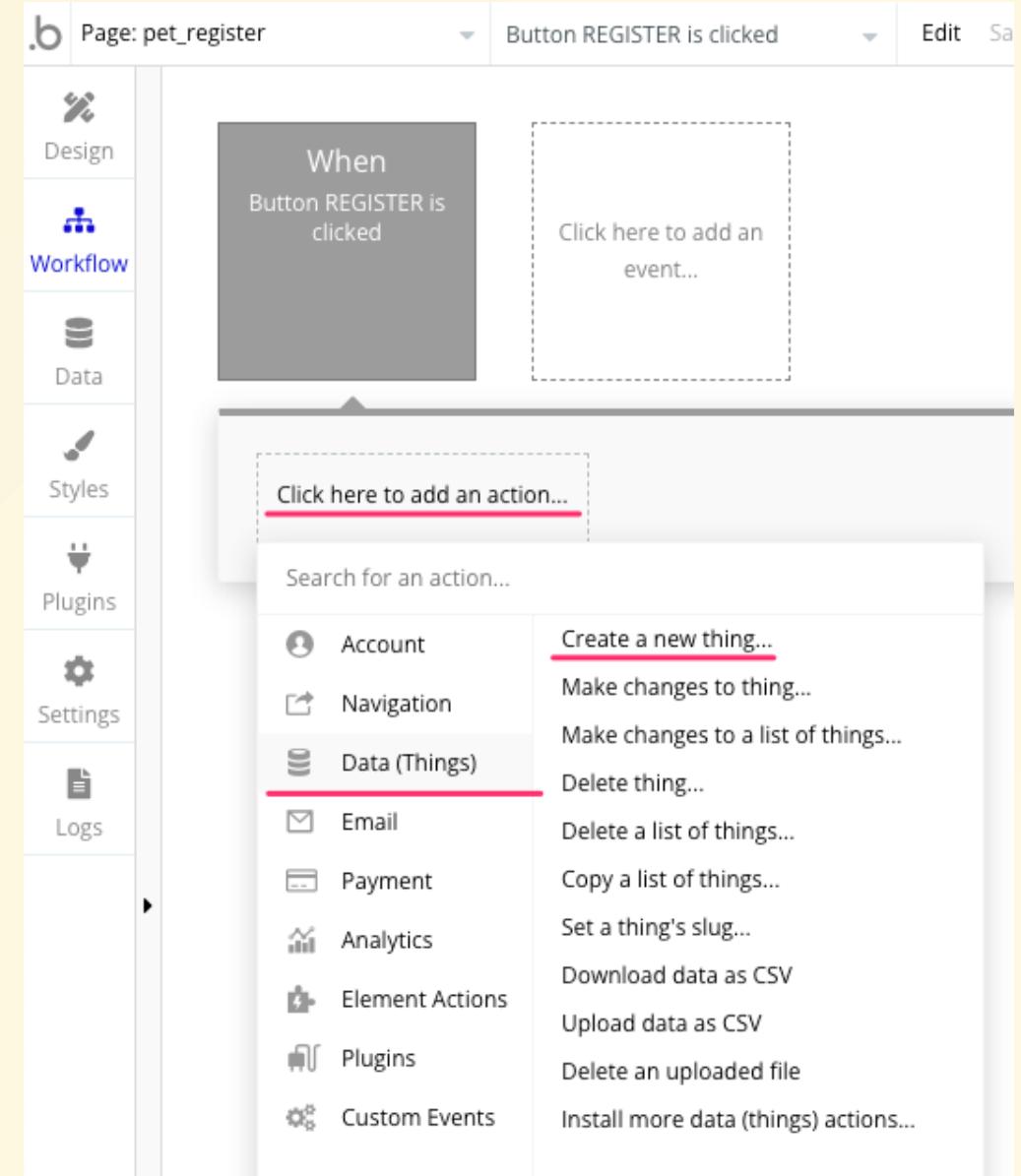
- Click here to add an action...
to display a popup window where you can specify various actions.
- Select **Data(Things)** for database operations (actions) like this one.



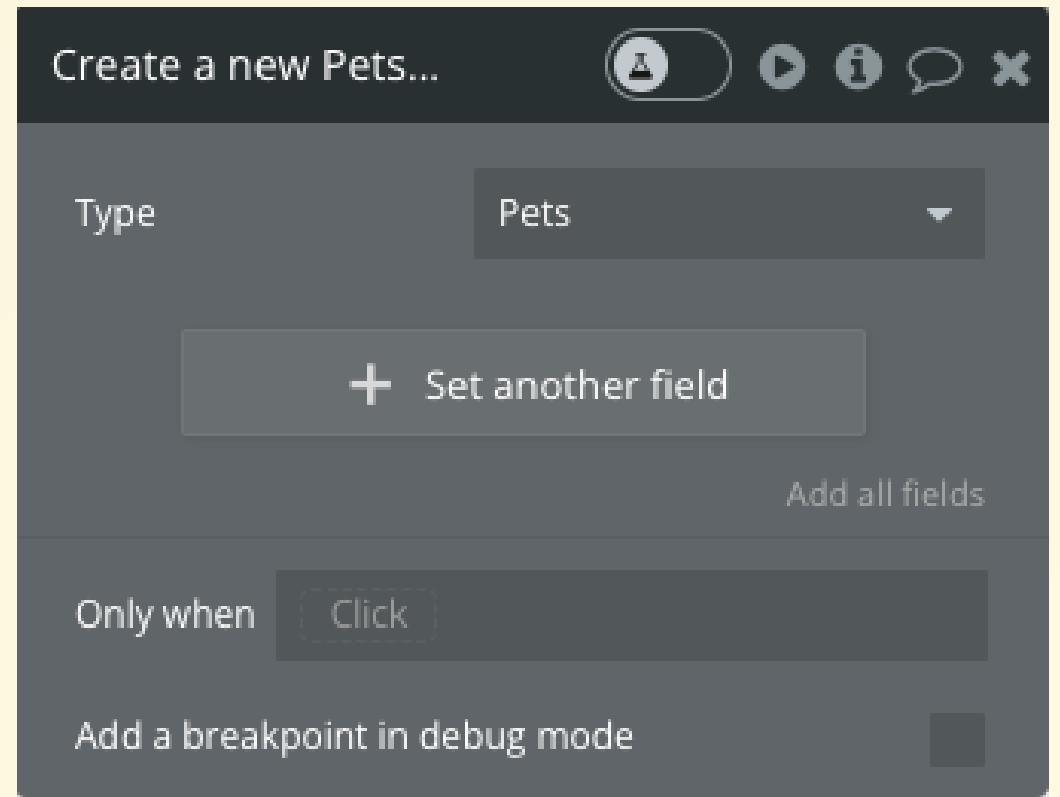
- Then you will see more sub-elements, select

Create a new thing...
Select

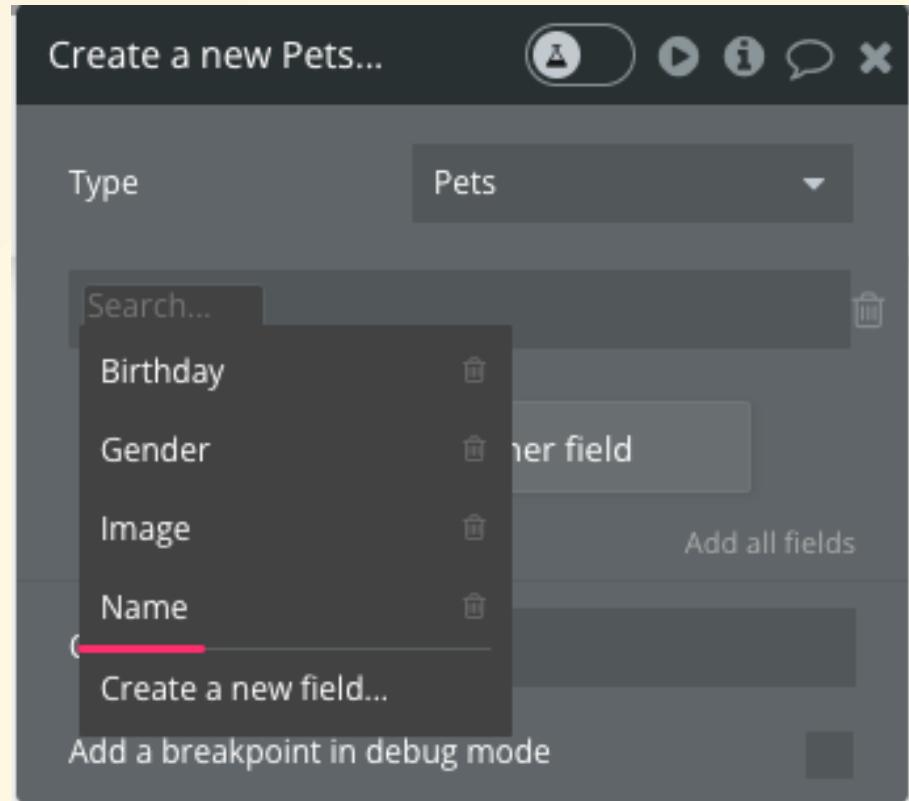
Create a new thing...` from
the list.



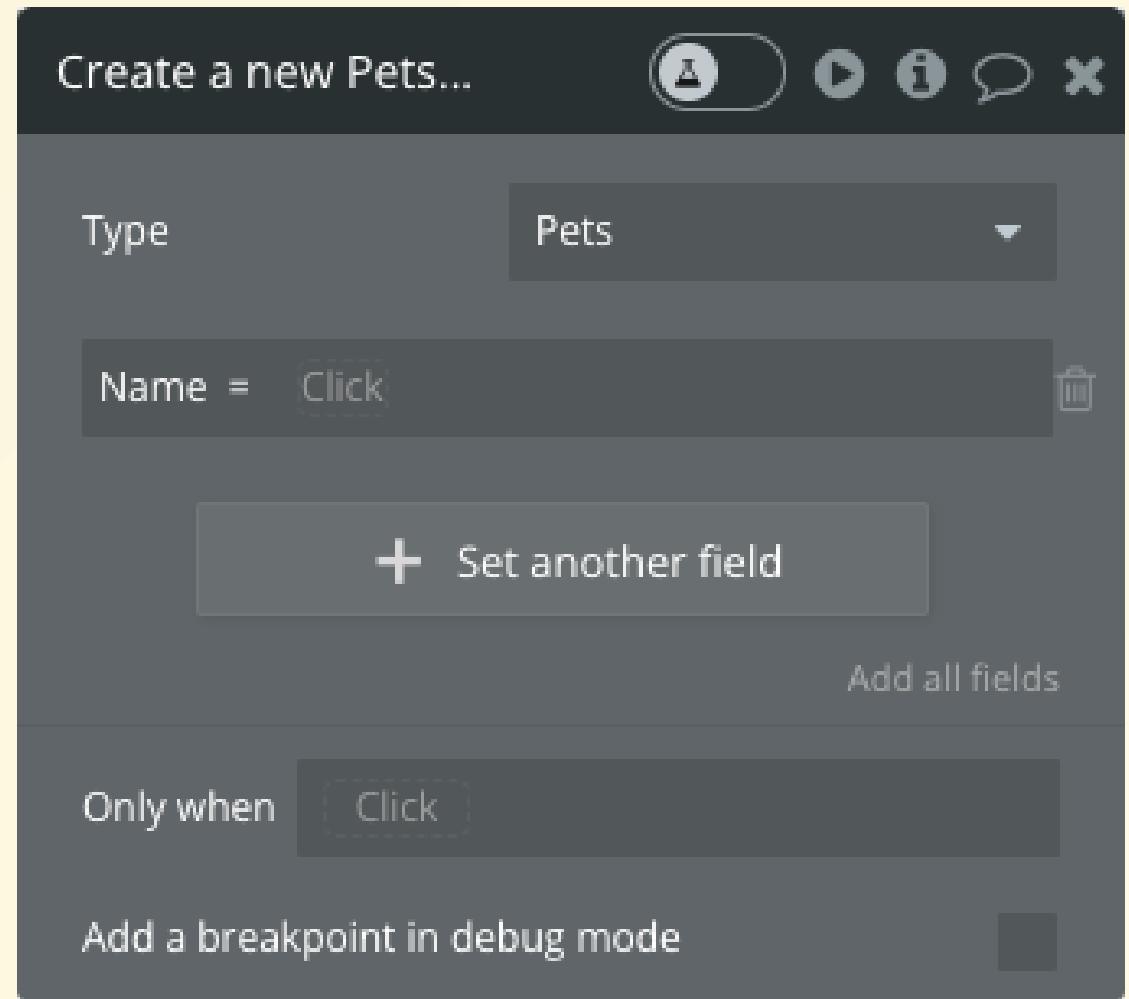
- The **Create a new thing...** dialog box will appear, and you need to specify the type of registration this time.
 - In this case, it is Pets.
- Then, a button **Set another field** will appear, and you will set the field to the Pets "field" as shown in the title of the dialog.



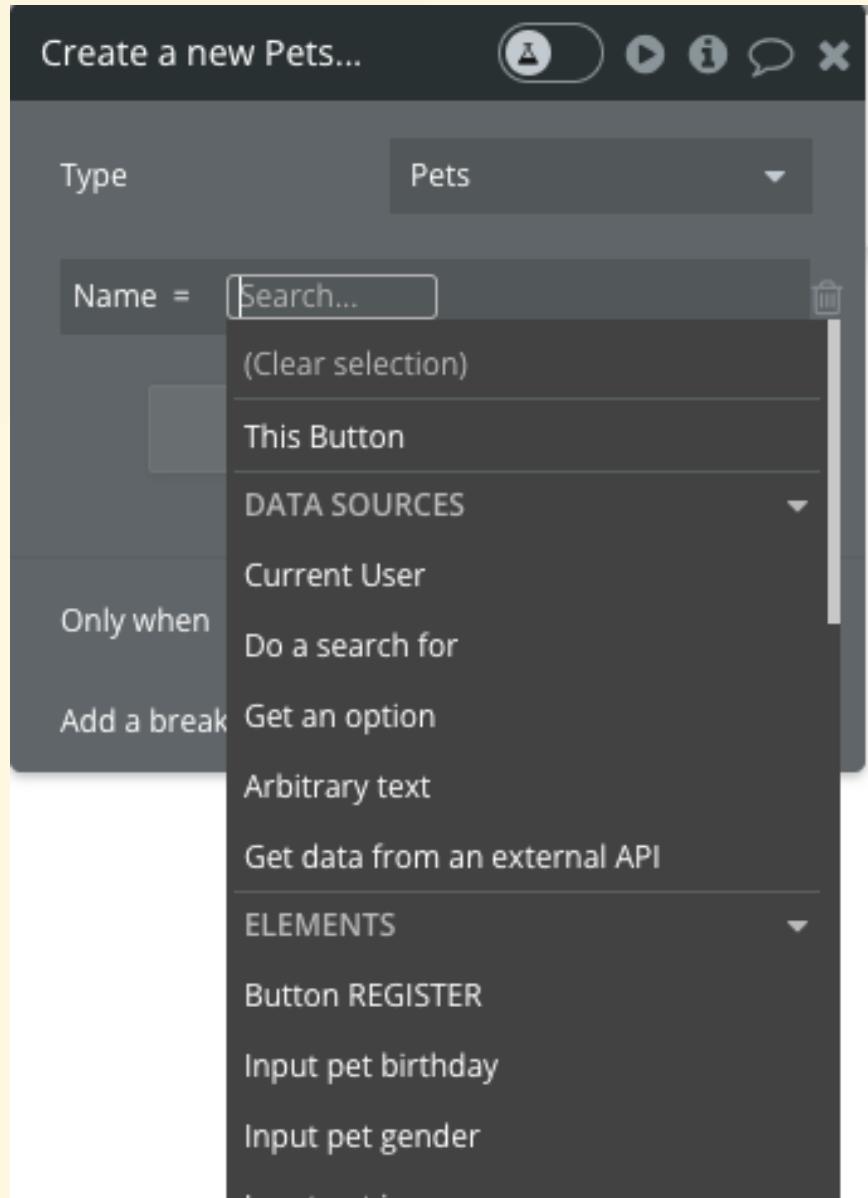
- Here we specify the "field" name of the item we want to store in Pets on the left-hand side, and the value we actually want to store in that "field" on the right-hand side
 - First, select **Name** for the left-hand side



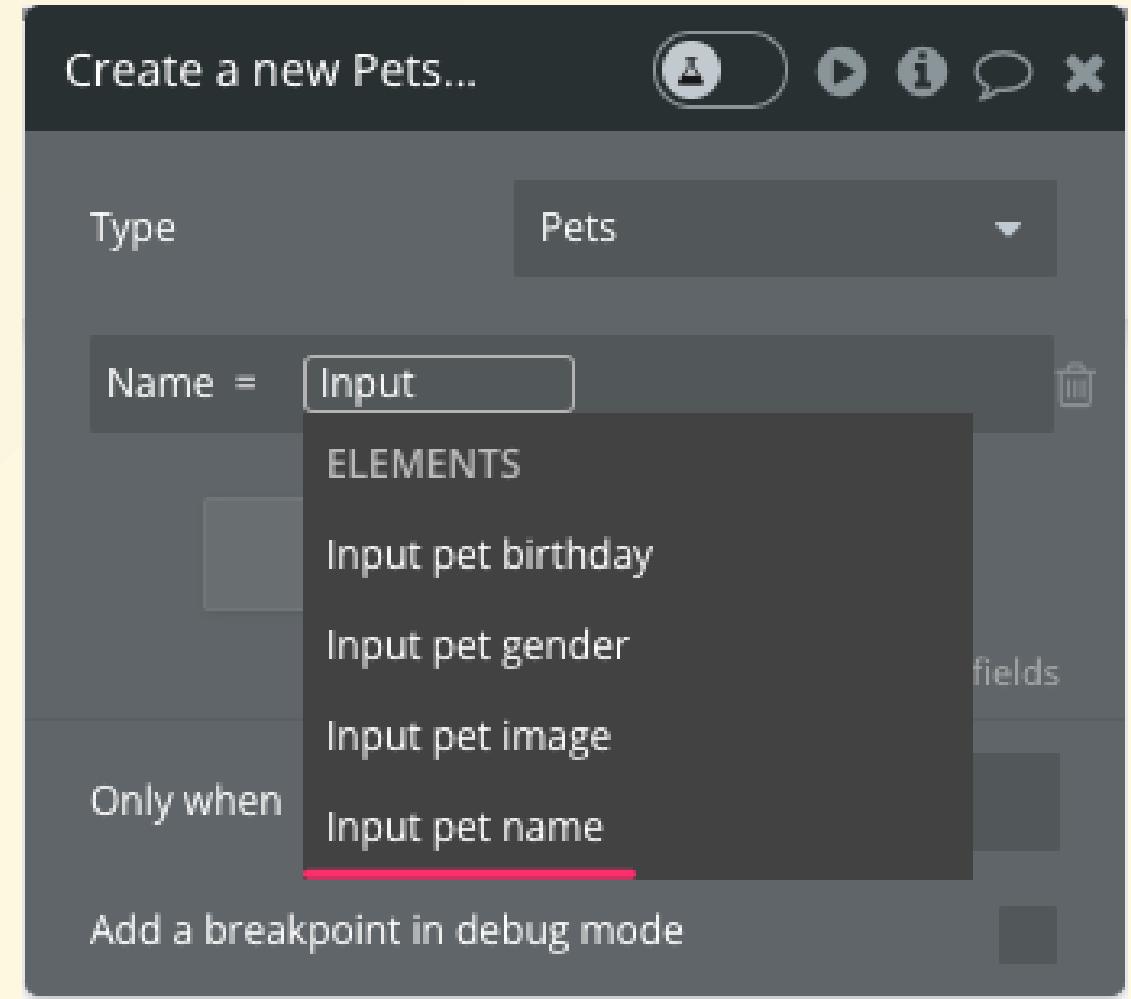
- If you specify the left side, you will see **=** and now **Click** on the right side.



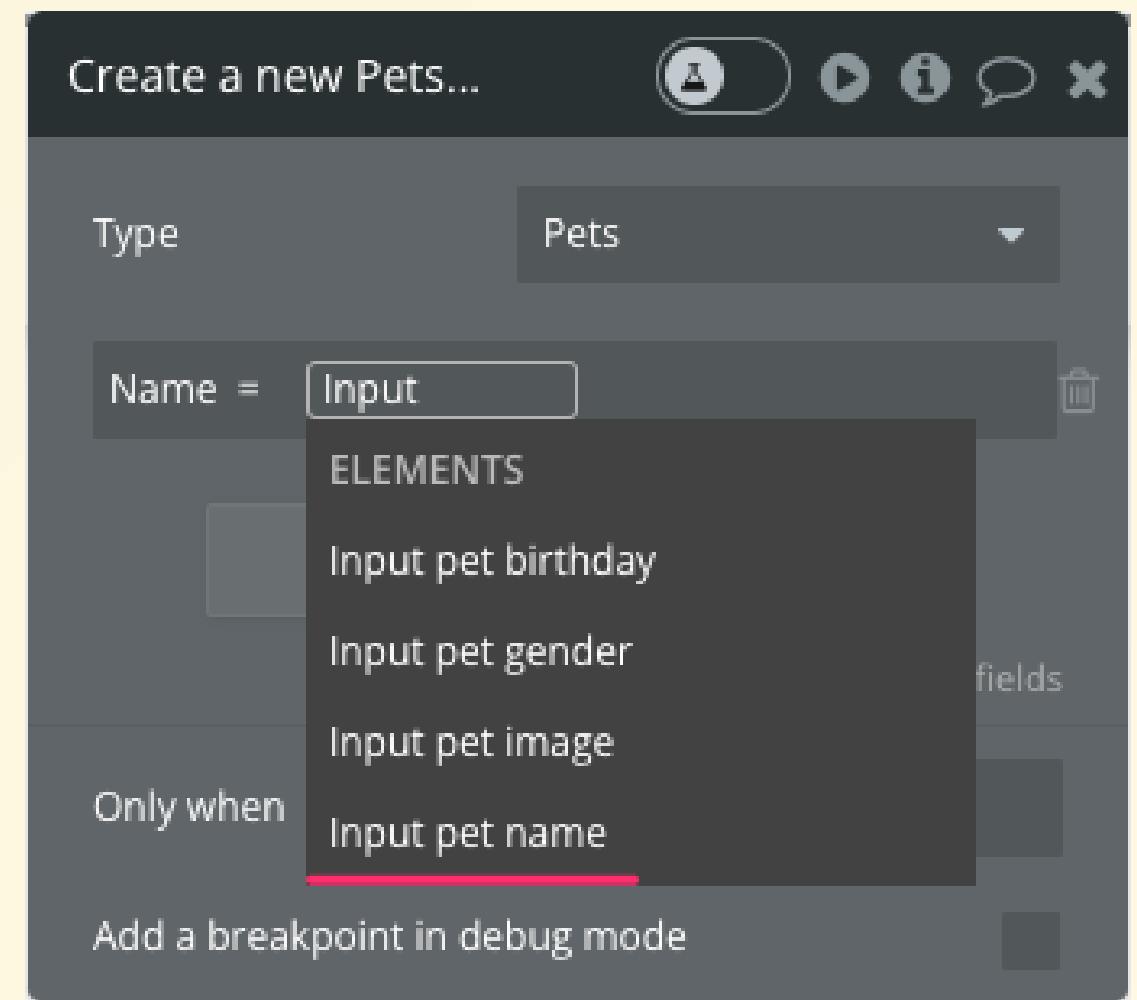
- Click here and you'll see a variety of options.
- I won't bore you with the details here, but you can specify what you want to do with the fields selected on the left side.



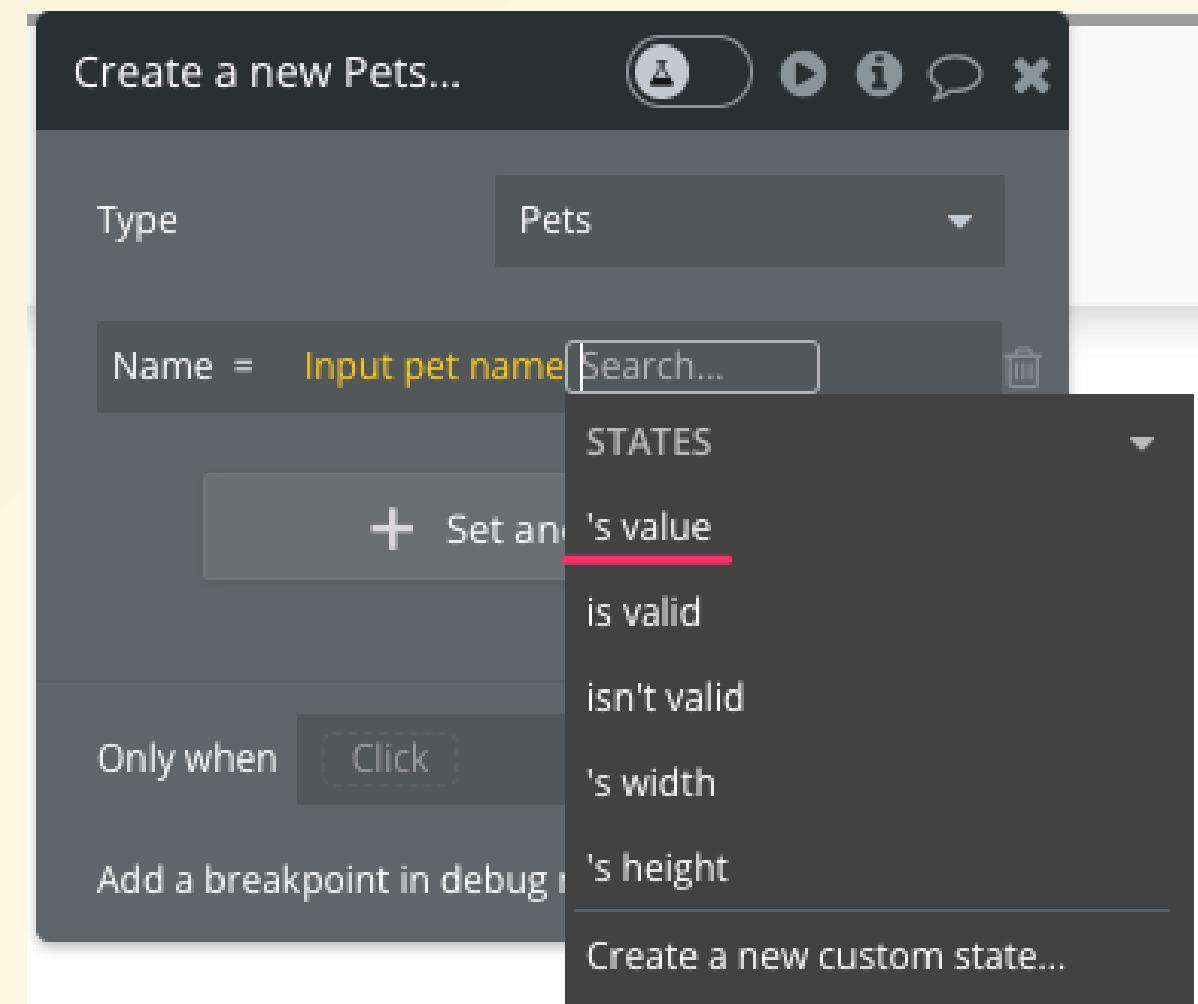
- In this case, select the **Input pet name** field, which will be the name of the pet entered on the screen.
- If there are too many choices, you can also narrow down the field by typing the name of the field.
 - In the capture, we have narrowed down the field by typing "Input".



- 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... (A) (P) (I) (C) (X)

Type Pets ▾

Name = Input pet name's value trash

Image = Input pet image's value trash

Birthday = Input pet birthday's value trash

Gender = Input pet gender's value trash

+ Set another field

Only when Click

Add a breakpoint in debug mode checkbox

Only when Click

Add a breakpoint in debug mode checkbox

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)
- Let's 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.

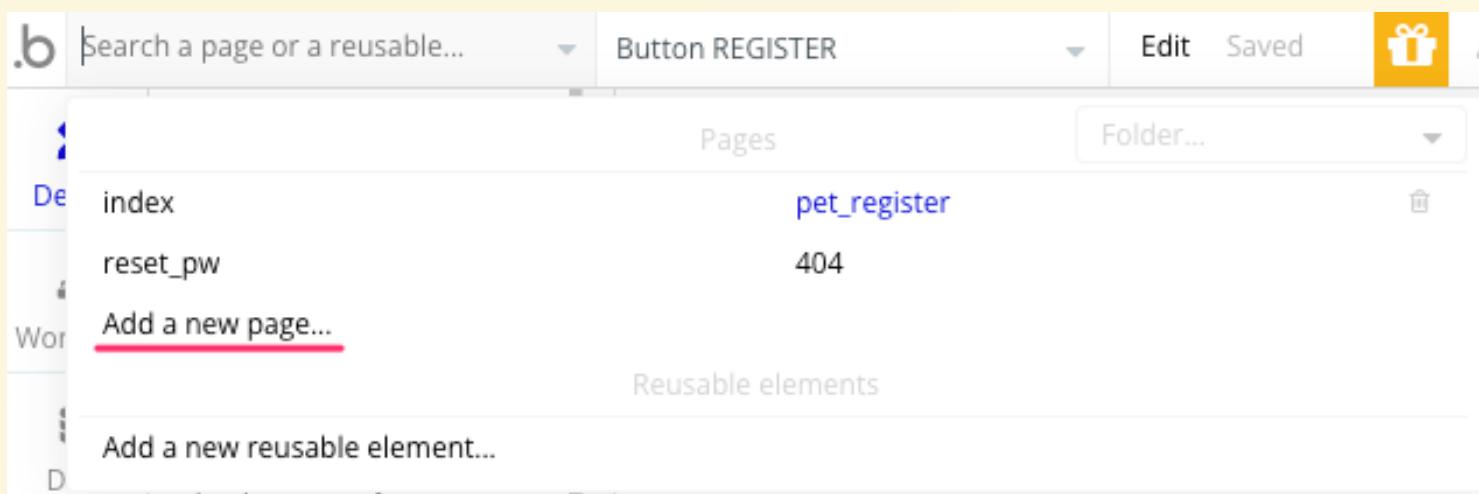
The screenshot shows the Bubble app interface with the following details:

- Header:** Page: pet_register, Edit, Saved, 0 issues, Upgrade to deploy, Preview.
- Left sidebar:** Design, Workflow, Data (selected), Styles, Plugins, Settings.
- Top navigation:** Data types, Privacy, **App data** (selected), Option sets, File manager.
- Section titles:** App data, Application data - All Pets - Development version, Copy and restore database, Switch to live database.
- Buttons:** New view, Primary fields, Search, Data entries (displaying 1), New entry, Delete (0), Upload, Modify, Export, Bulk.
- Search bar:** Search, Views or data types.
- Data table:** Shows 1 entry (displaying 1). Columns: Birthday, Gender, Image, Name, Created Date, Modified Date, Slug. The first row contains:
 - Birthday: Jul 18, 2022 12:00 am
 - Gender: Female
 - Image: 🐕
 - Name: うさこ
 - Created Date: Nov 2, 2023 1:19 am
 - Modified Date: Nov 2, 2023 1:19 am
 - Slug: (empty)A red box highlights the entire row for "All Pets".
- Panel below table:** All Pets, All Users.

- Now we have reached the point where we can save what we entered 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 pop-up window.
- Click **Add a new page...** in the pop-up window. ` in the pop-up window.



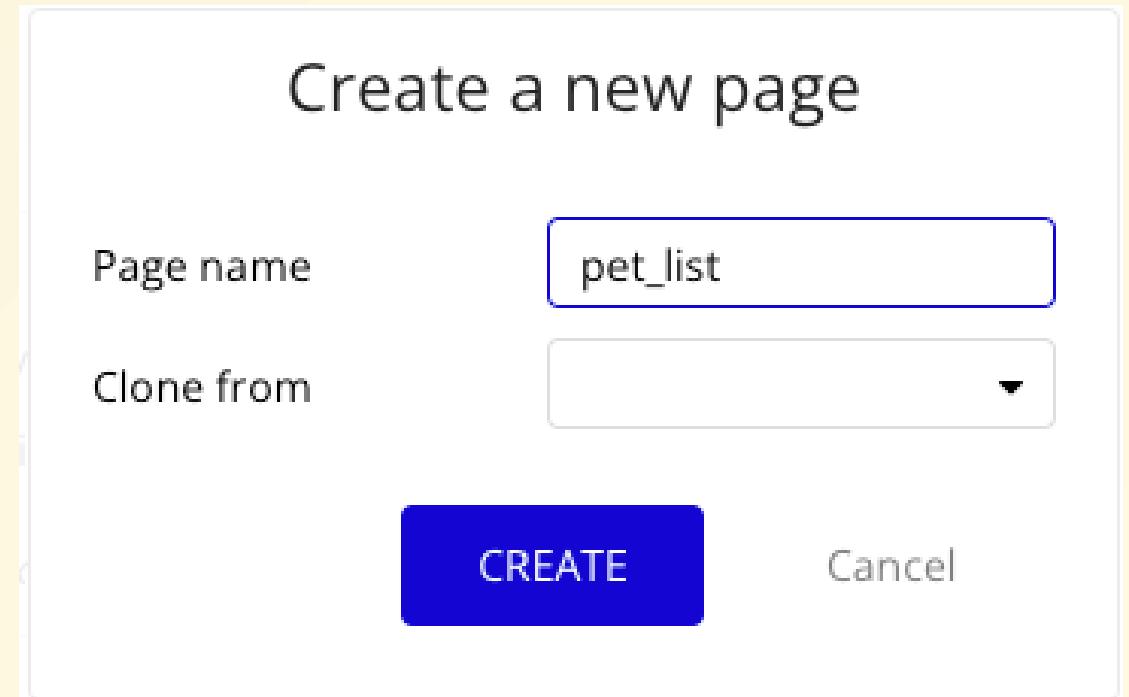
- Add a new page... to pop up a new page
- Page name should be pet_list.
- Clone form is left blank this time
- Click the "CREATE" button.

Create a new page

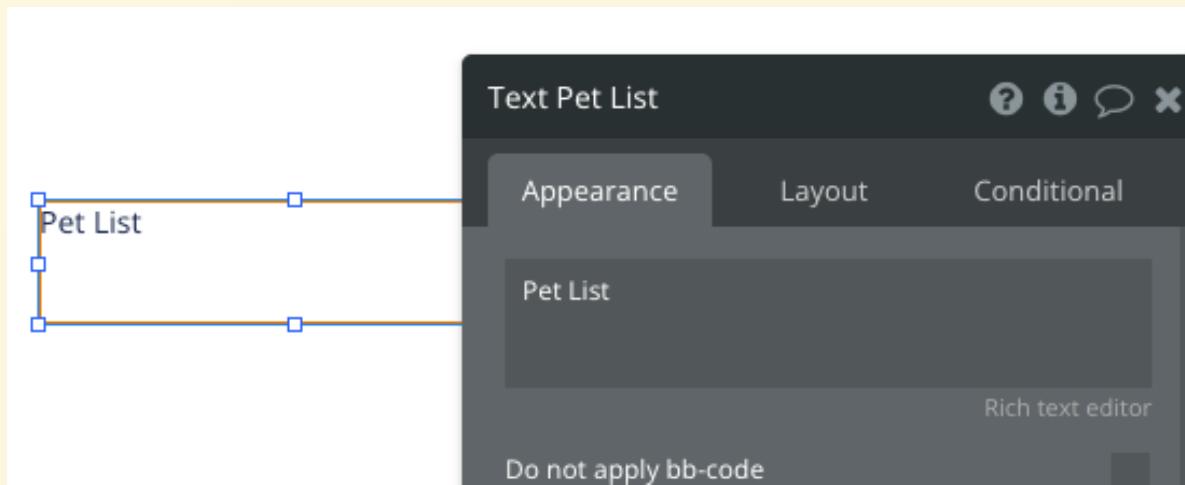
Page name

Clone from

CREATE Cancel

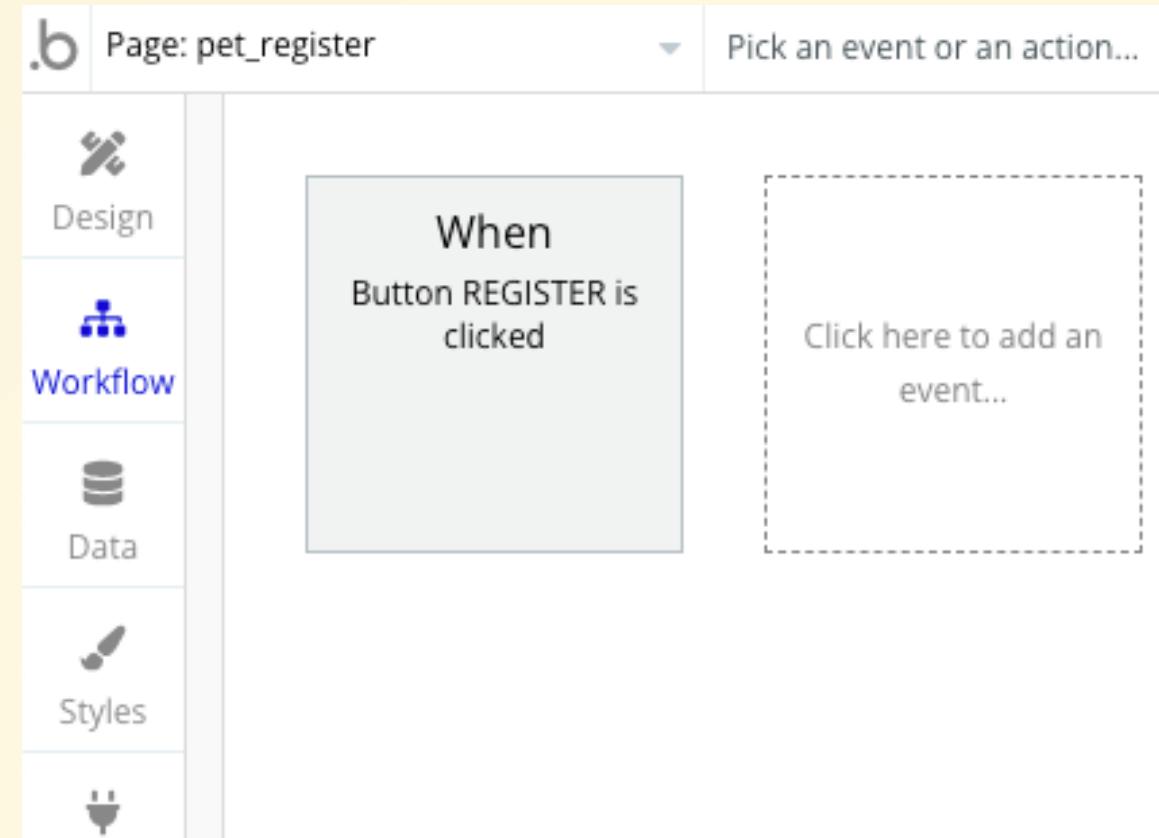


- Since it is difficult to tell which screen is which if the screen is blank, let's place only the text label from the Visual elements and call it "Pets List".
- Now only the frame of the Pets List screen has been created.

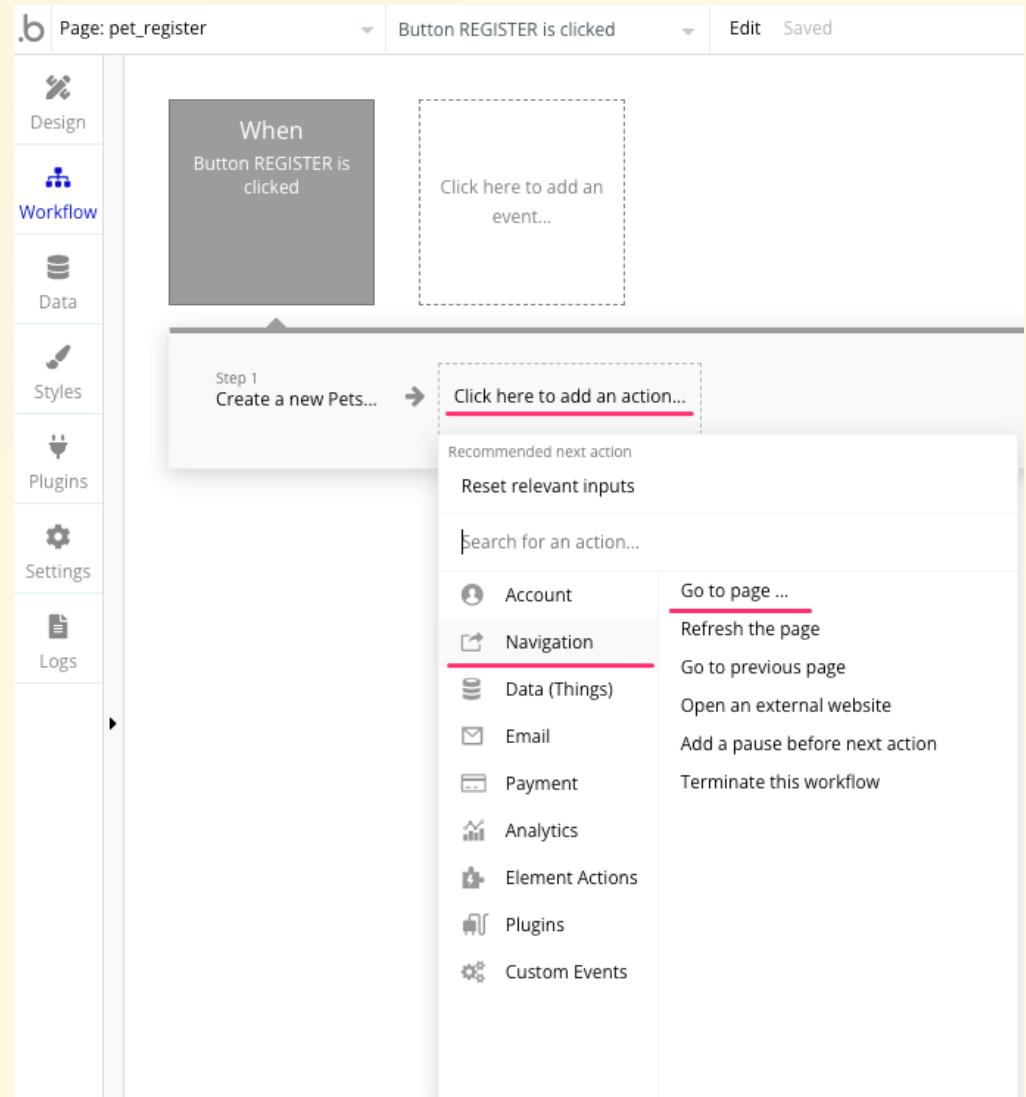


Let's make a screen transition to the Pet List screen.

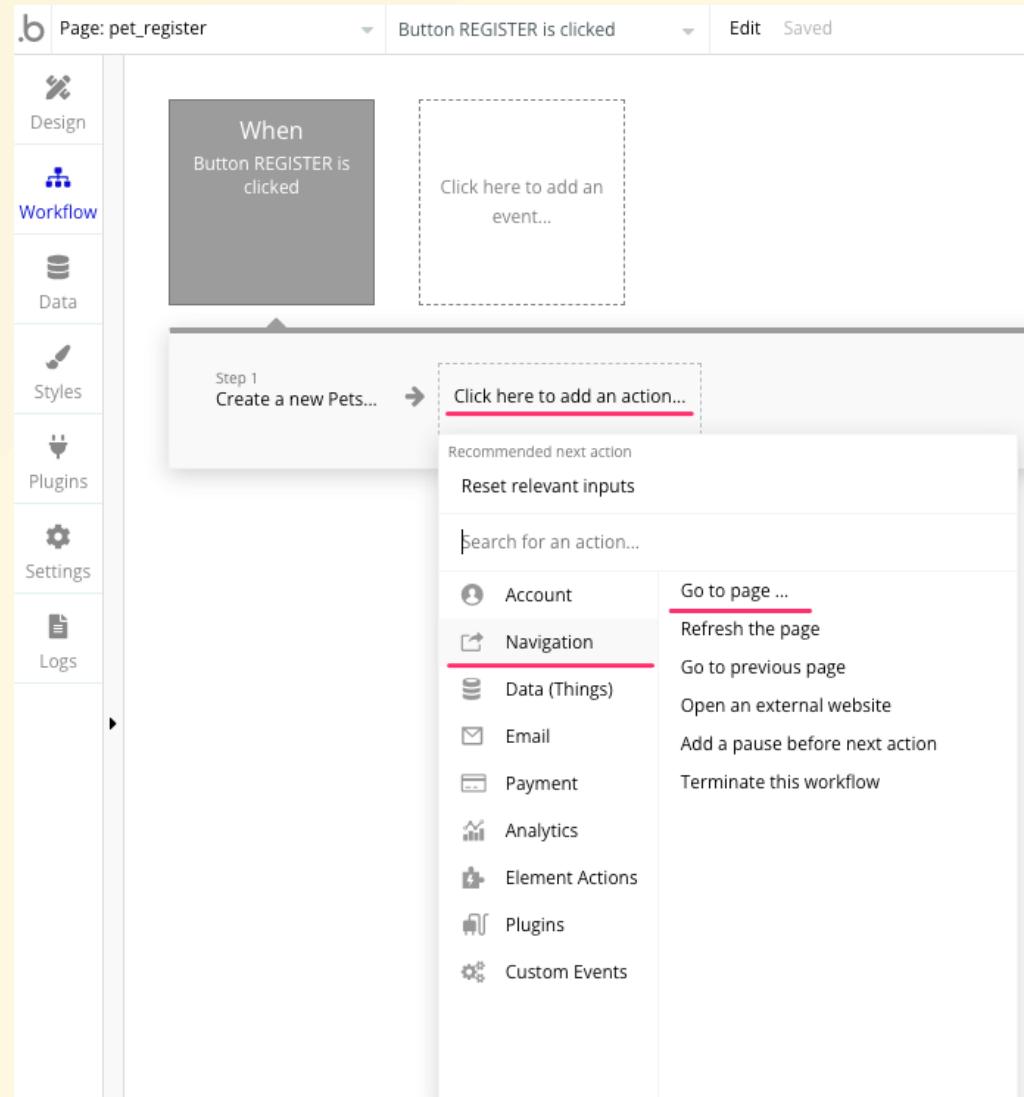
- Select pet_register page from the top left
- Click on the Workflow tab from the left menu
- You should see the behavior when you click the "REGISTER" button that you just set up.



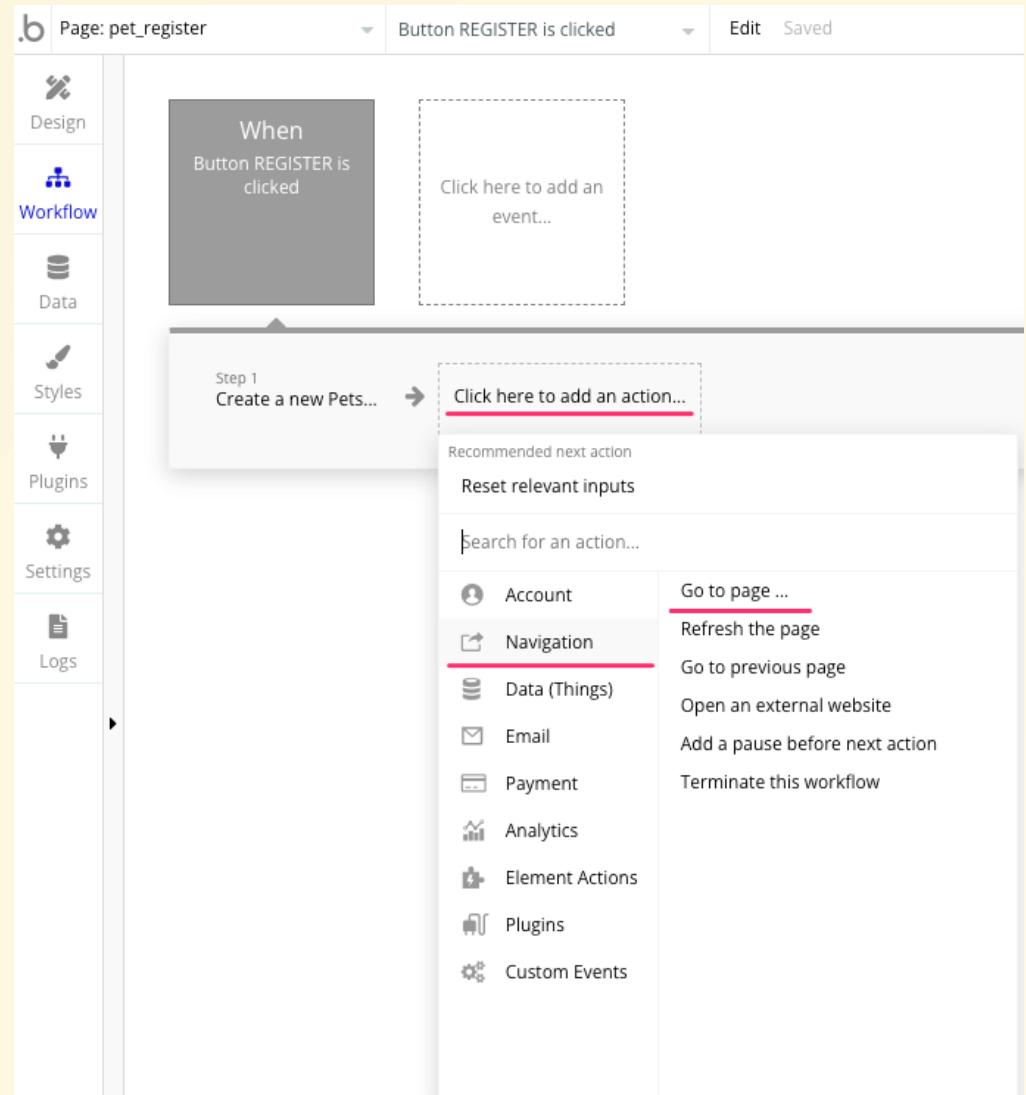
- Here, to the right of "Create a new Pets..." Click here to add an action...
- Click here to add an action...



- Now select **Navigation** and click **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.



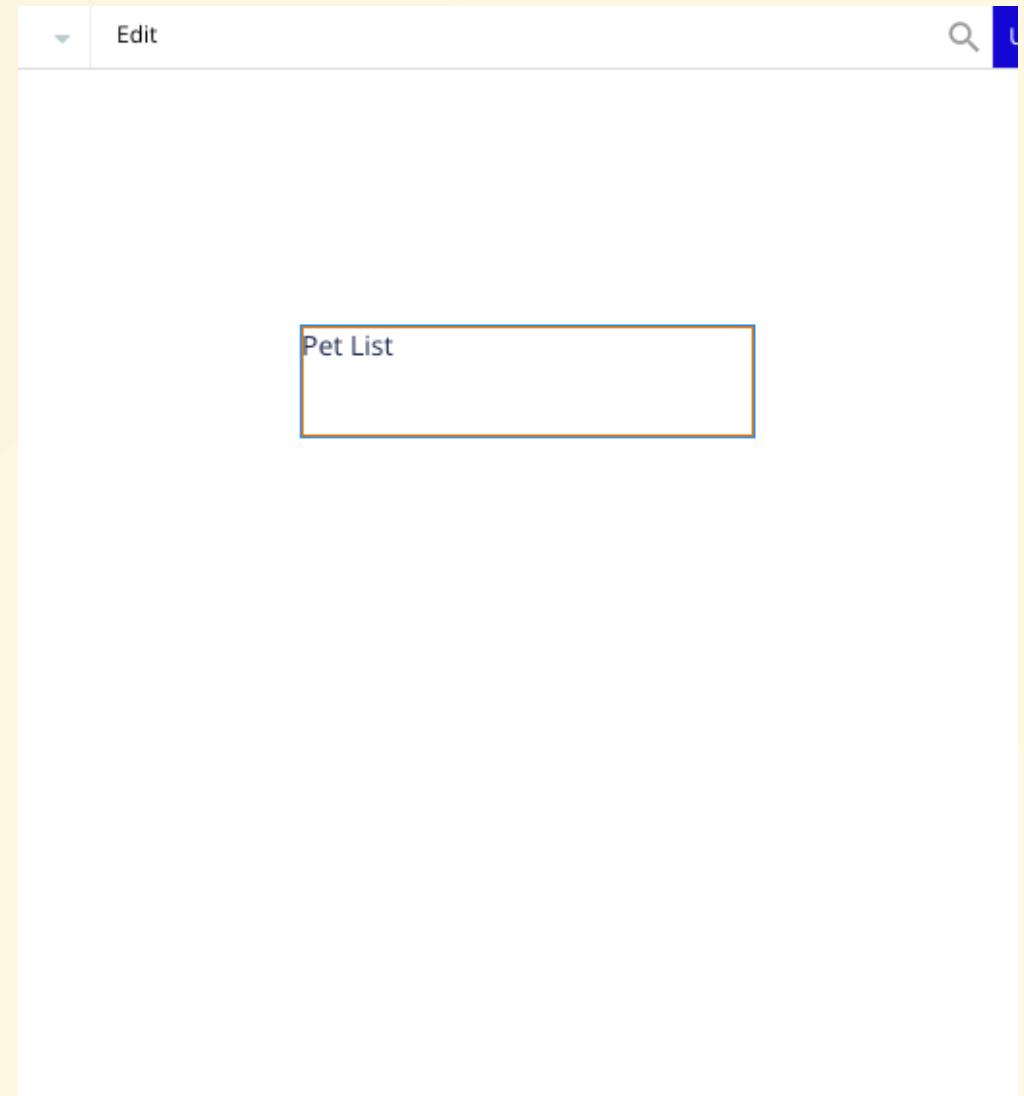
- Now let's preview it.
- Enter the pet information as before and click the "REGISTER" button.
- Then, you will see a transition to the Pet List screen.
- Now you have mastered the screen transition! Now you have mastered the screen transition! 

Listing Pets

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

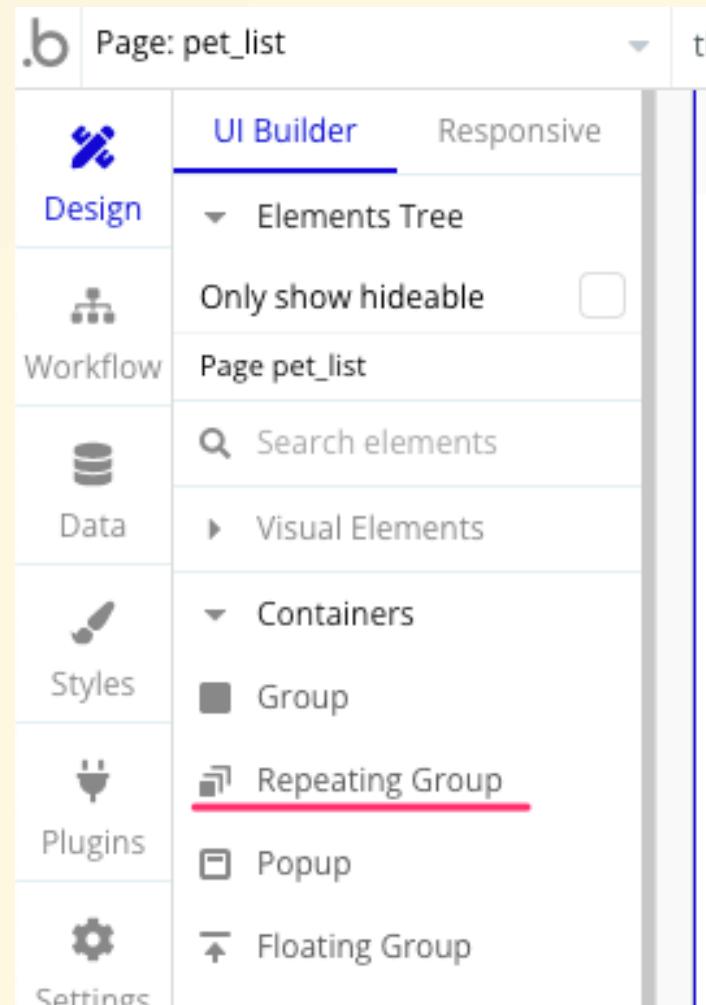


- Select `pet_list` from the list of pages in the upper left corner.
- Delete the text of the pet list you have just prepared.
 - Select the element and press Backspace to OK.

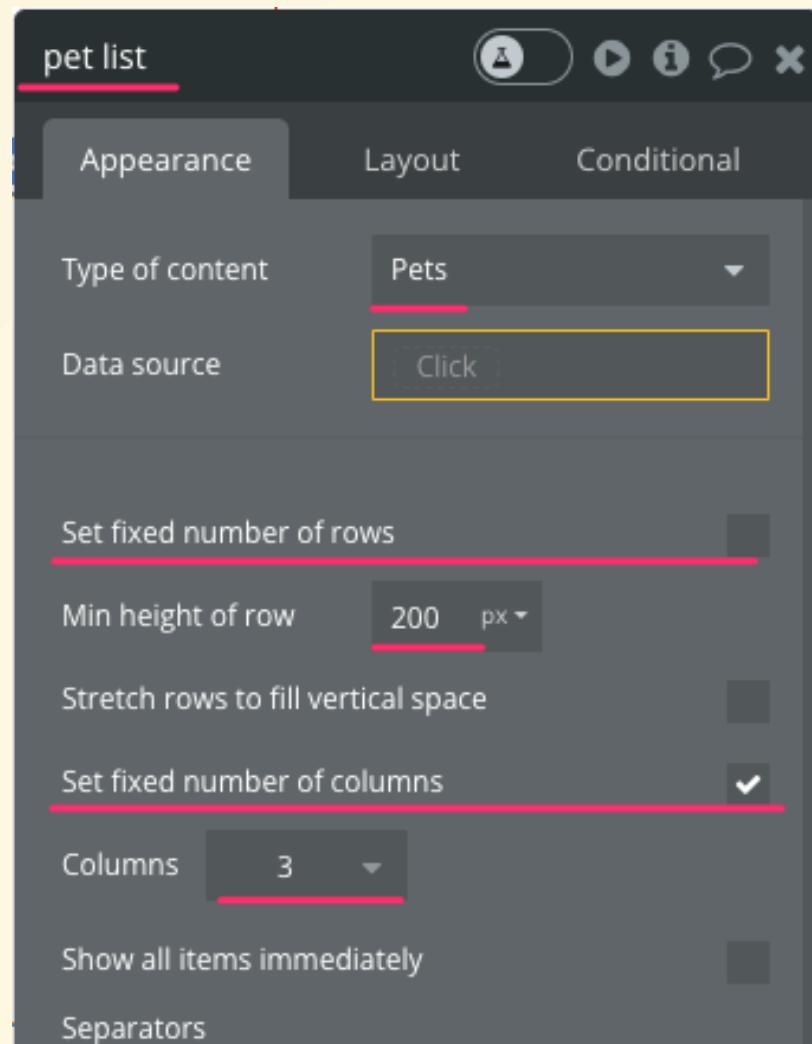


We will create a pet list screen.

- 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`.

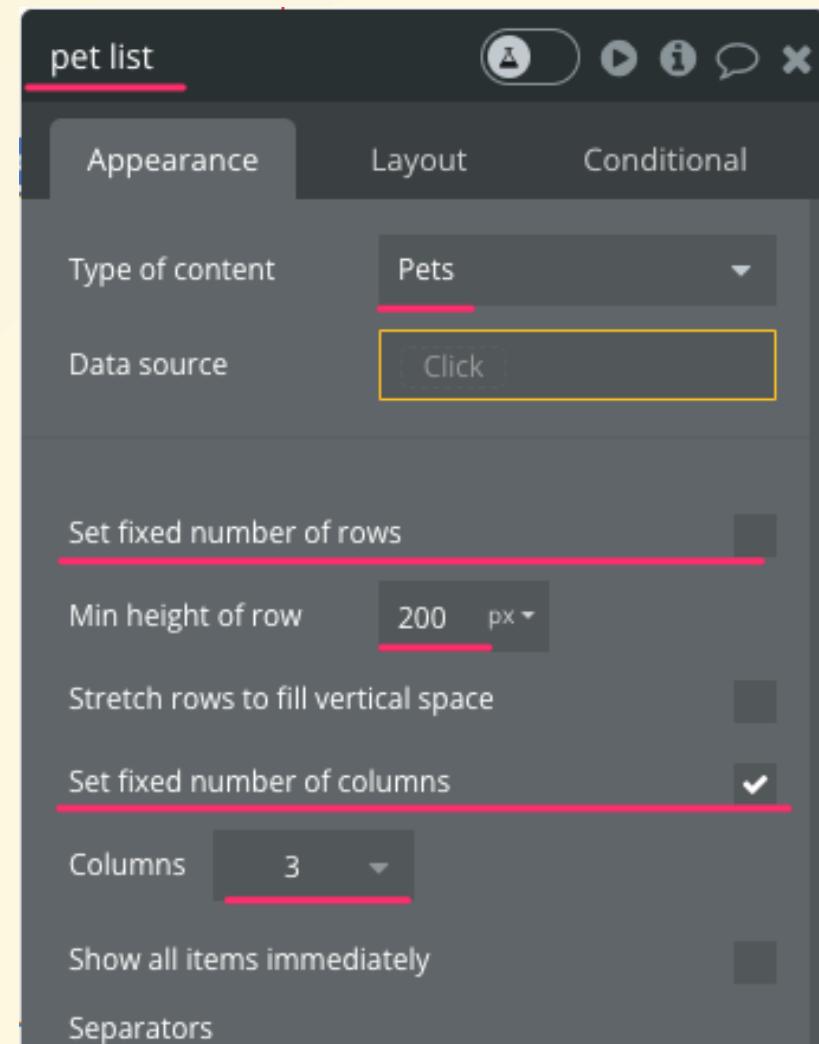


- Specify the number of rows and columns in the list

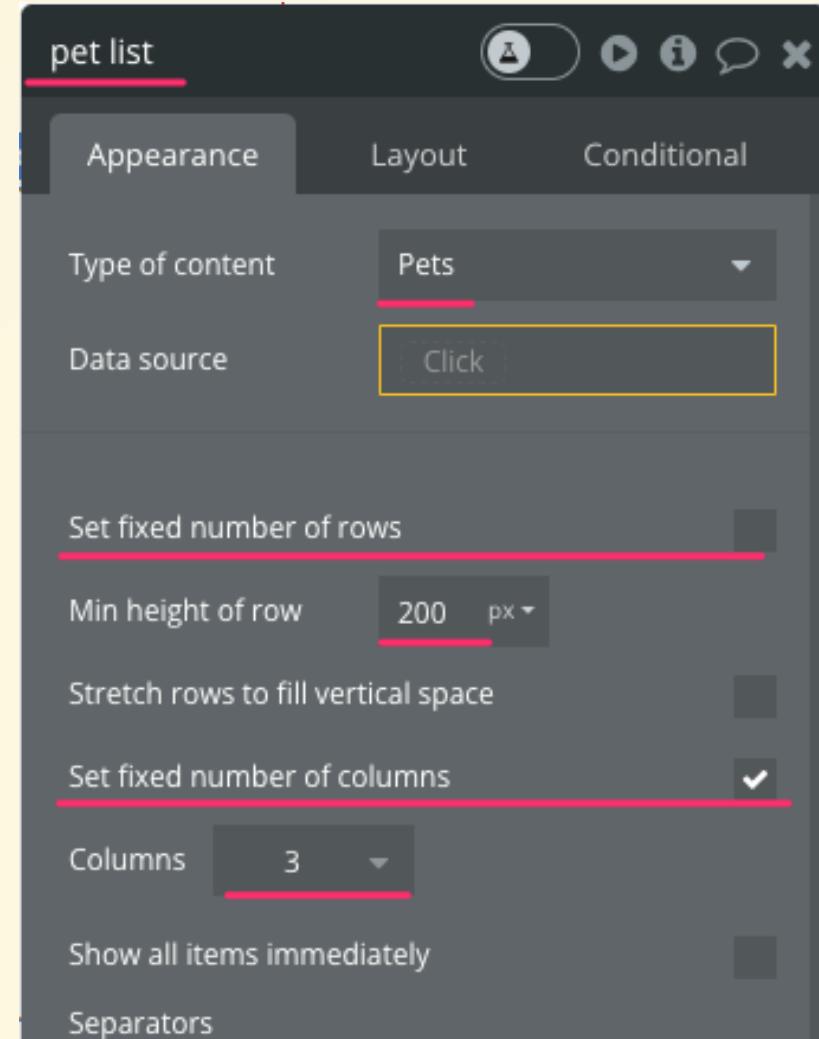
- Uncheck

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

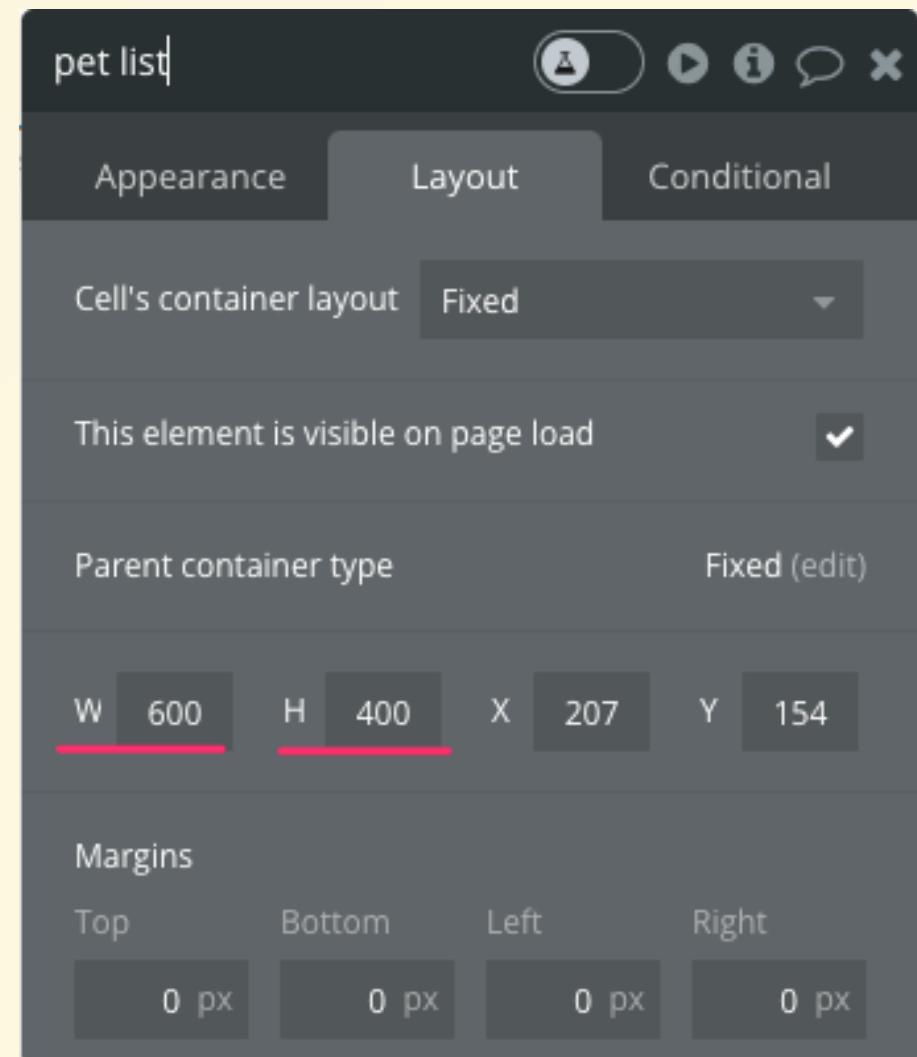
- Instead, specify the minimum height of a row
 - In this case, set it to "200px".



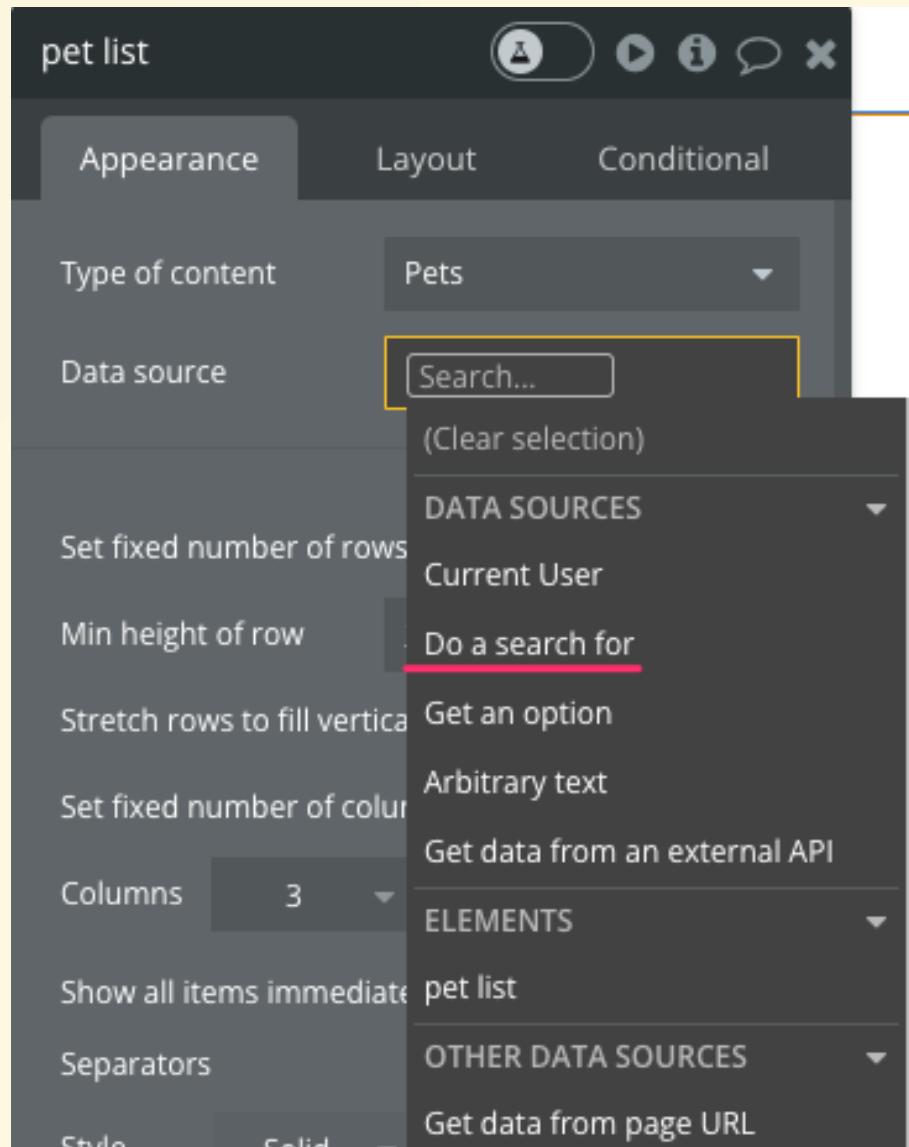
- Leave the Set fixed number of columns checkbox ON since we want to display 3 images per row.
- Set the value of Columns to "3



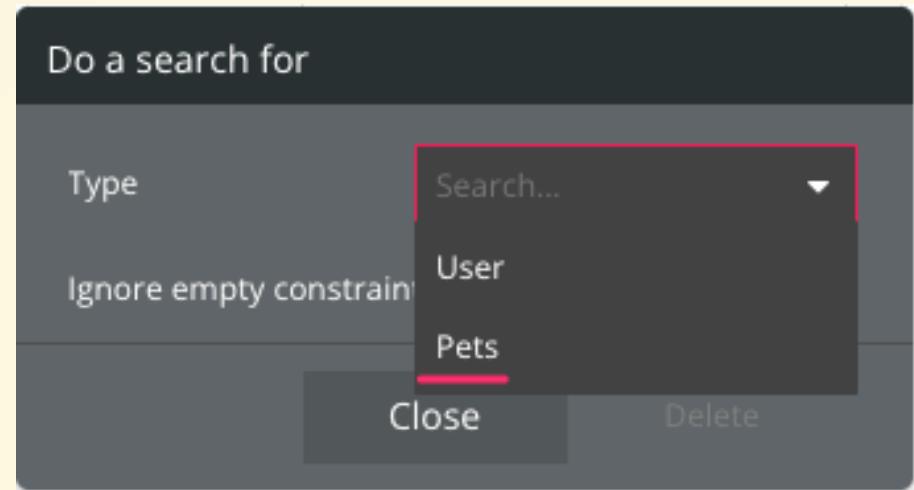
- Select **Layout** and set the value of **H** to "400" to keep the height of the entire table for two rows.
- Set the value of **W** in **Layout** to "600" to make the image size per row square.



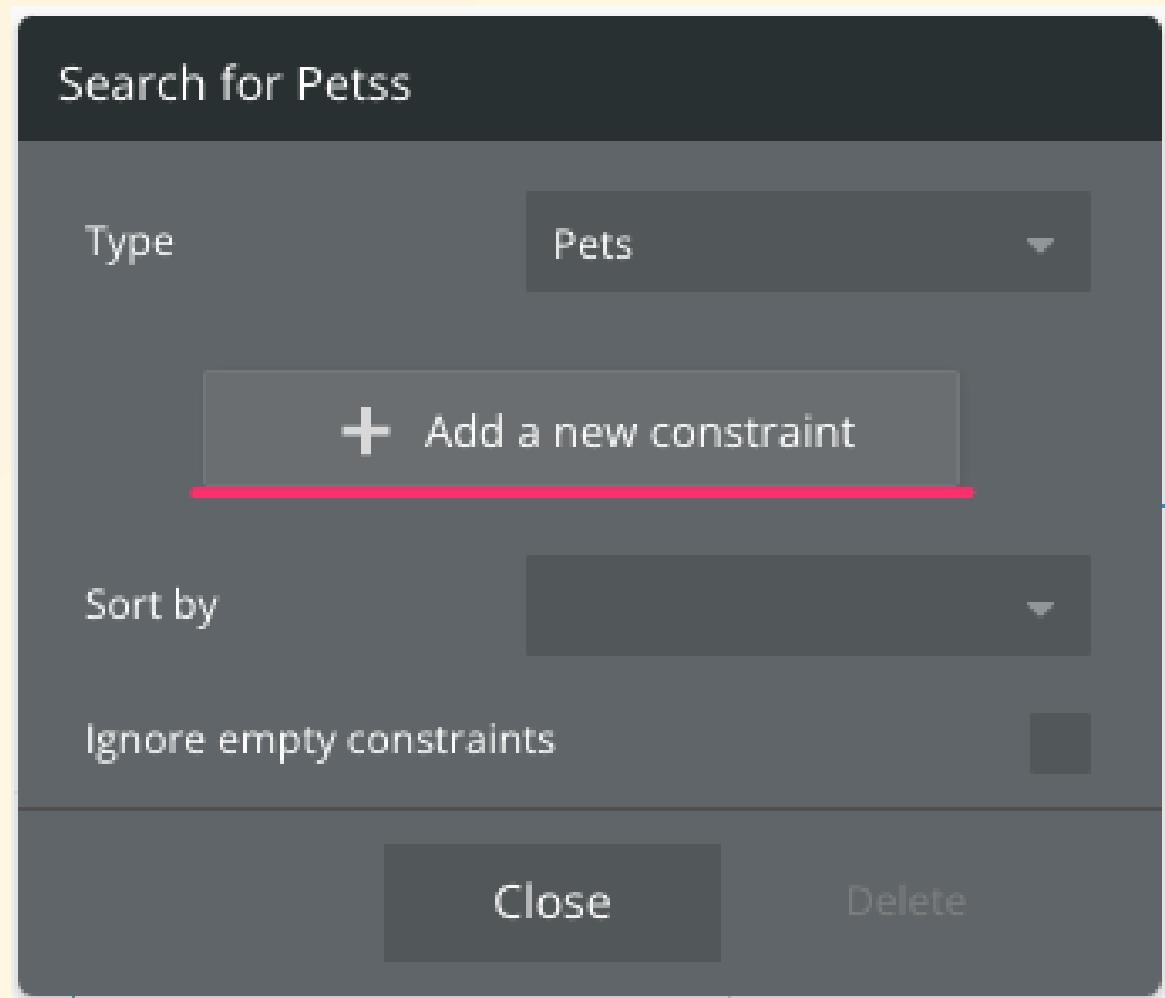
- Next, specify the data to be displayed in the list
- From the Appearance tab, click on Data source, then Do a search for.
 - This is to specify the data to be listed



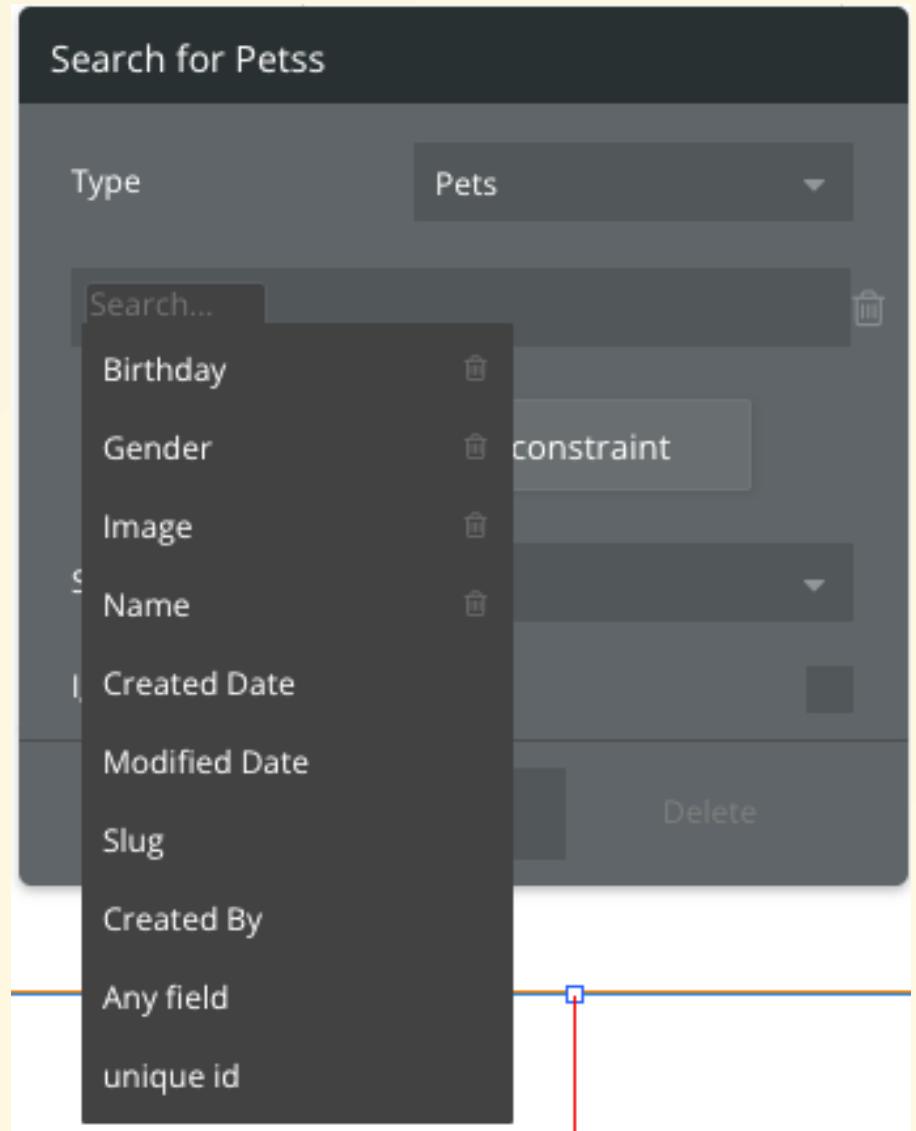
- Do a search for popup will appear, specify **Pets** for Type.



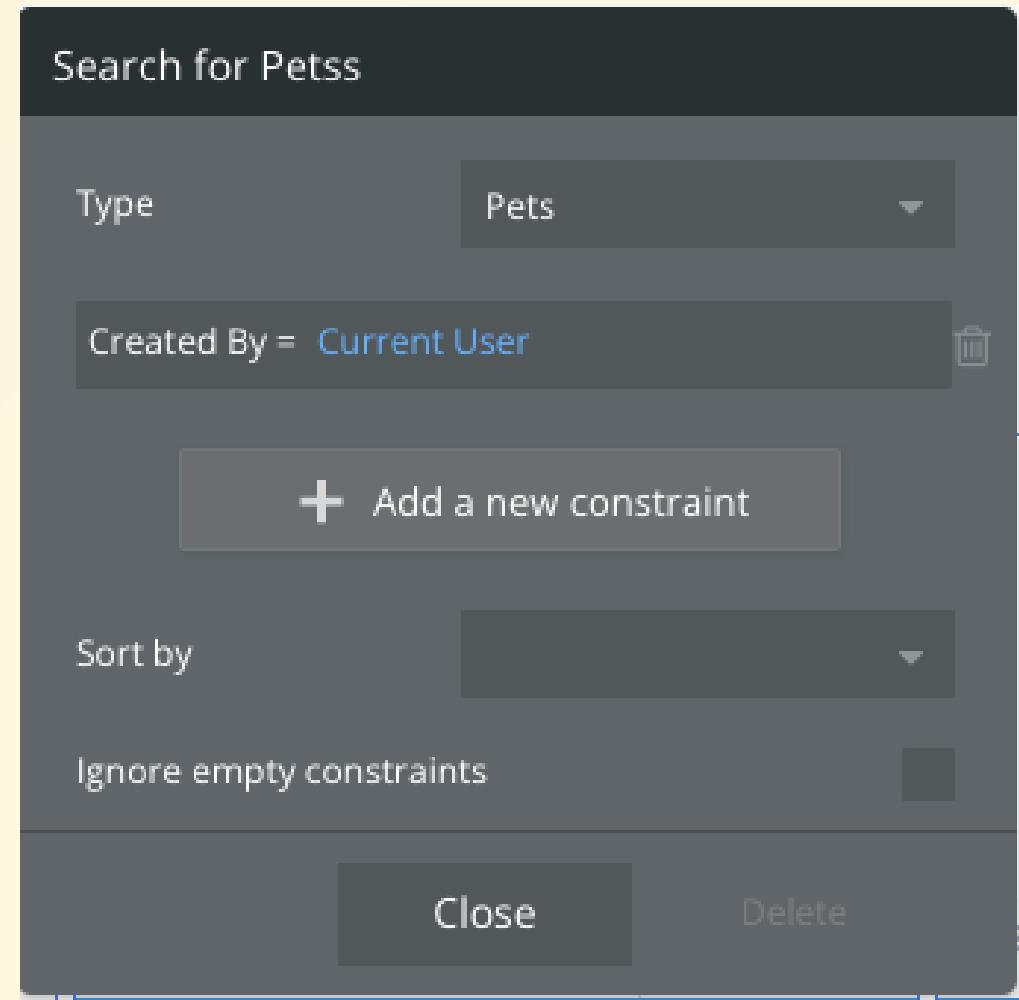
- If this is all, the list of pets can be displayed even if the user is not logged in, so we will add a constraint to display only pets registered by logged in users.
- Click "Add a new constraint" and add more conditions



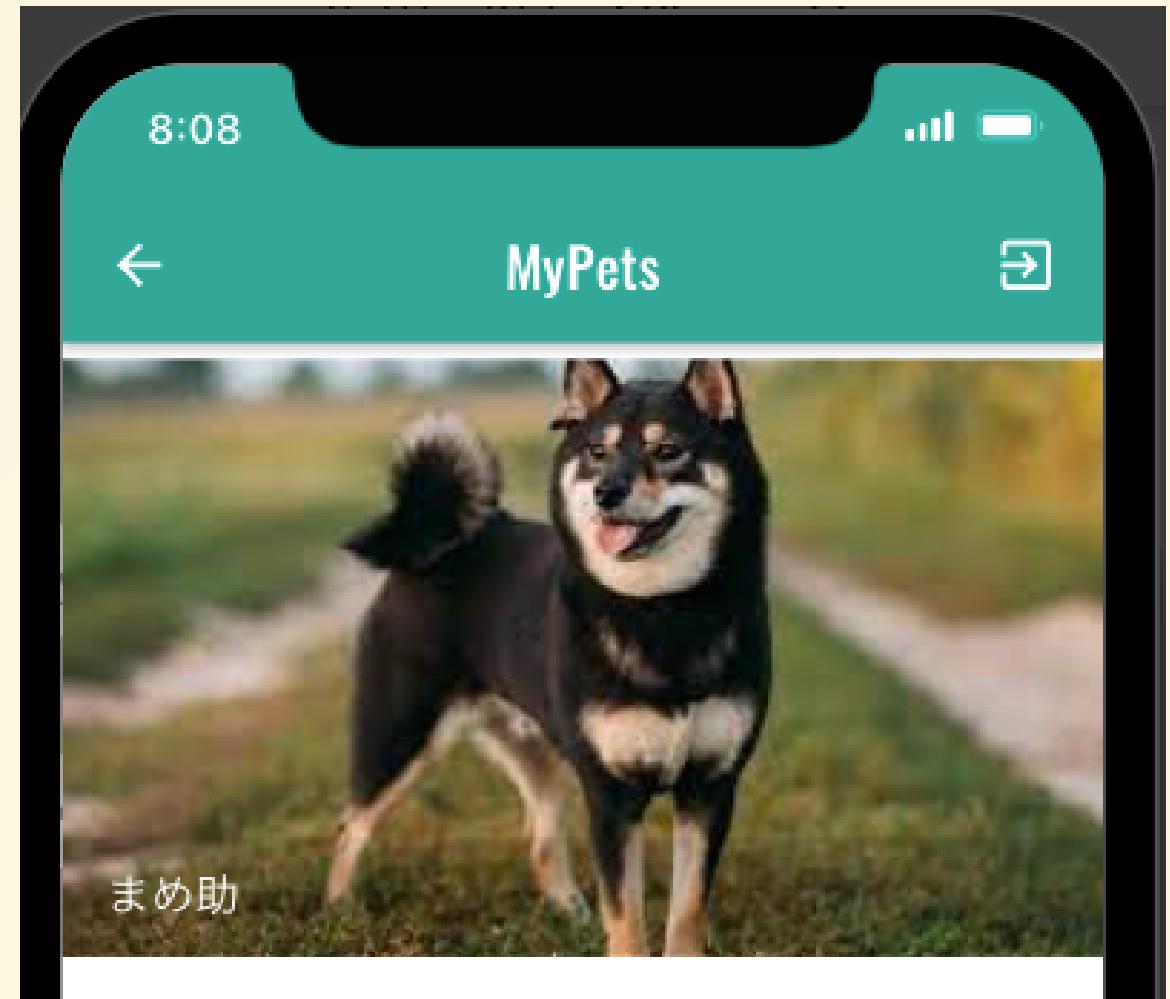
- What are the settings?
- The condition to be set is that the "Created By" of the pet must be the same as the "Current User."
- You will find the answer on the next page.



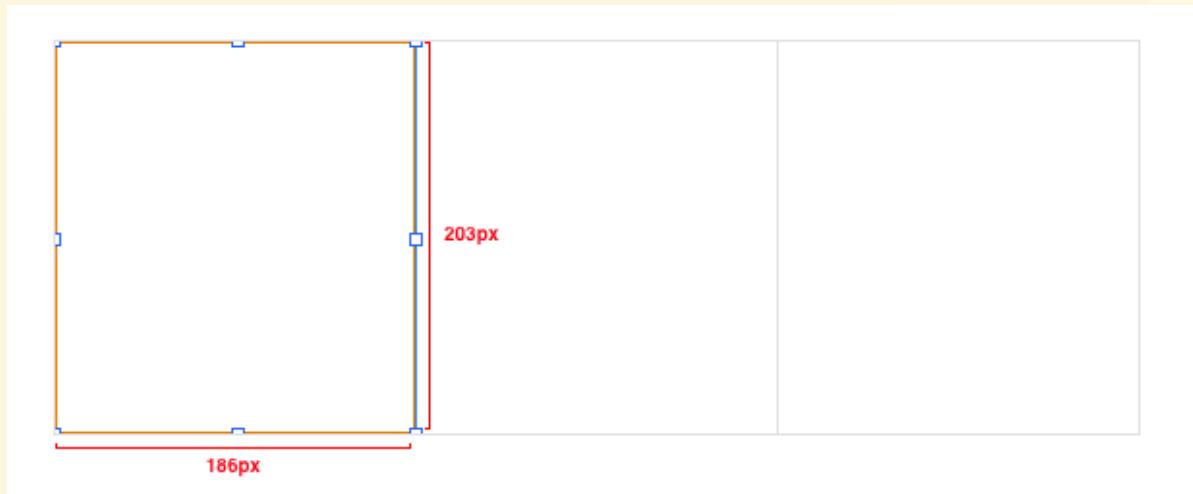
- It looks like this.
- This completes the specification of the data to be displayed in the list.



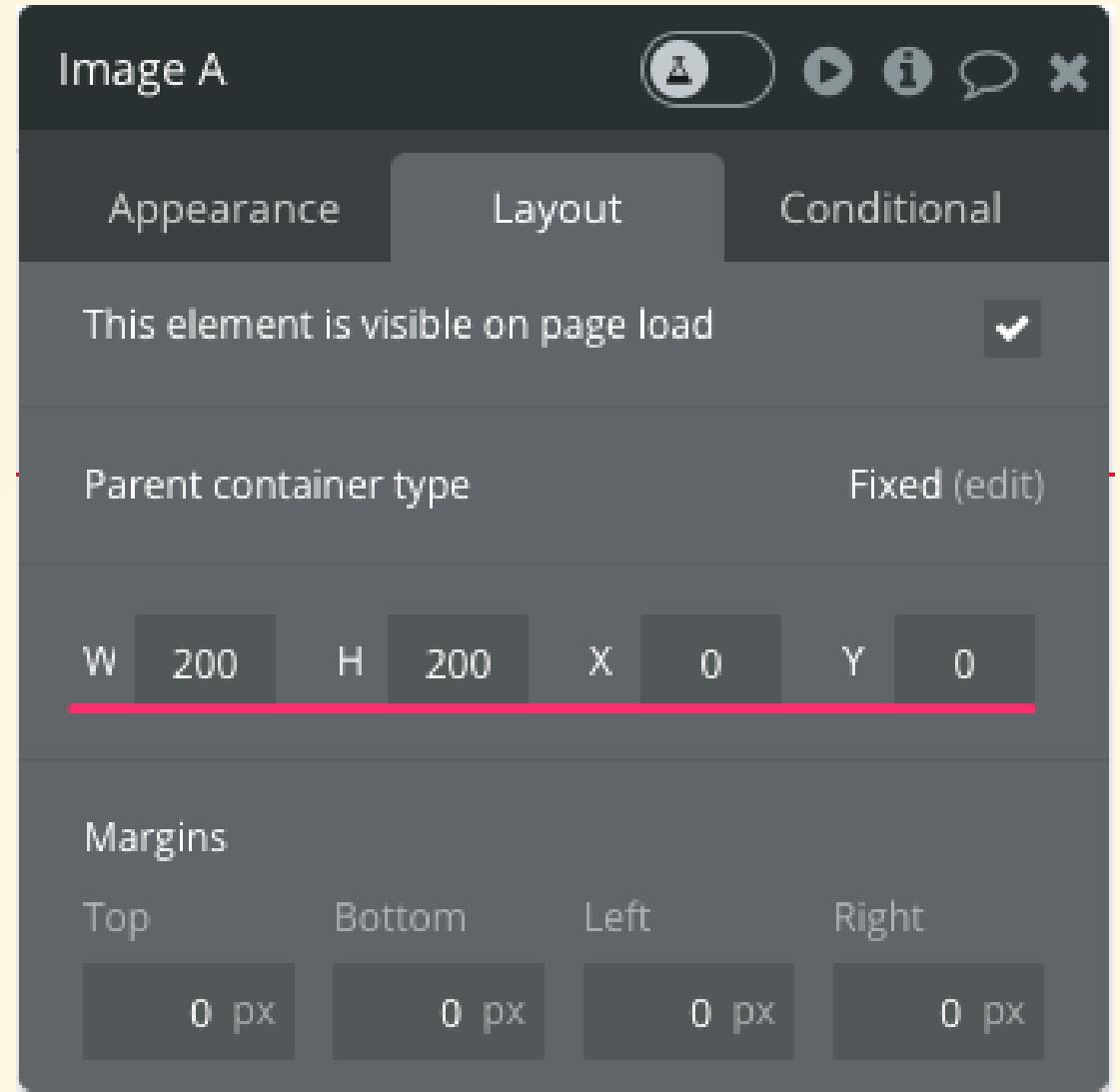
- Next, let's set up the content to be displayed repeatedly
- Let's display the image and name as we did for Adalo



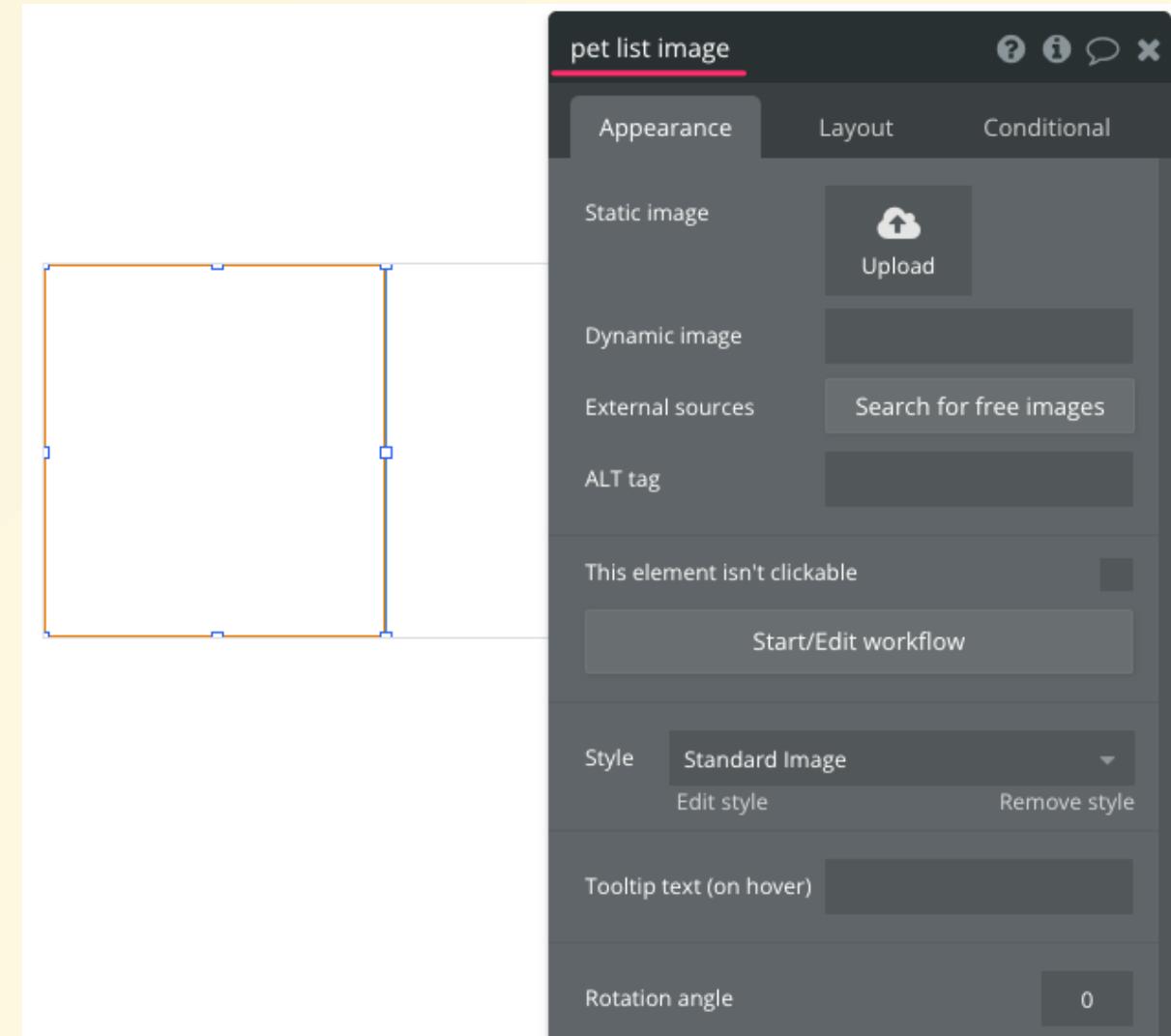
- First, select **Image** from **Visual elements** and drag it to the right panel.
 - If you drag the image into the **Repeating Group**, you will not have to move it later.
 - It is also possible to include it in the Repeating Group later.



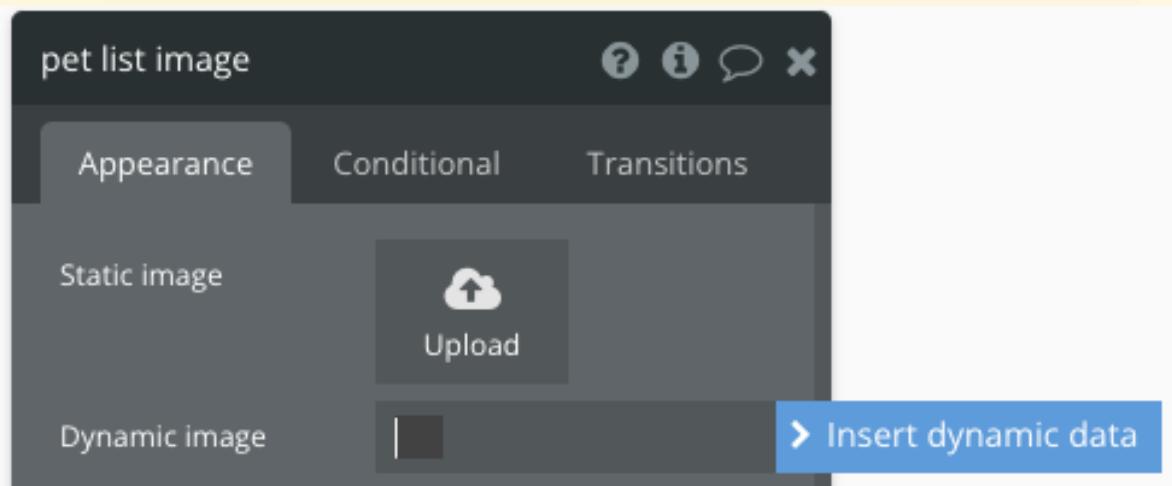
- Once placed in the Repeating Group, adjust the image size and position in the Repeating Group from the **Layout** tab.
- W: 200 / H: 200 / X: 0 / Y: 0**



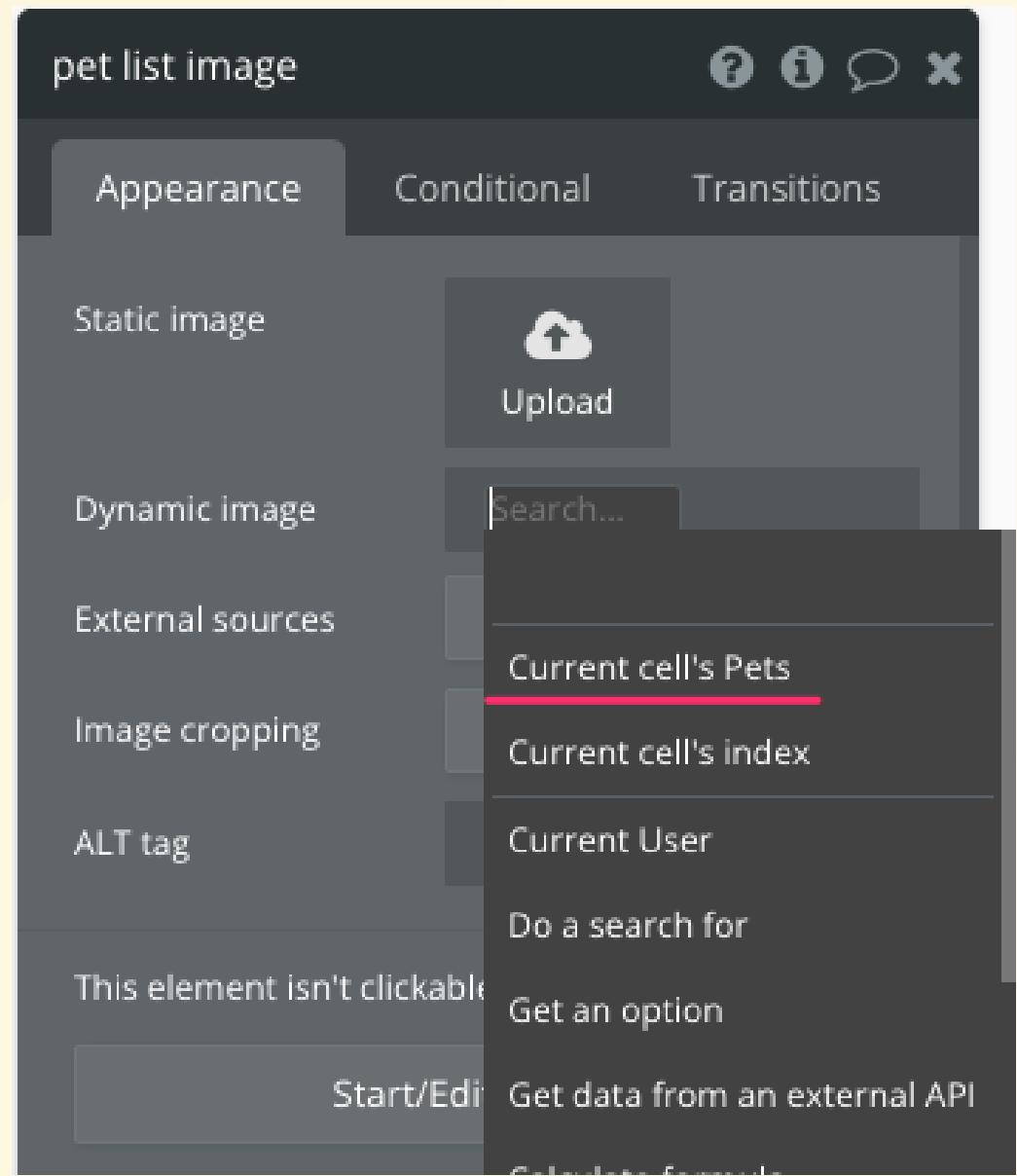
- The element name should be pet list image.



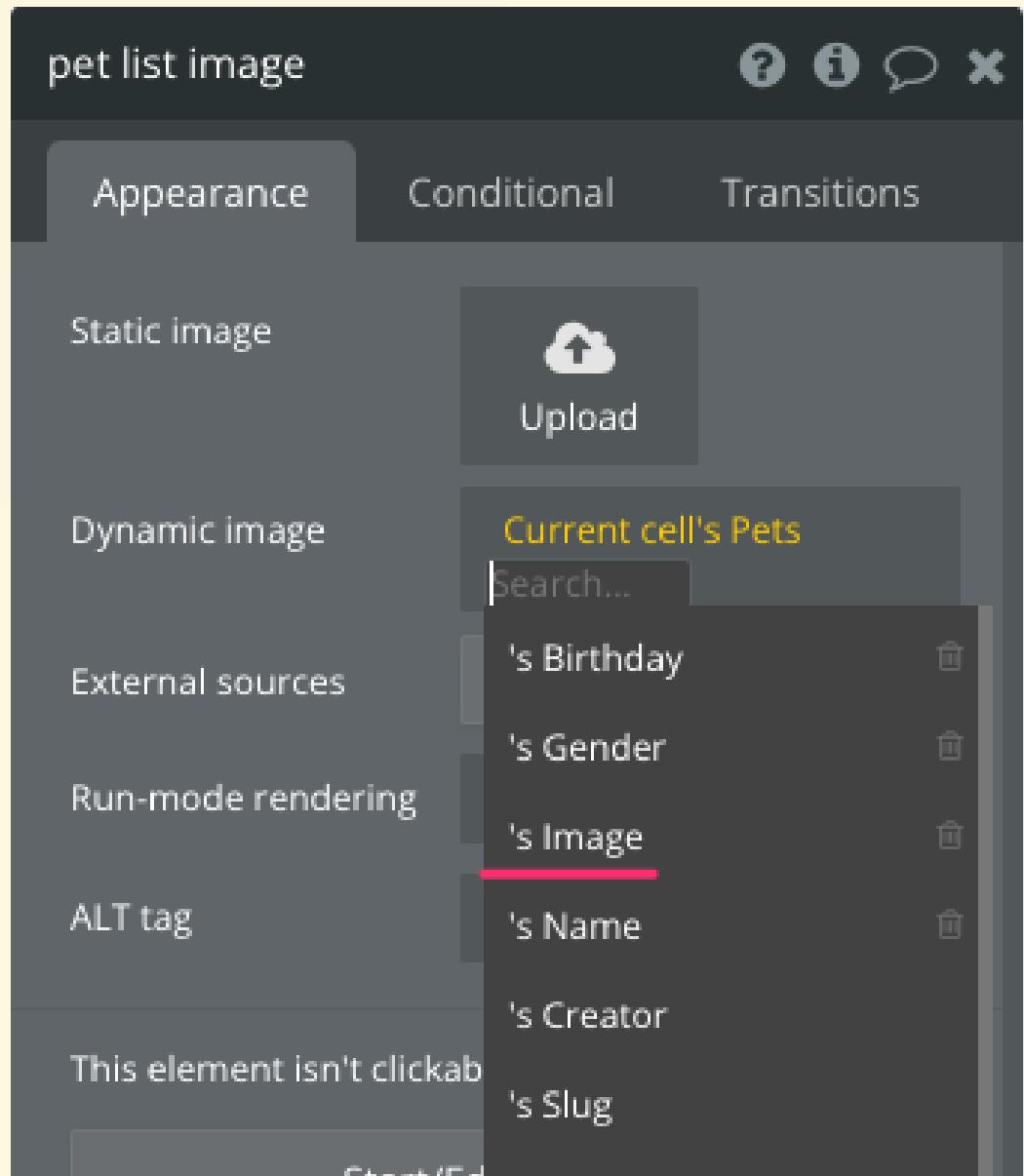
- After placing the element, select "Dynamic image" from the Advanced popup that appears.
- Click on the **Insert dynamic data** button.
 - This function is used when you want to display values or images dynamically, rather than just fixed values or images.



- 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.



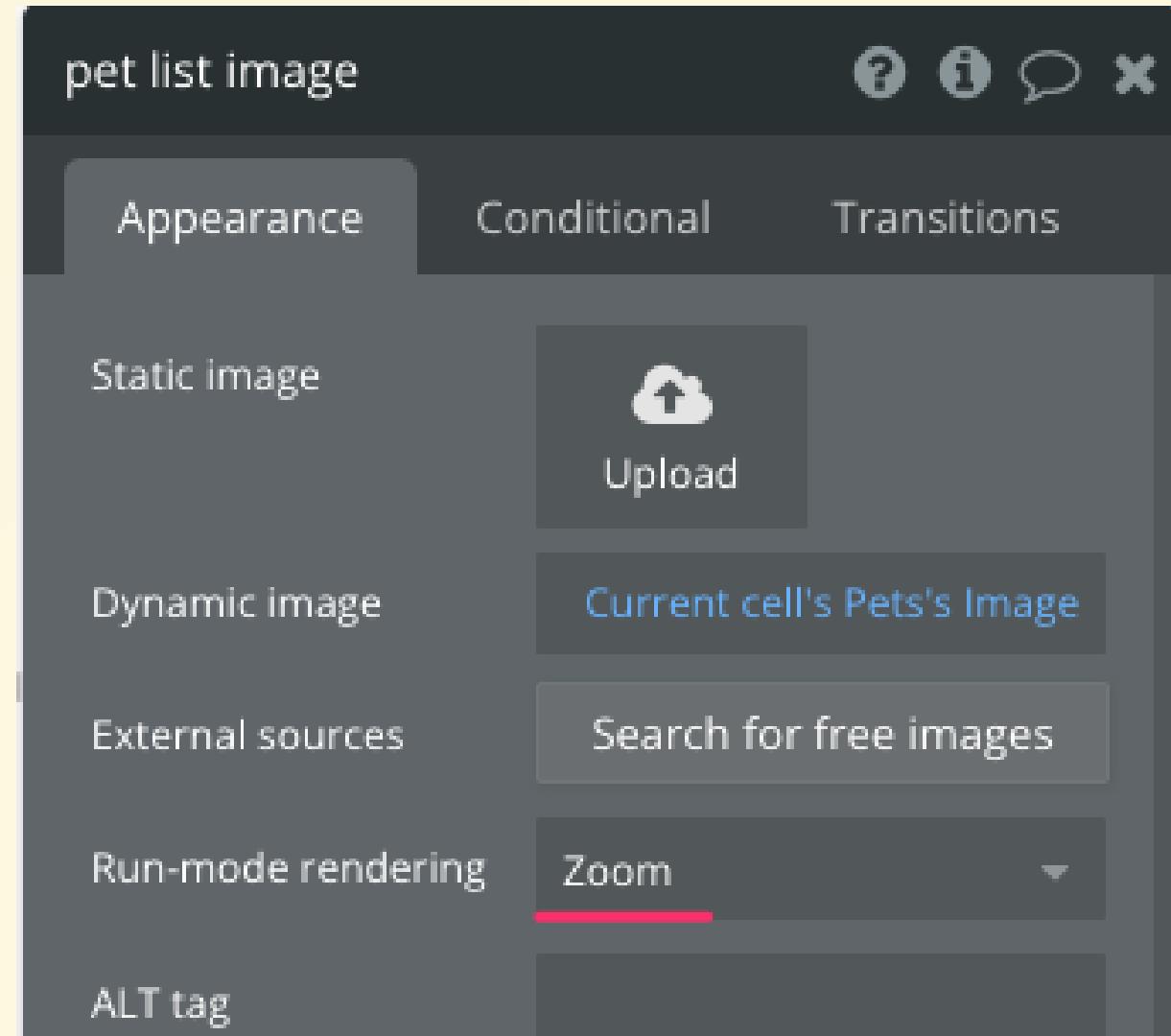
- Select `Current cell's Pets` and a further pull-down will appear showing the fields that the Pets type has, so click on `'s Image`.
 - This will cause the image of the pet in the current cell to be displayed, so that if there are multiple pets, each cell will have an image of the pet.



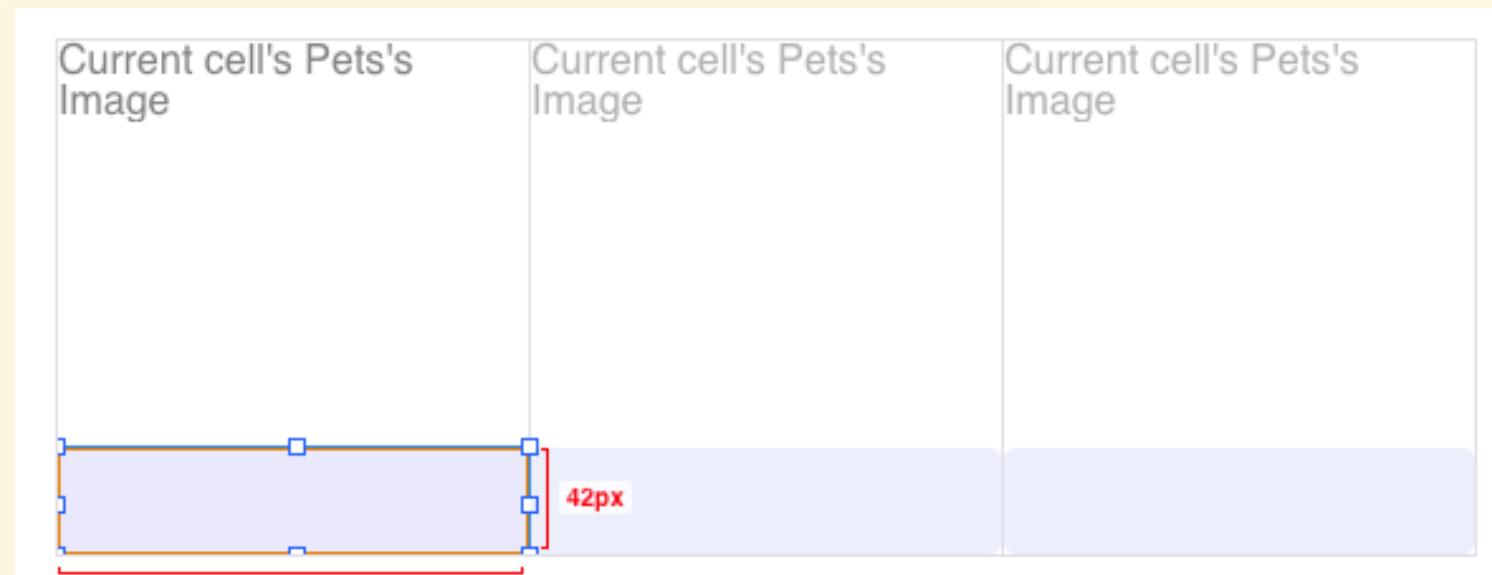
- We have just set the image to fill the entire width of the cell, but we should change the

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

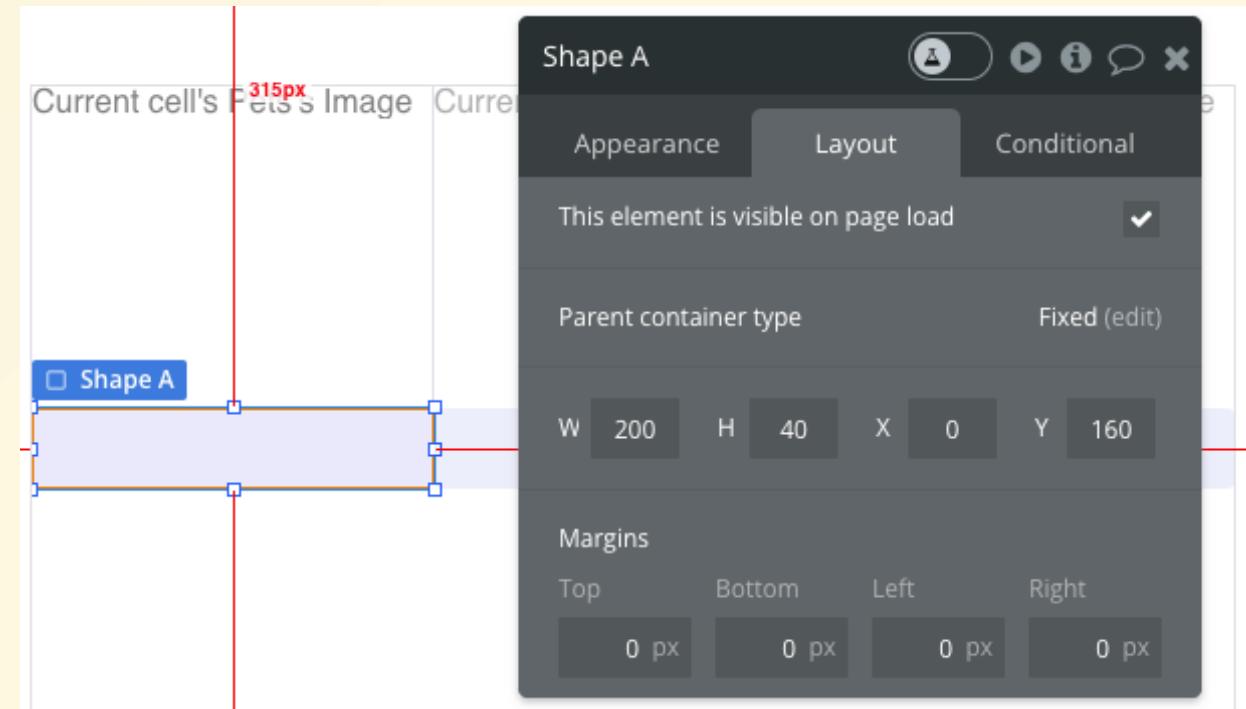
- Otherwise, the image will automatically change its aspect ratio in the cell, so change it to Zoom.



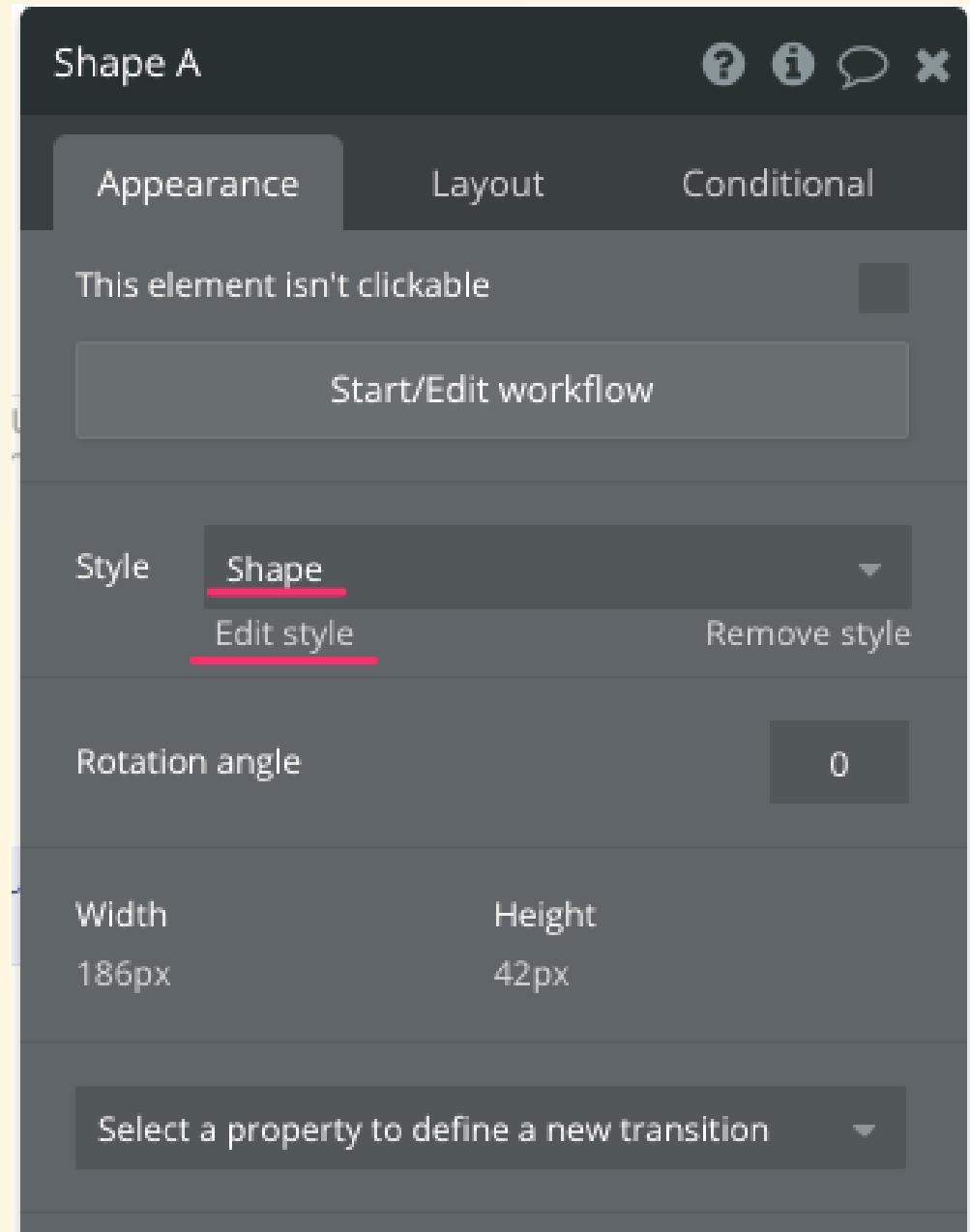
- Next, let's set the background for the name that will appear below the image
- Select **Shape** from **Visual elements** and drag it to the right panel.
 - Drag it into the "Repeating Group" as well.
 - After placing it, move it to the bottom of the Repeating Group and drag it to the full width.



- The position of this placed element is also set from the **Layout** tab.
 - W**: 200 (Same width as image)
 - H**: 40
 - X**: 0
 - Y**: 160 (To place an object of height 40 all the way down, $200 - 40 = 160$)



- 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



- You will then be taken to the Styles tab, where you can edit the style of the "Shape" you have just selected.

The screenshot shows the Figma interface in the 'Styles' tab. The left sidebar lists various element styles, and the main area displays the 'Shape' style. The 'Appearance' panel on the right shows settings like Opacity (100%), Background style (Flat color), and Color (Primary #0205D3). A preview of the shape is shown as a light purple rounded rectangle.

Page: pet_list

Edit Saved

Design Workflow Data Styles Plugins Settings Logs

Element styles Style variables

Element type Search by name

Apply theme Add style

Shape

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 (Default)

Dropdown - Standard Dropdown (Default)

File Uploader - Standard Uploader (Default)

Floating Group -

Remove this style as a default for all new Shapes

Find all elements using this style

Appearance Layout Conditional

Opacity 100 %

Background style Flat color

Color Primary (#0205D3)

Define each border independently

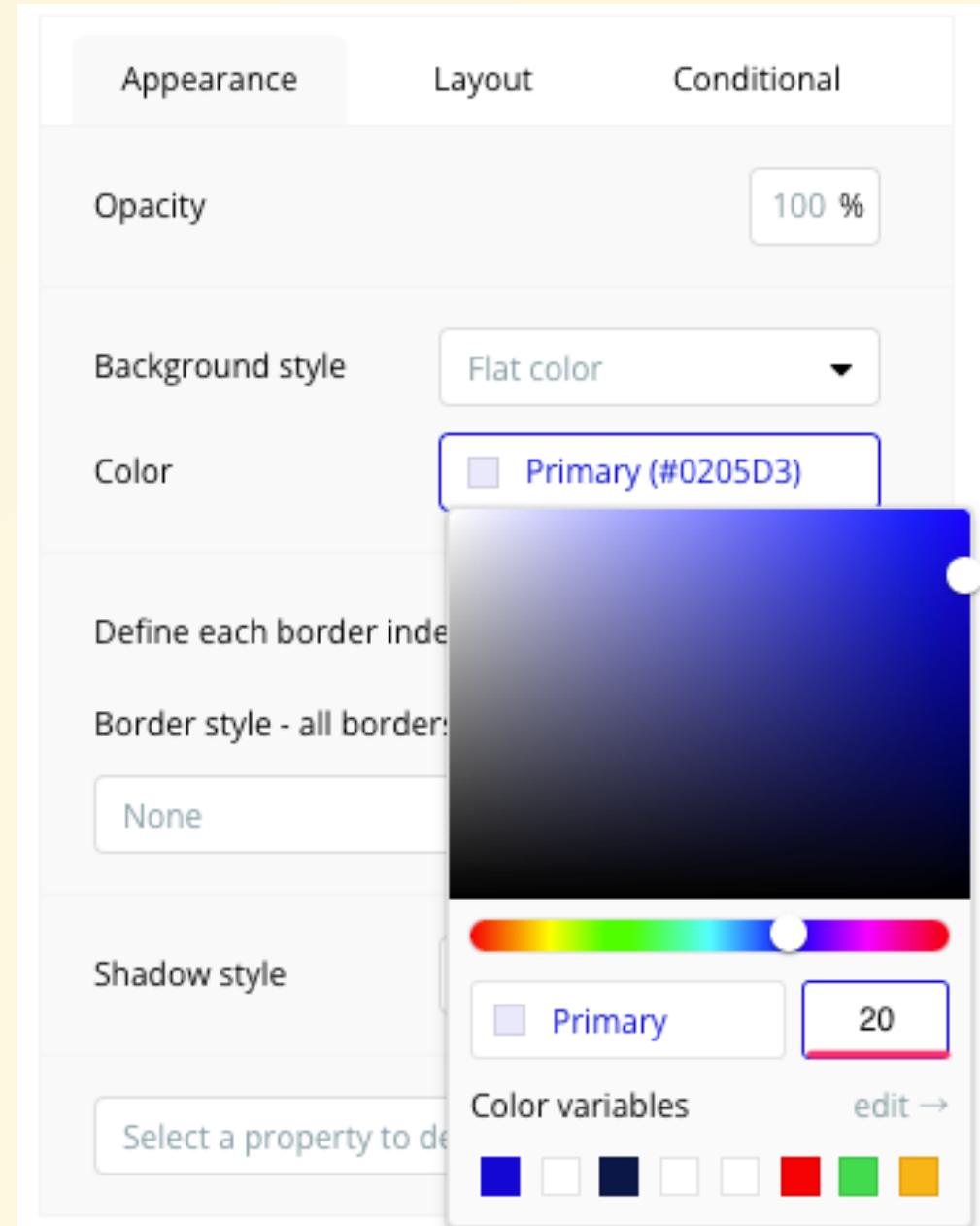
Border style - all borders Roundness

None 5

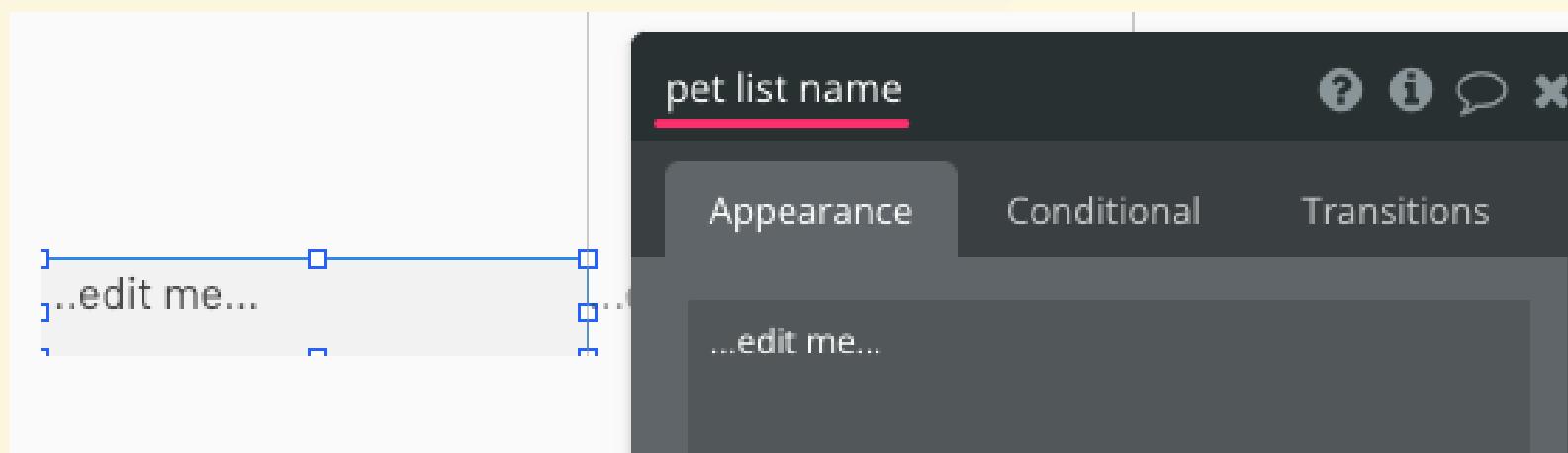
Shadow style None

Select a property to define a new transition

- 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 10% to 20%. 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**.



- Like the pet image, the pet name uses Dynamic data

.... . edit me... Click on

- the

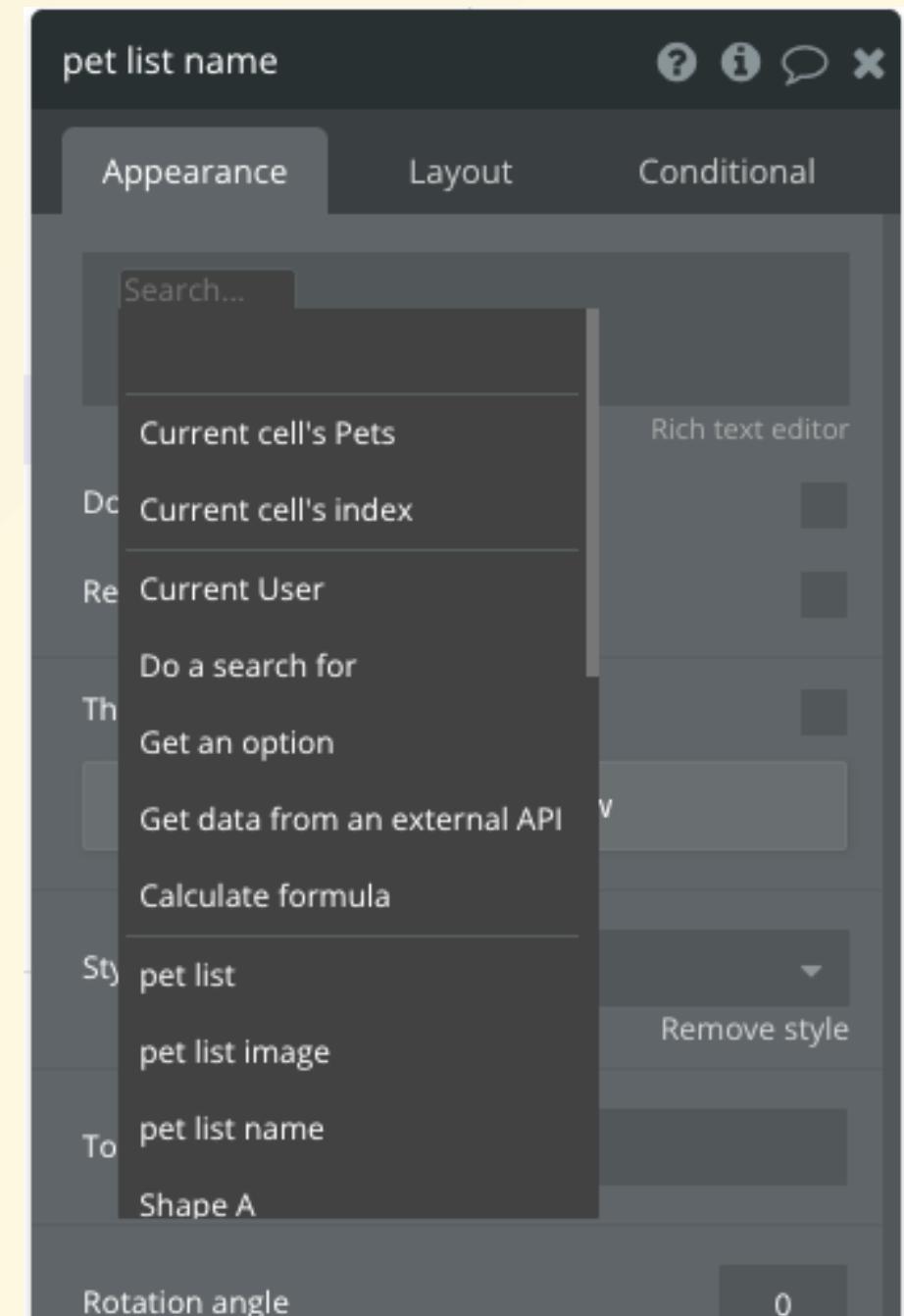
...edit me...

and you will see Insert
dynamic data

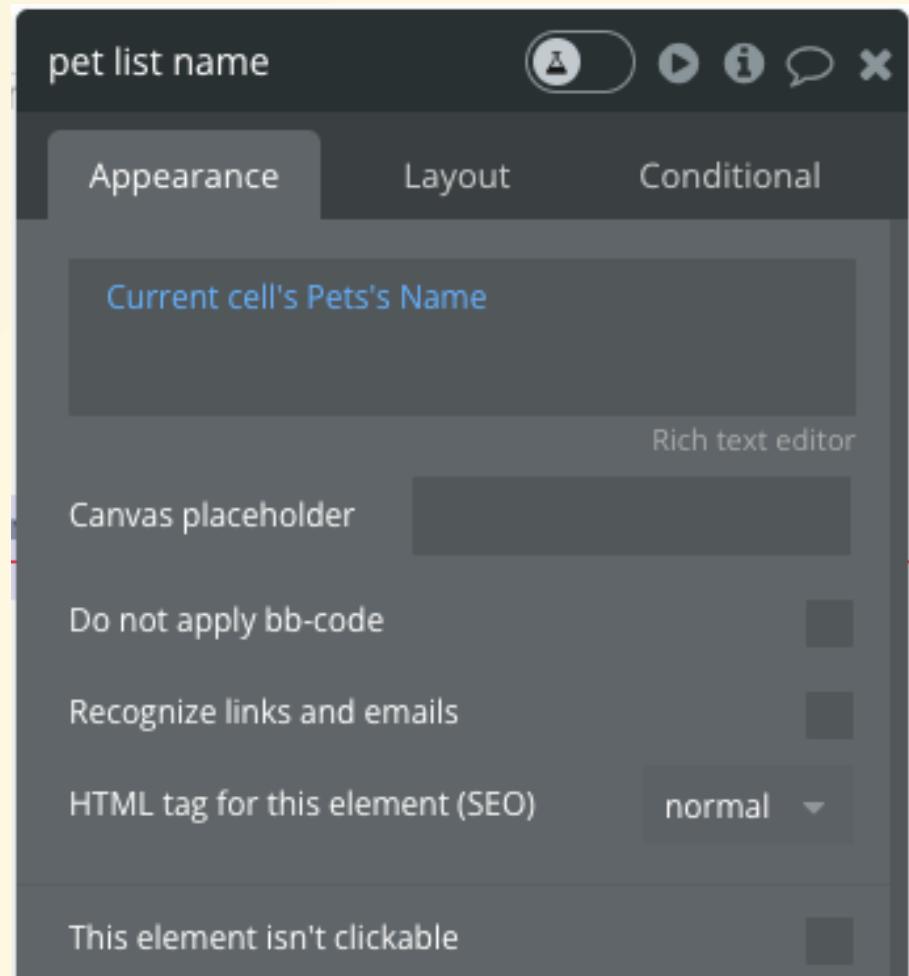
as you did for the image,
so let's set the

pet name in the current
cell` from there.

- You will find the answer on
the next page.

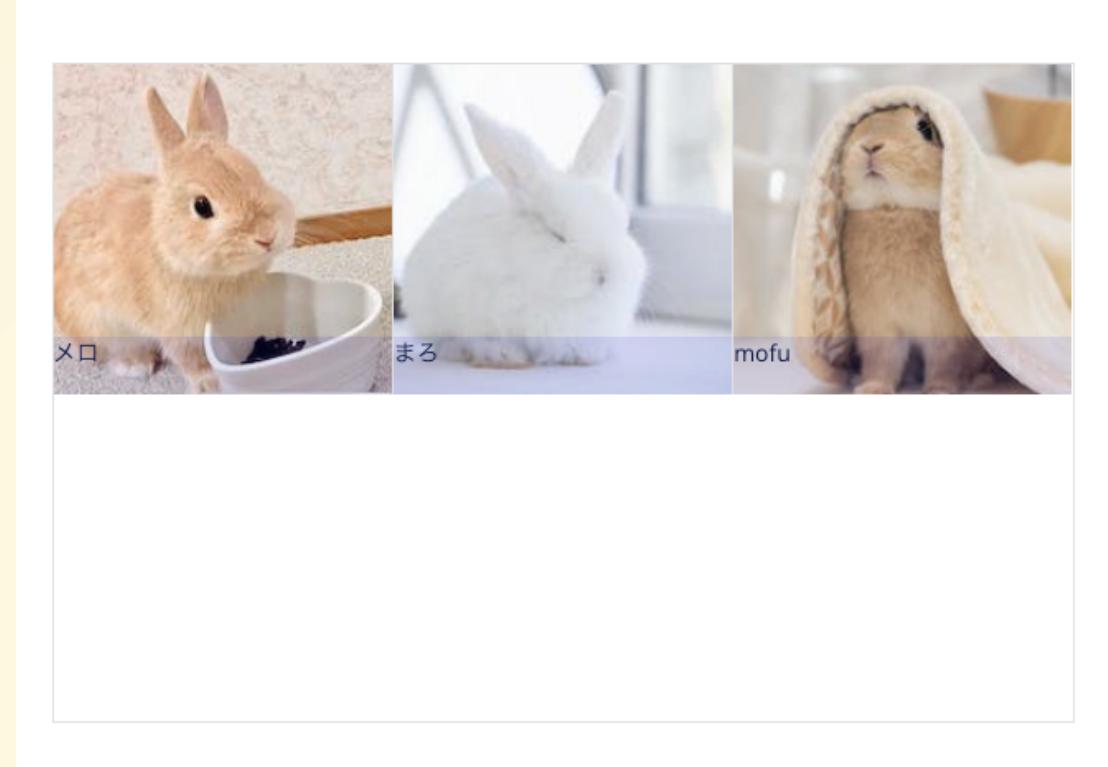


- Current cell's Pets --> 's Name Select
- This completes the configuration of the content to be displayed in the listing.



If you've made it this far, let's preview it!

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



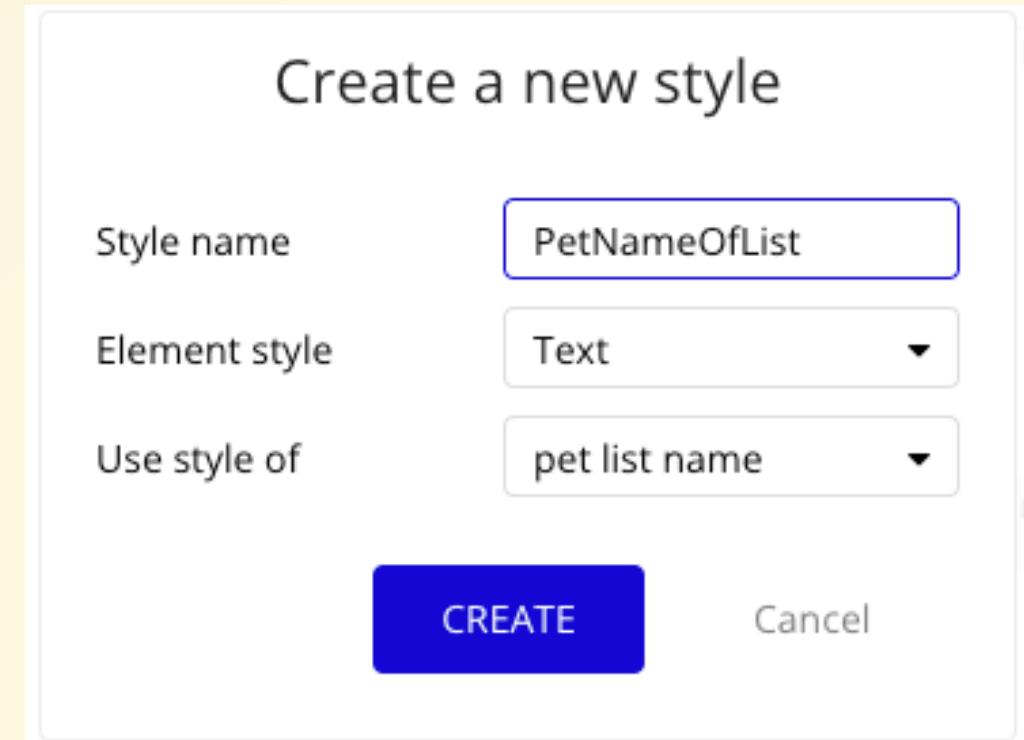
Exercise 1

- Since the pet's name is in the upper left corner, let's center it and make the text a little bigger!



- 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

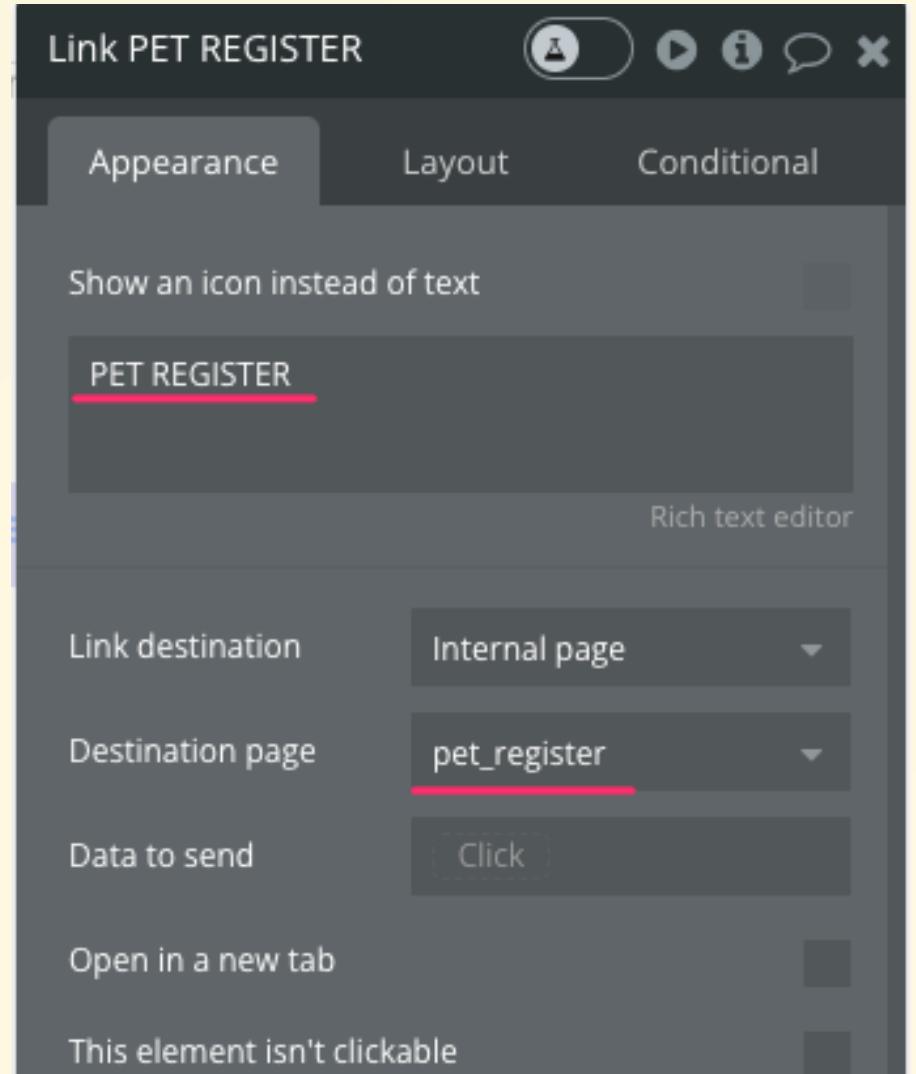
- Now that the pet list screen has been created, let's prepare a lead from the list screen to the registration screen.
- Let's prepare this lead to the top of the pet list screen.



- 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

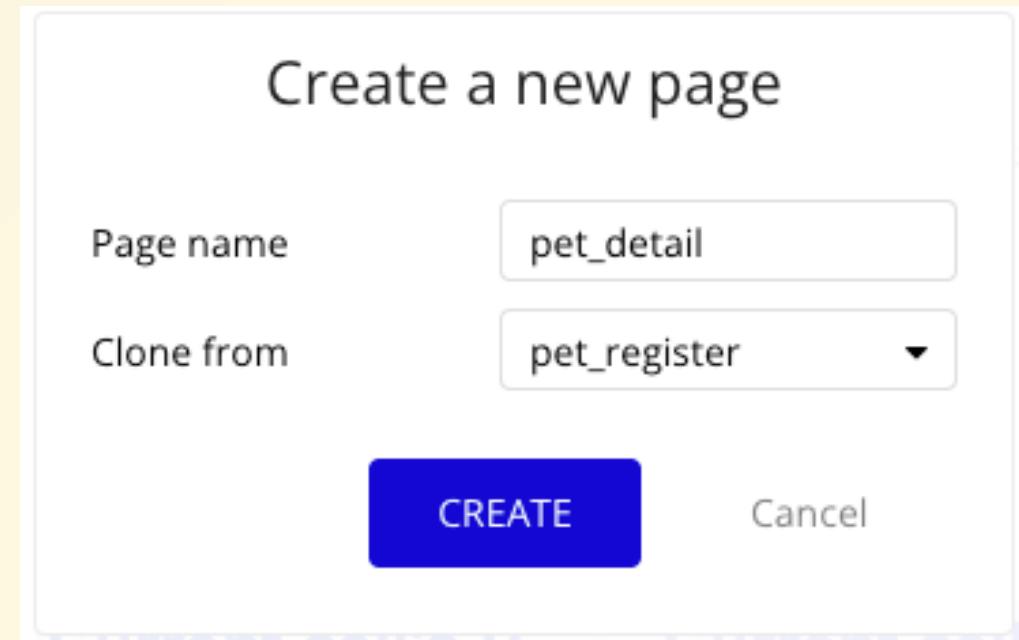
Gender

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`.
- Clone from" should be `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

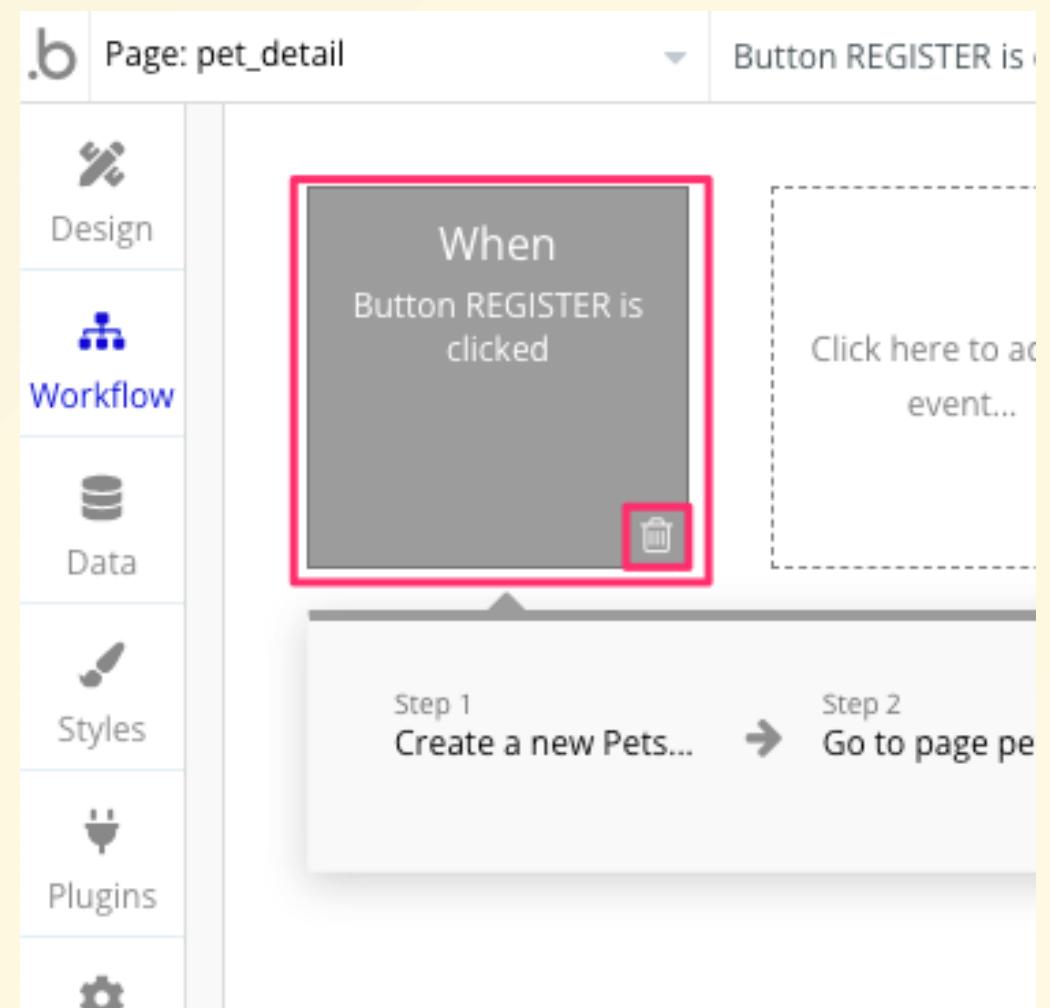
Image

Birthday

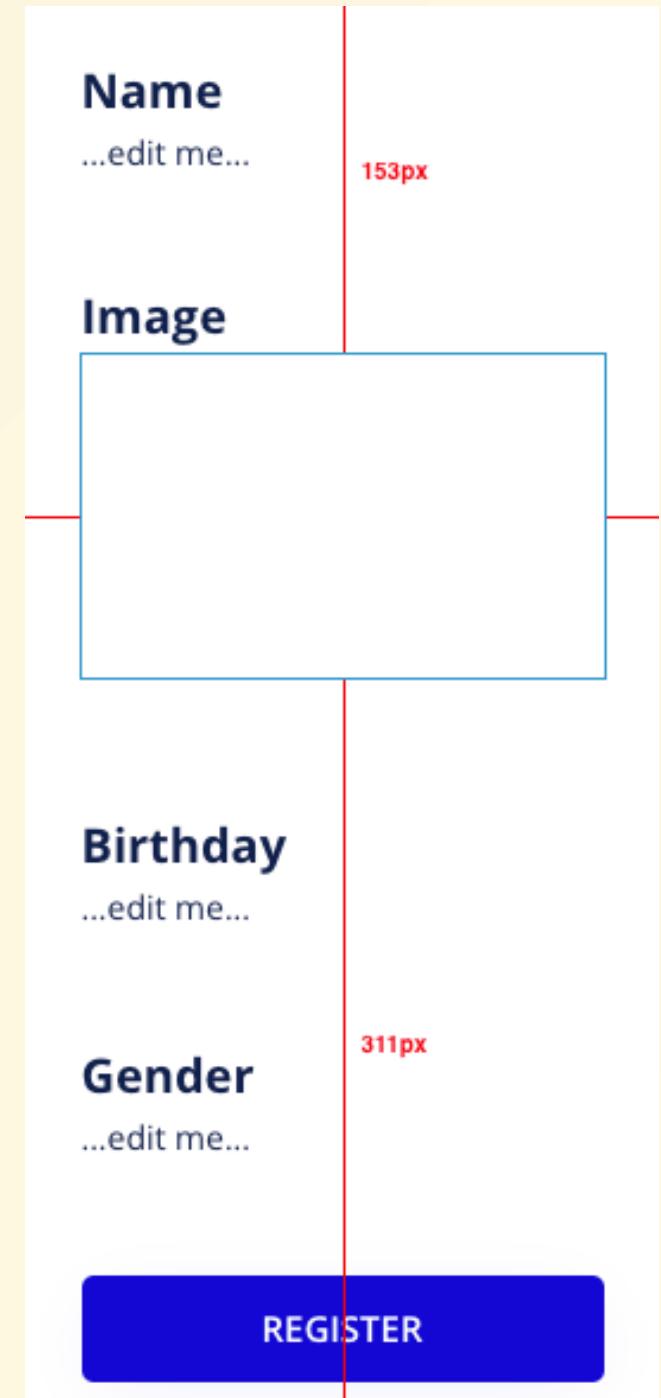
Gender

REGISTER

- In addition, delete the settings when the REGISTER button is pressed from the Workflow tab
- When you select a workflow with the REGISTER button, a trash can icon will appear in the lower right corner, and you can delete the workflow definition by pressing the trash can icon.

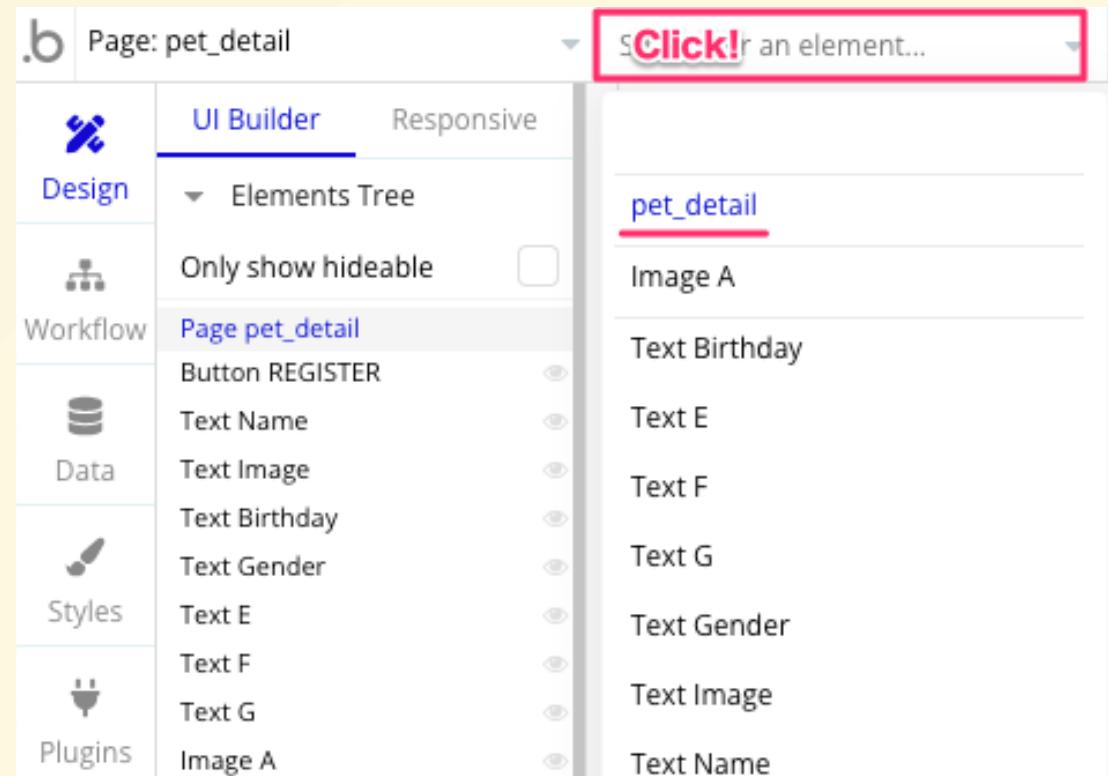


- Next, let's place the elements for display
- Let's drag each element from "Visual elements"
 - Name: Text
 - Image: Image
 - Birthday: Text
 - Gender: Text
- All elements should have a width (w) of 220px

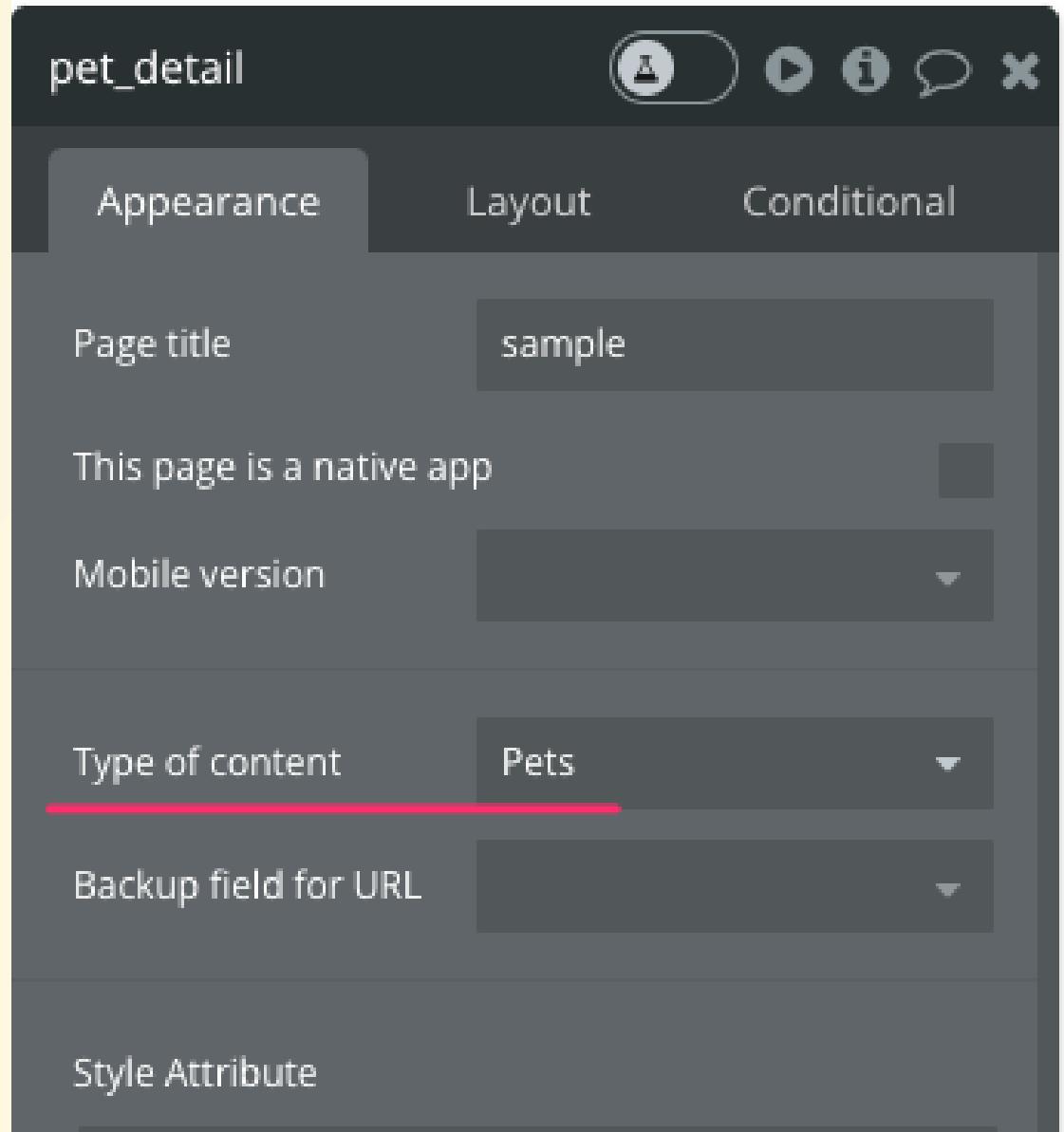


- Once the elements are ready, we will define the actual values to be referenced from the database
- First, we will display a dialog for advanced configuration of the screen itself

- 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 from 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.

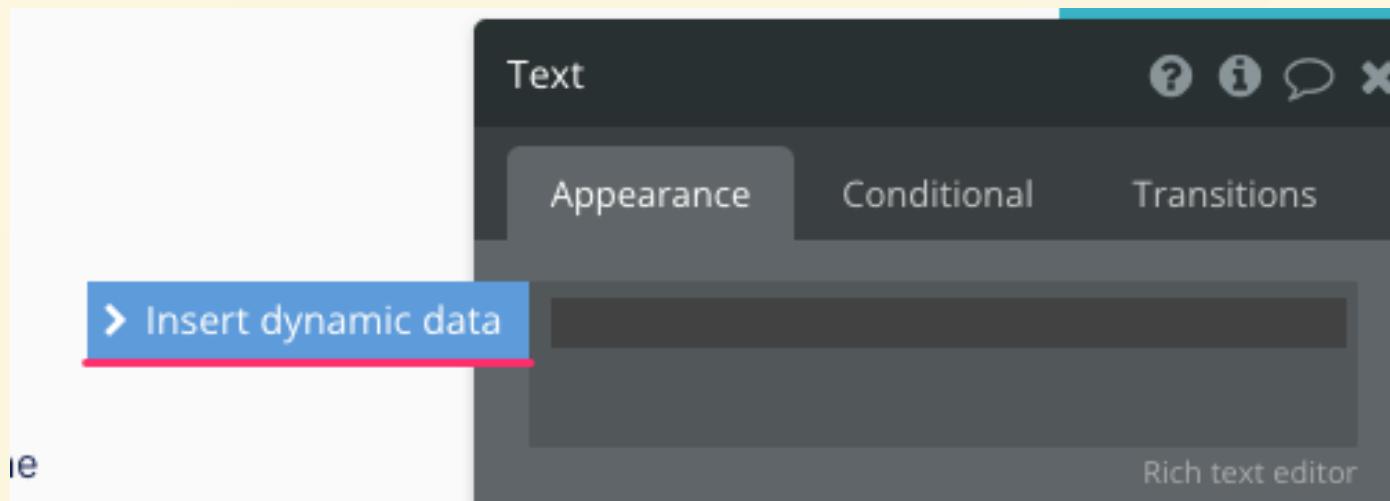


- By the way, although the type could be specified, the specific information on which pet (Porgy or Tama) is specified when moving from the list screen, which will be set later.

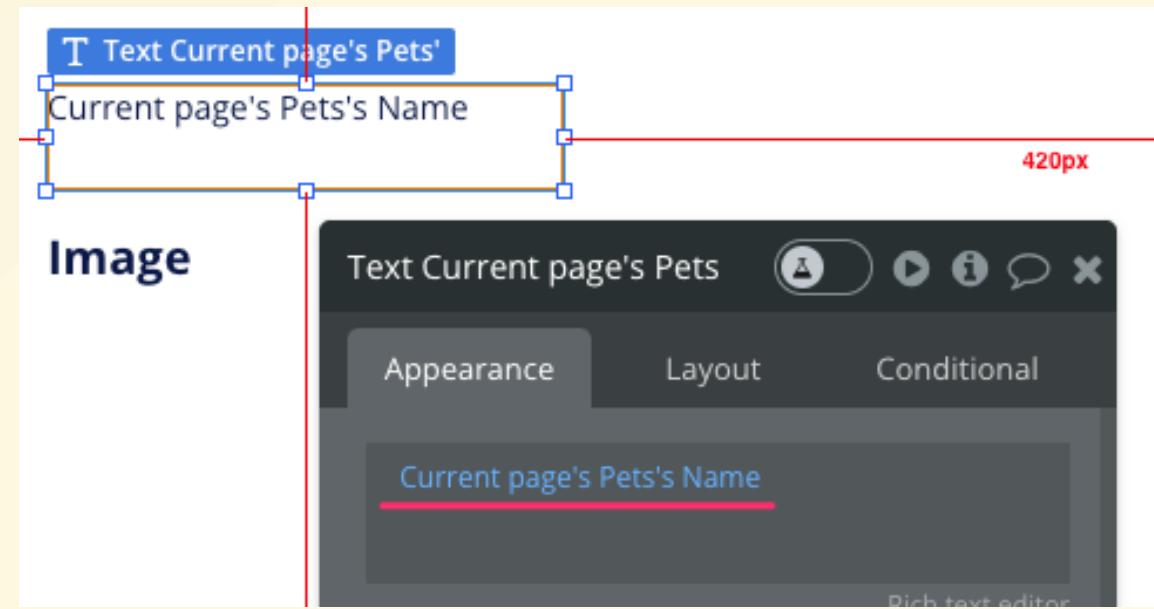
- Now let's link the elements for display with the Pets type field.
- Let's start with Name.
- Remember how we want to dynamically display values retrieved from the database, as in this case?



- Yes, I do! 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 an element while it is selected -->
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.

Name
Lorem ipsum...

Image

Birthday
Nov 12, 2022 10:00 pm

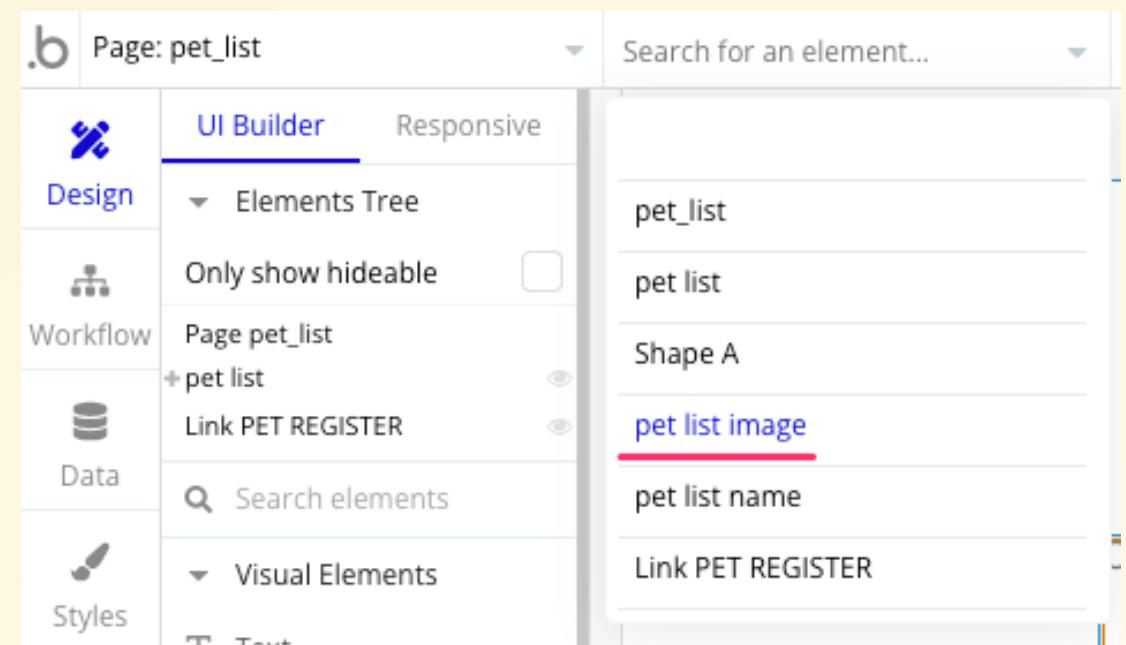
Gender
Lorem ipsum...

REGISTER

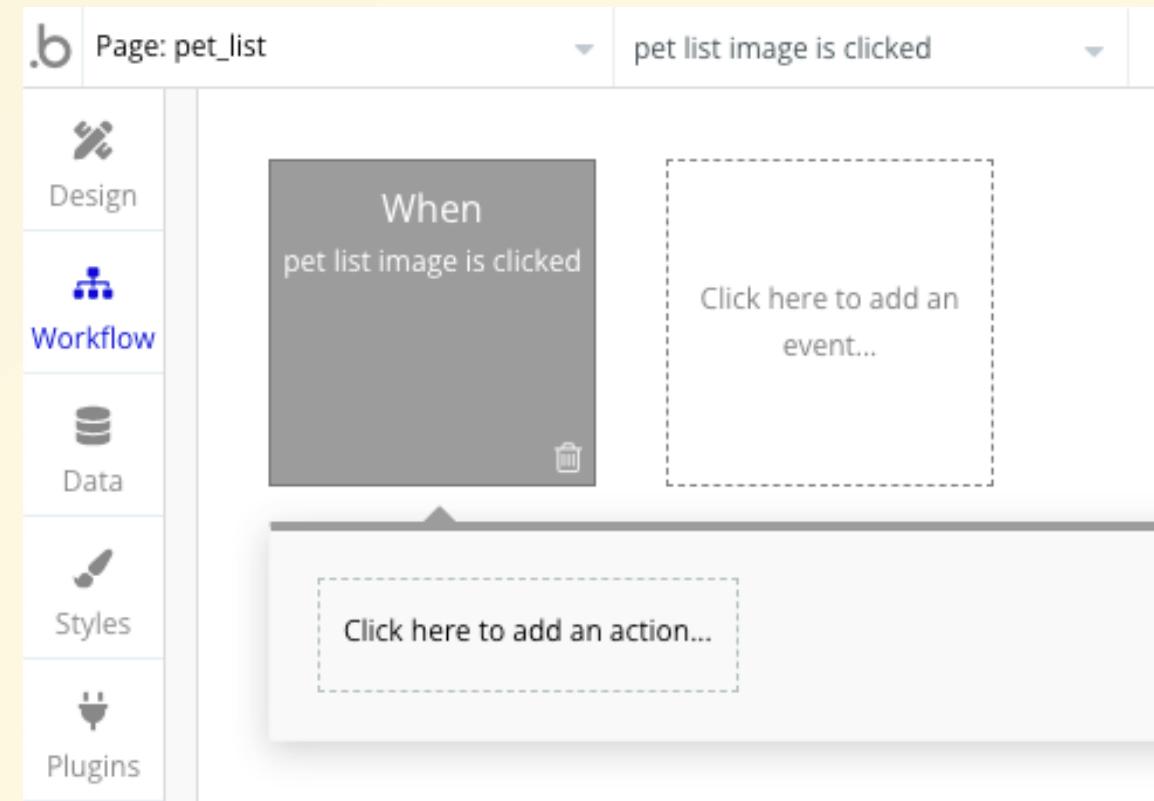
Now let's connect the list page with the detail page.

- Switch to the pet_list 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 on the pet list screen is clicked.
- So, let's set up a workflow for the pet image on the list screen.

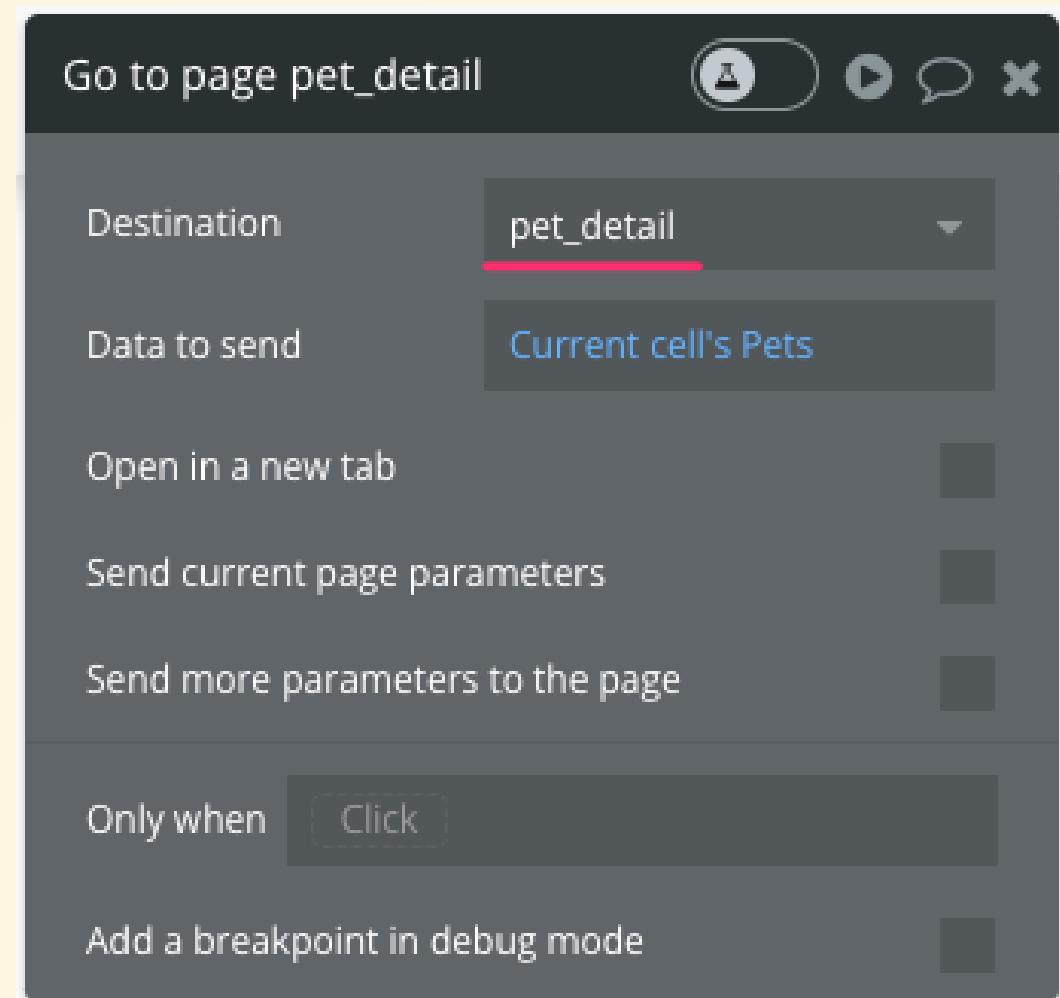
- Click on **pet list image** from the element list
- Click **Edit workflow** in the dialog for advanced settings



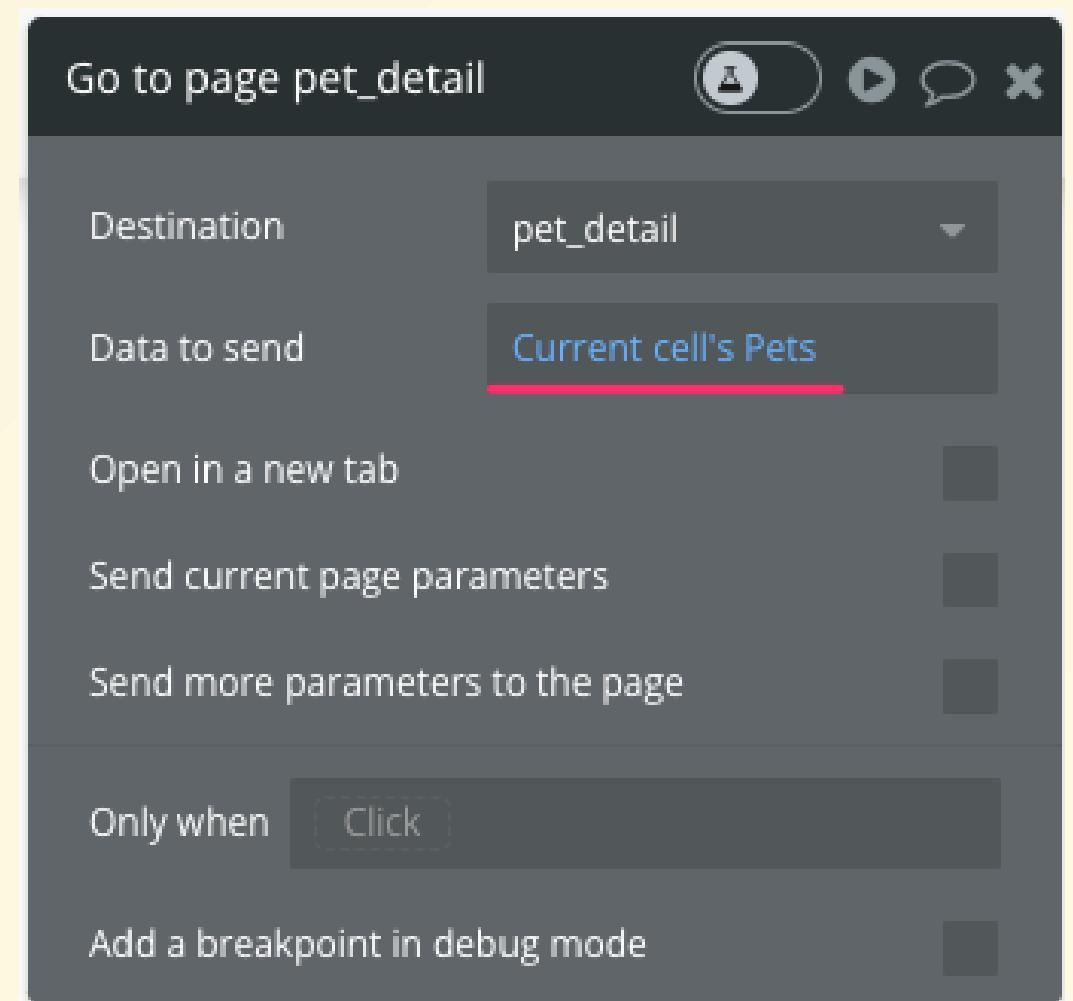
- Then you will see that the "When pet list image is clicked" box is in the state it was in before you set the behavior.



- 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 little 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

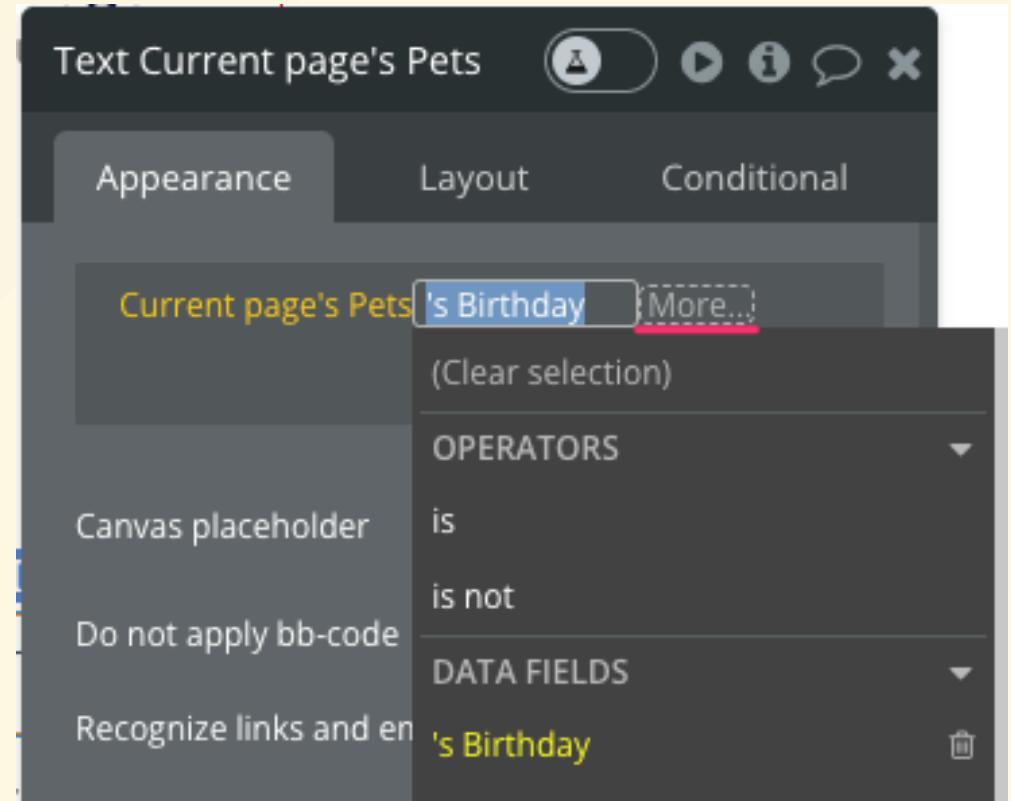
REGISTER

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

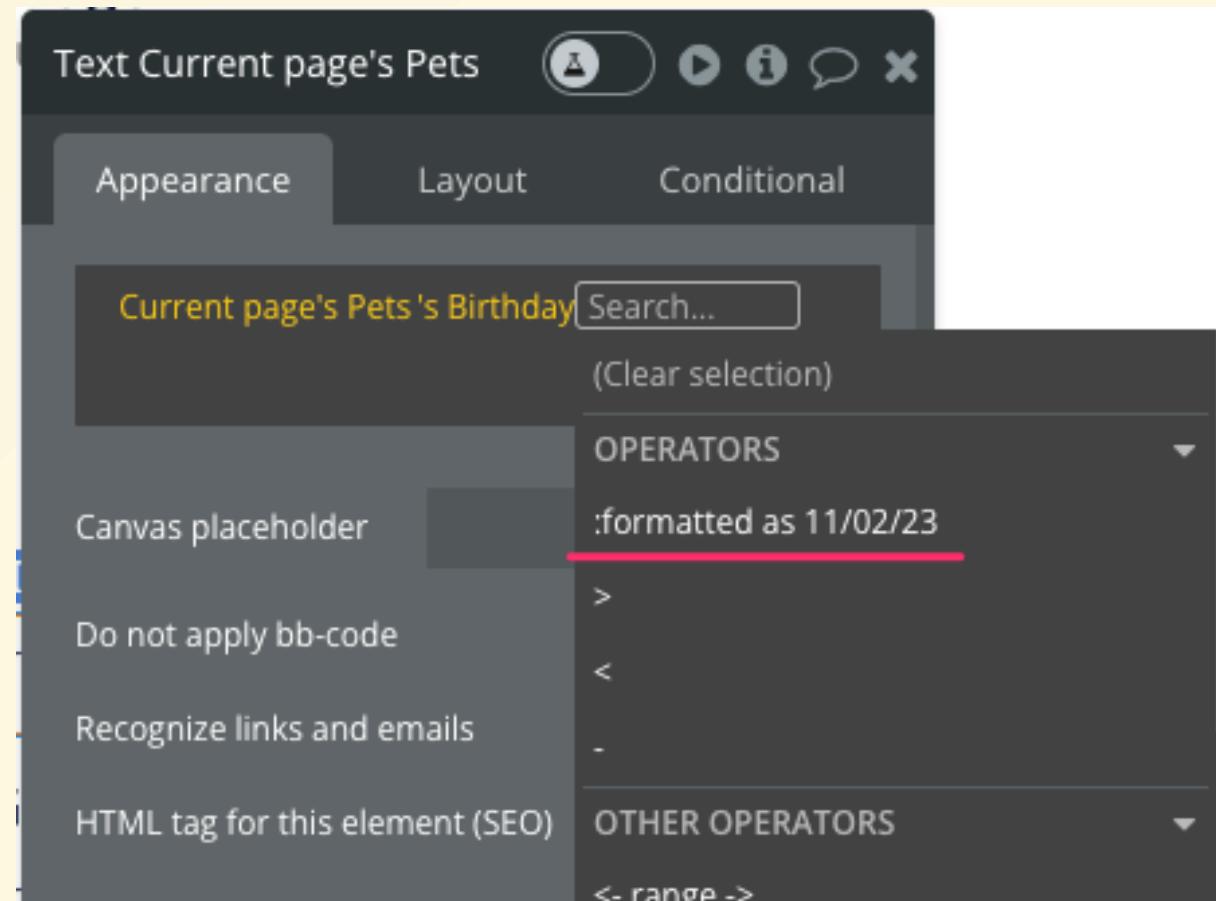
- Then click on the Birthday section of the

Current Page Pets's
Birthday

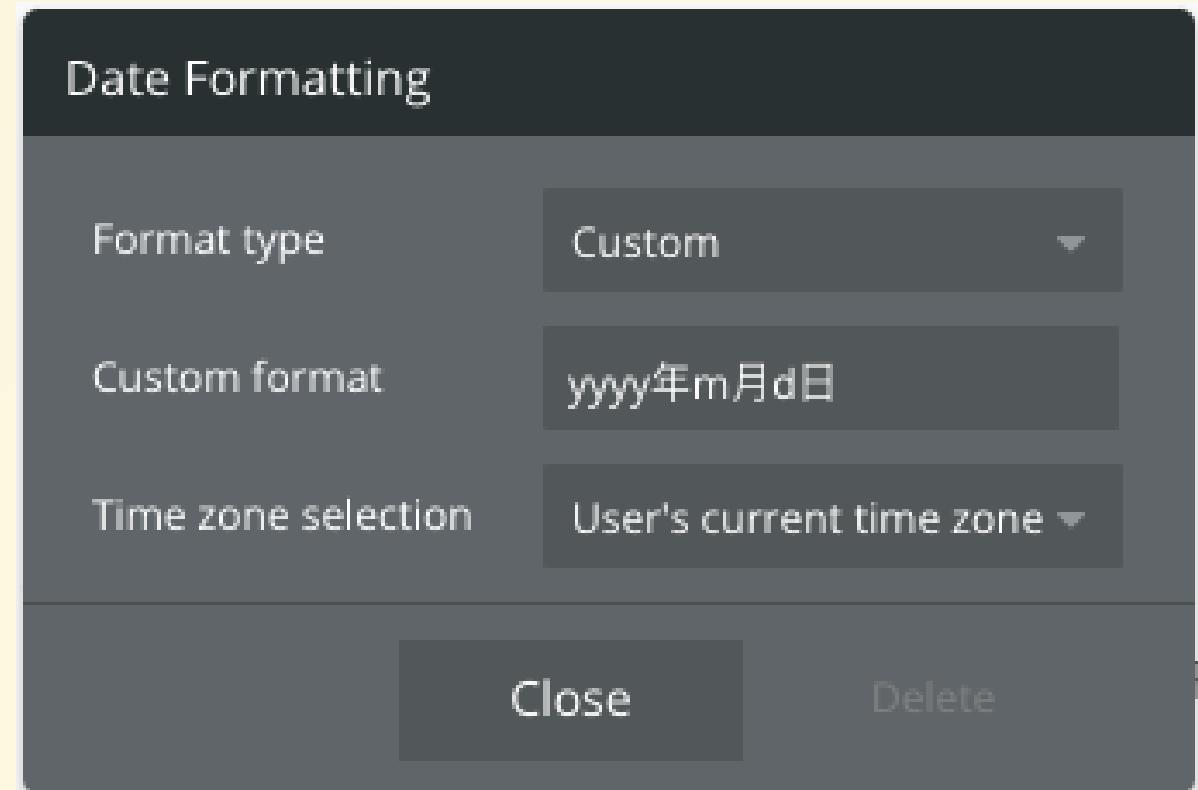
and you will see a More item at the back of the page.



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



- 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年m月d日 .



- 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 should be "Boy", and if the gender is "Female", the label should be "Girl"!

Name

メロ

Image



Birthday

2022年9月5日

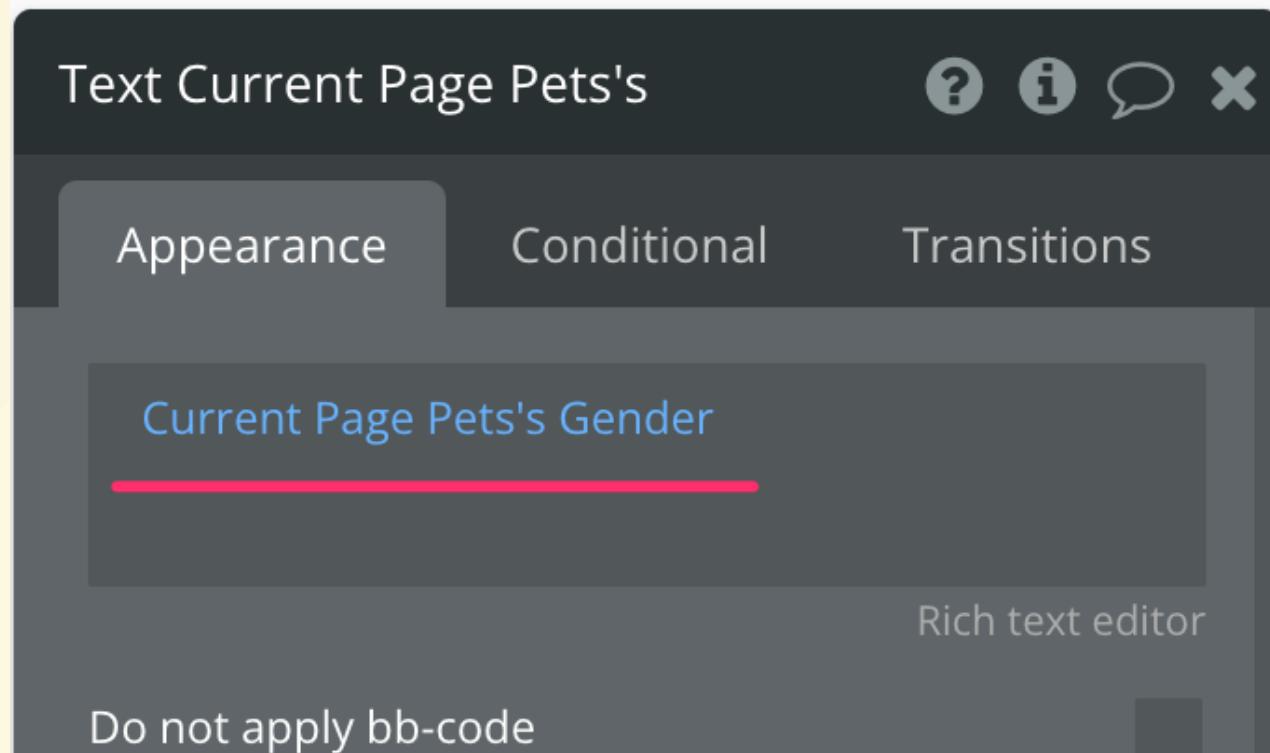
Gender

男の子

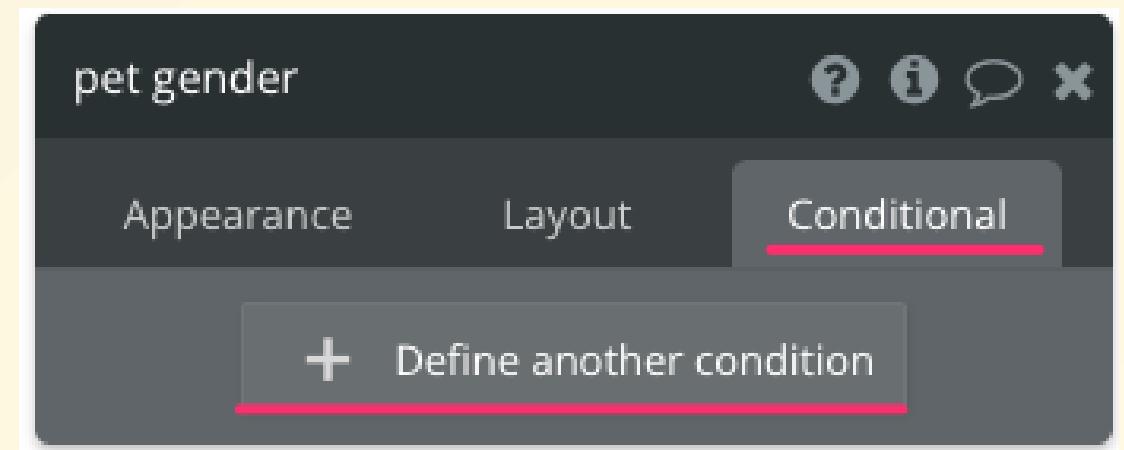
REGISTER

- 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

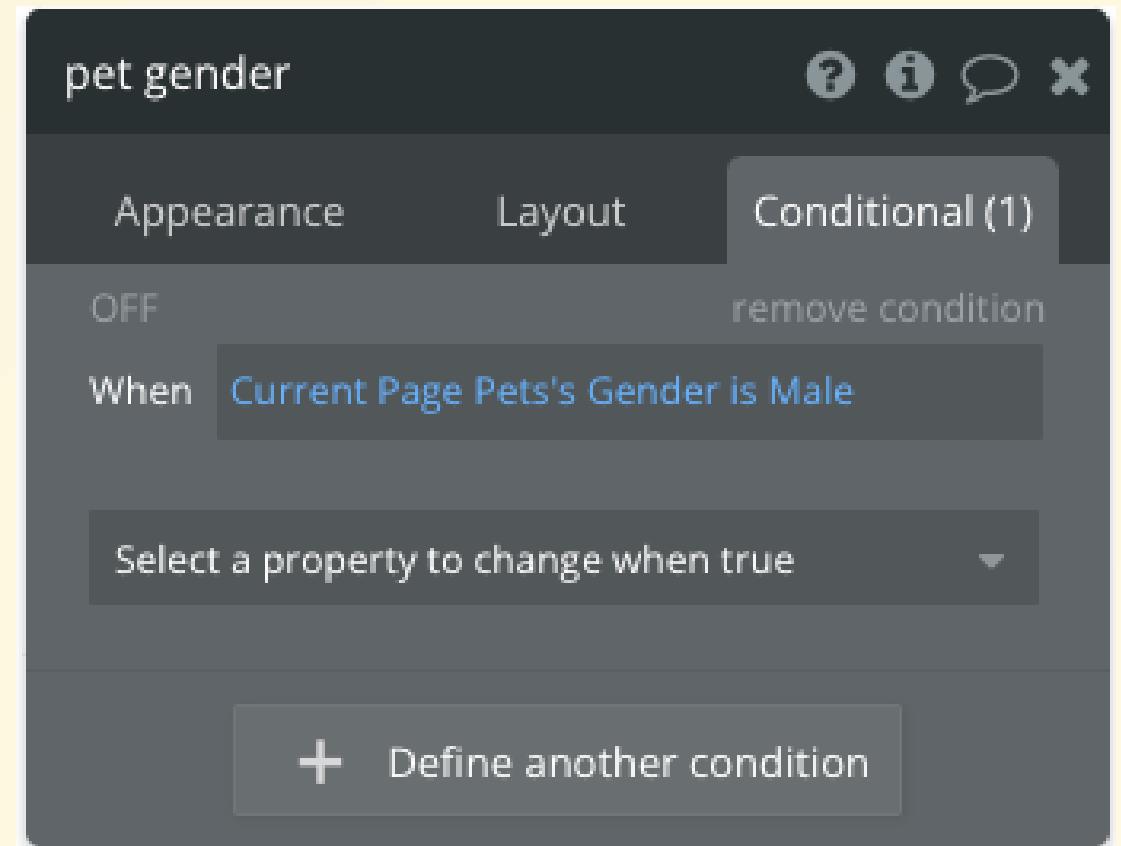
- First, move the cursor to the
Current Page Pets's Gender
field that you have just
entered on the Appearance
tab, and delete the entry
with the Delete button.



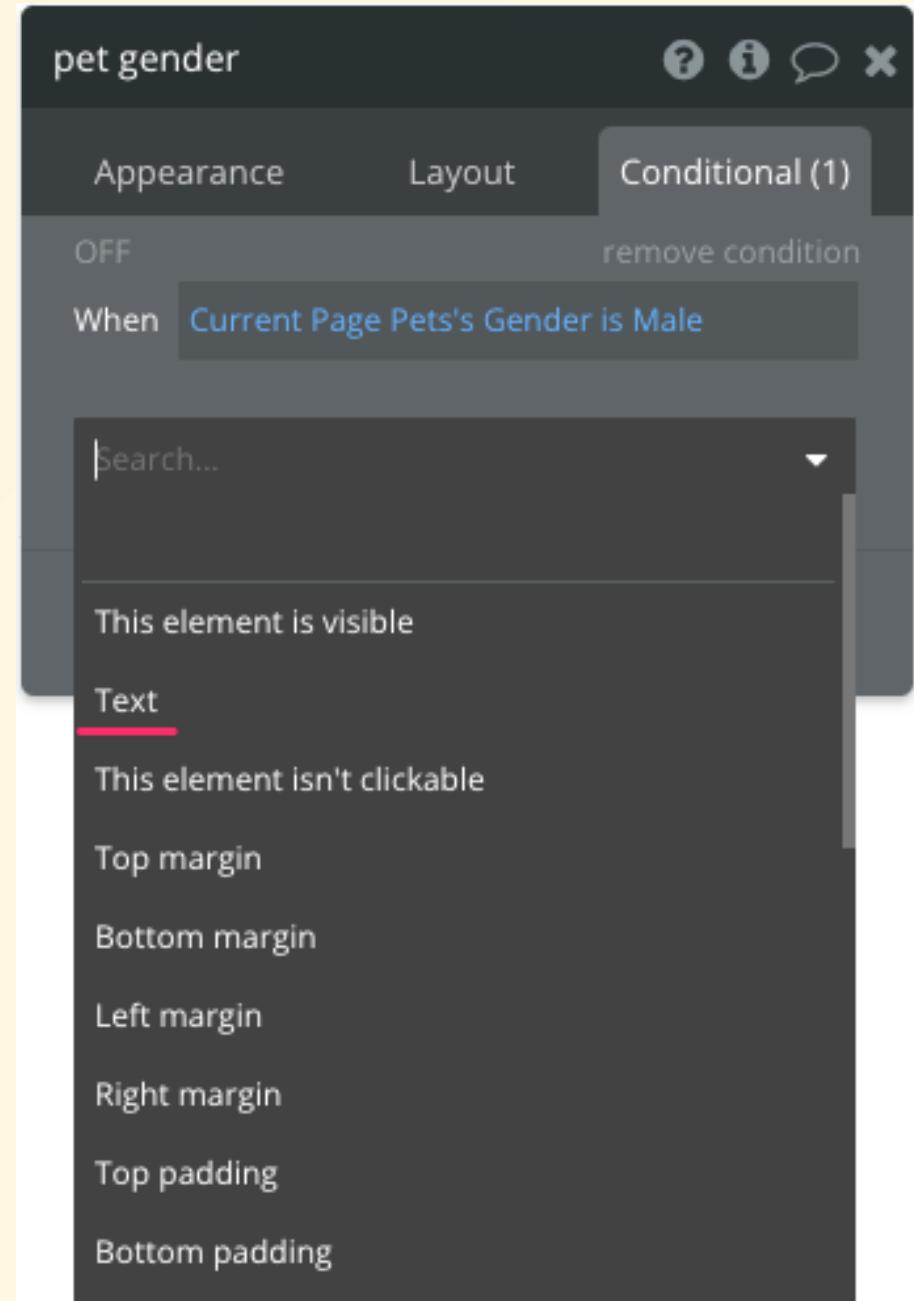
- Then open the Conditional tab and click the Define another condition button to add a condition.



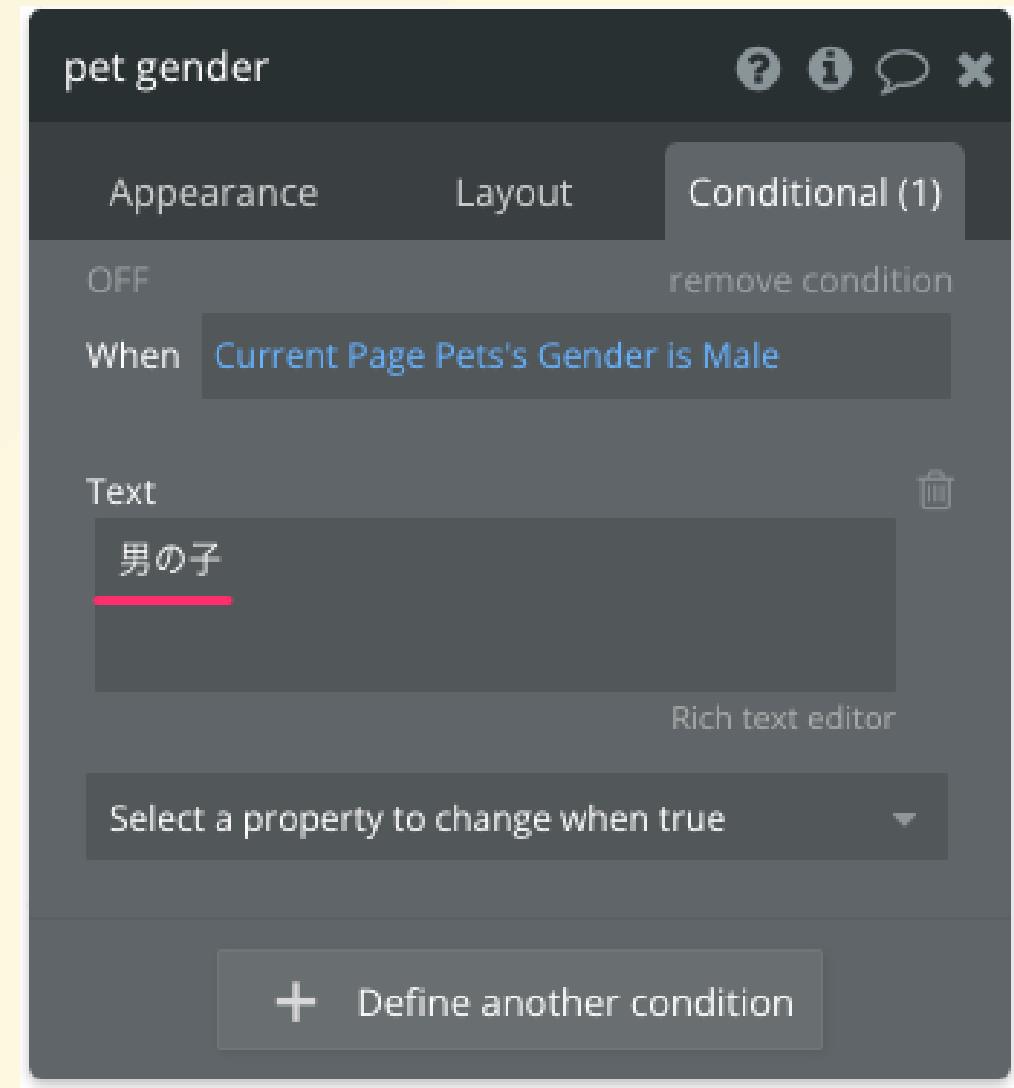
- First, specify the condition "If the currently displayed pet's gender is "Male"" in the When field.
- Only Male is manually entered, the others are selected from the pull-down menu.



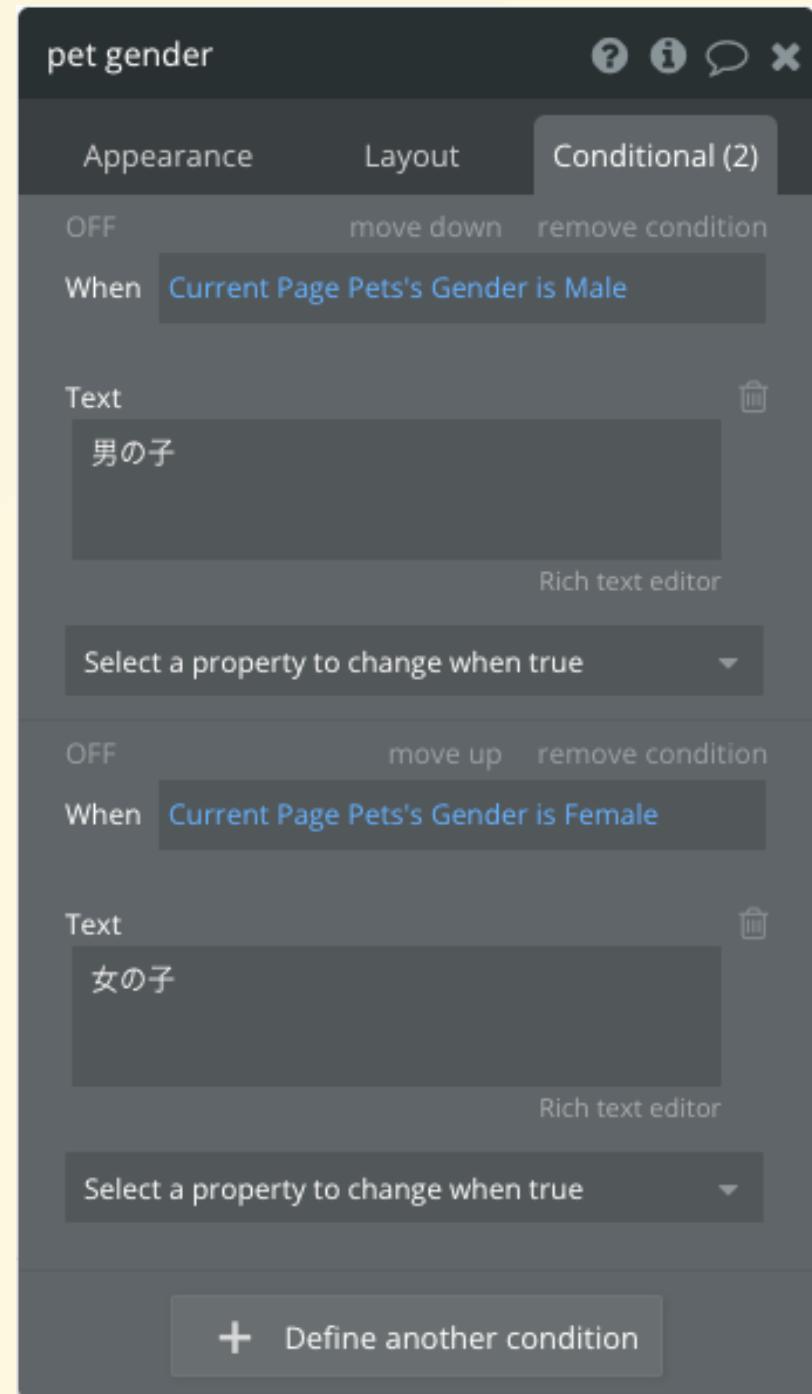
- Once the "When" condition is set, set the behavior when the condition is true
- In this case, we want to display a specific string, so we select "Text" from the "Select a property..." pull-down menu. In this case, we want to display a specific string, so select "Text" from the "Select a property..." 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 be displayed as "Boy"; if it is Female, it will be displayed as "Girl",
If the current pet's Gender is Female, it is displayed as a Girl.

Here we have a bit of a programming element!

- If you want to process the value of the element itself, as in Birthday, it is better to 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



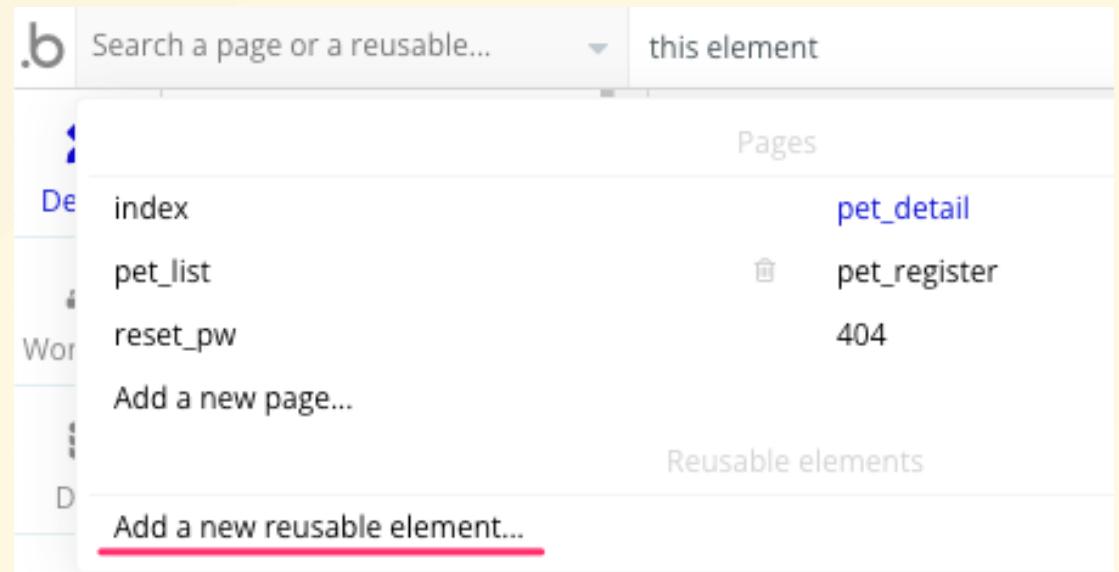
Why create common parts?

- 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.

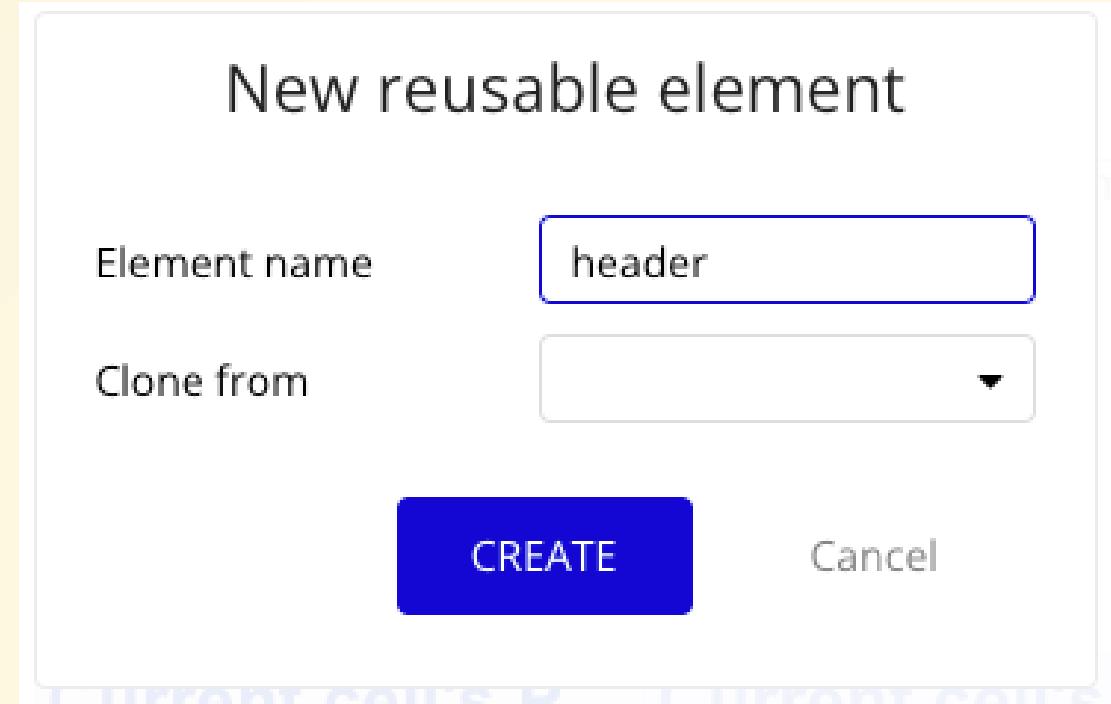
- Bubble solves such problems by using "Reusable elements" components!
- In this case, we will create a header component as "Reusable elements" and place it on the pet list, registration and details screens

Creating a header component

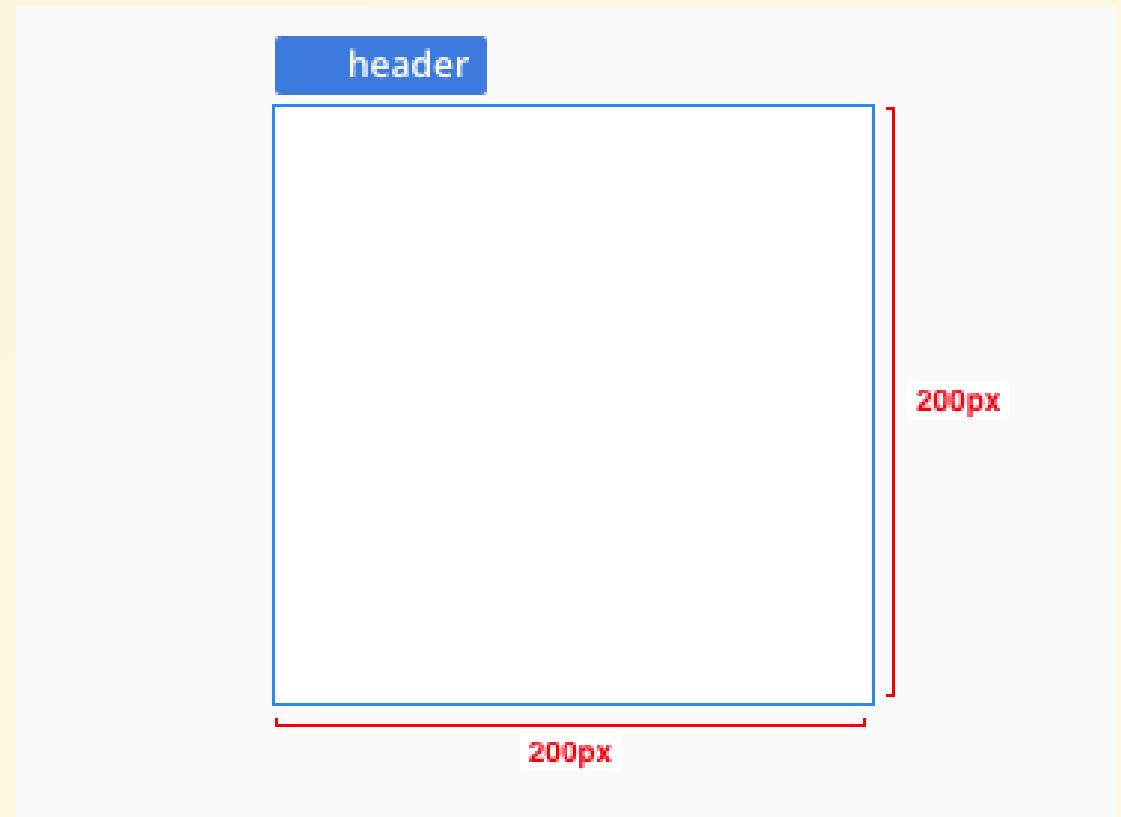
- Open the list of screens in the upper left corner and in it "Add a new reusable element..." Click on "Add a new reusable element..." in the list.



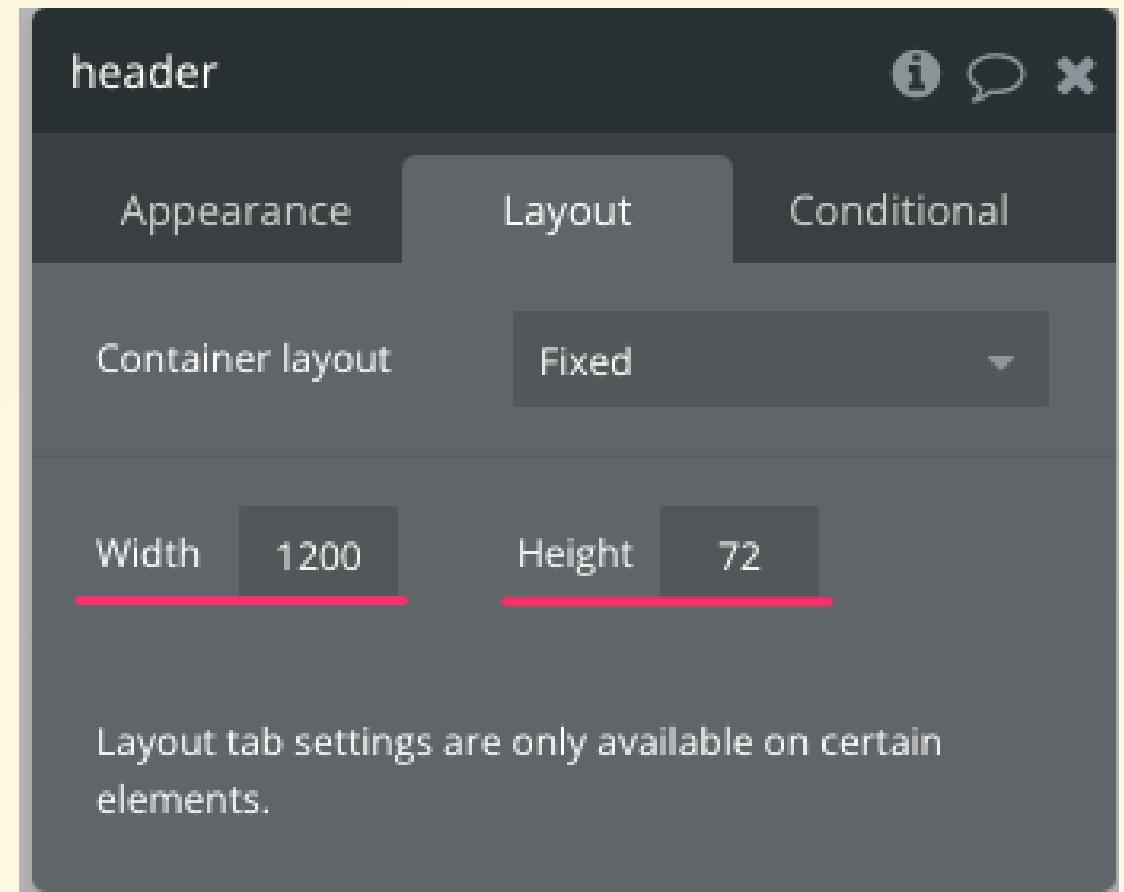
- Then, the same popup as when creating the screen is displayed. This time, enter "header" as the name of the common part and click CREATE.



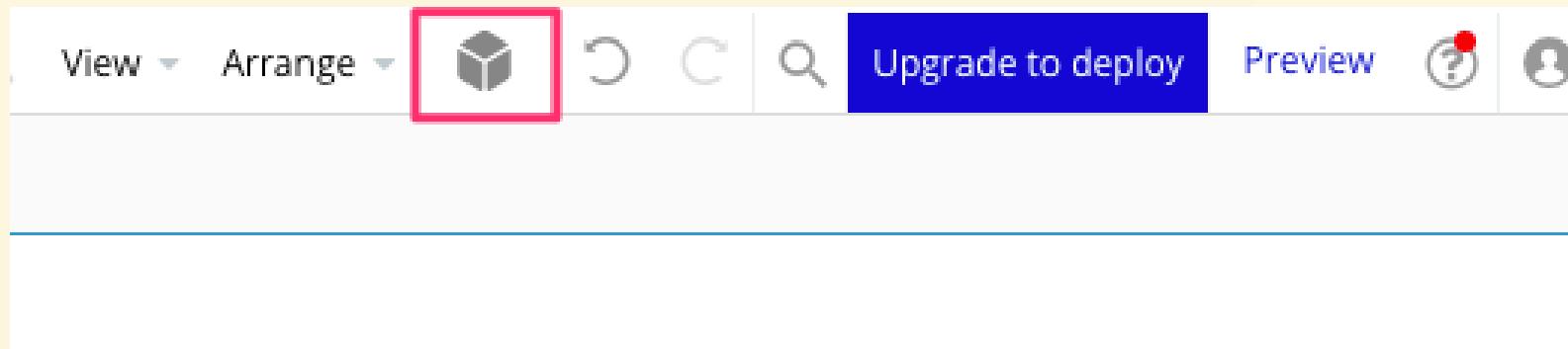
- Then an area of 200px in width and height will appear in the right panel.
- The width of the area is a little narrow as it is, so first adjust the area.



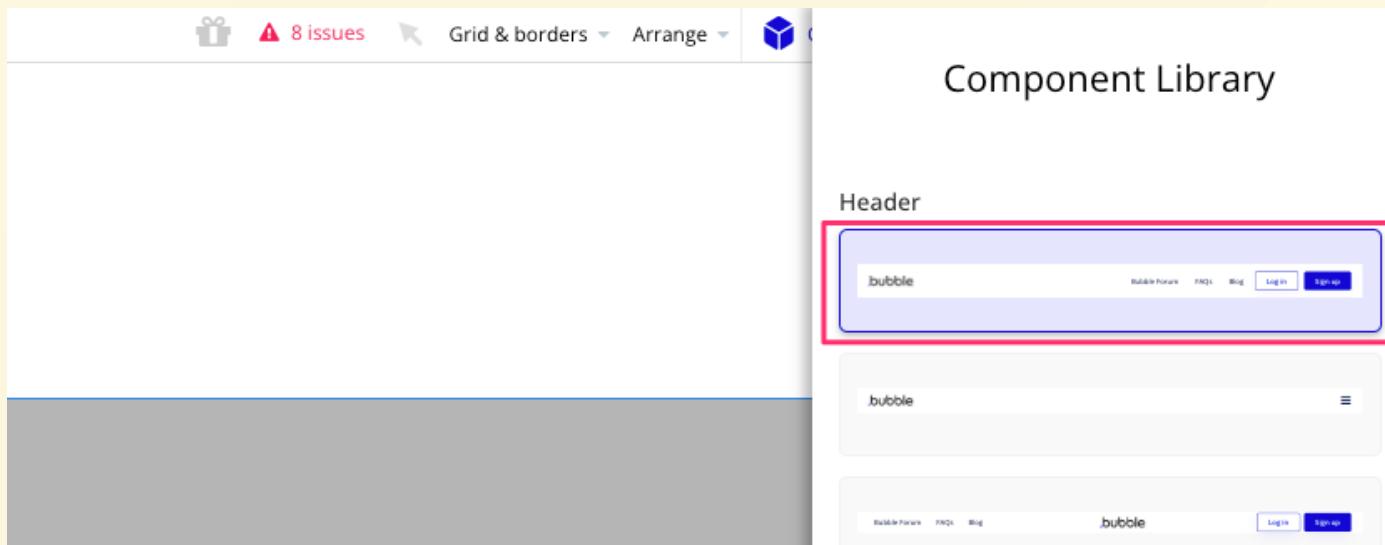
- Right-click in the right panel area --> Edit
- Select the "Layout" tab in the familiar pop-up window.
- Change **Width** to "1200" and **Height** to "72"



- Now we will create the common header components, but here we will use a useful feature
- Let's use the header components from "Components", where we first used the signup/login components



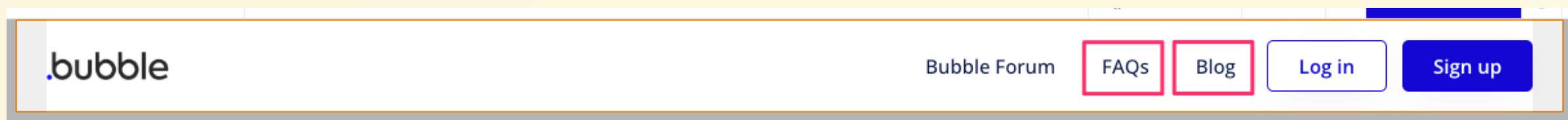
- In this case, select the topmost component from the "Header" components and drag it to the right panel.
- Once placed, click the Close button under the Component Library to close it.



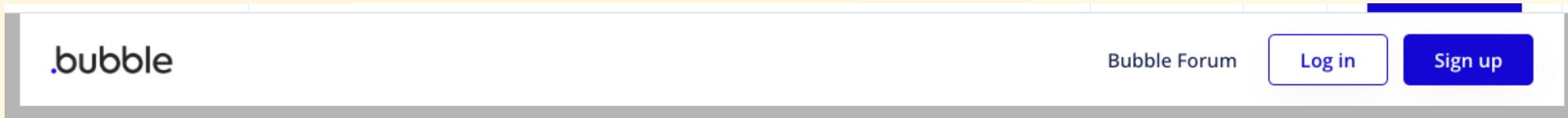
- Then, the sample Header that Bubble provides is displayed on the right panel.
- This time, we will customize it.



- 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

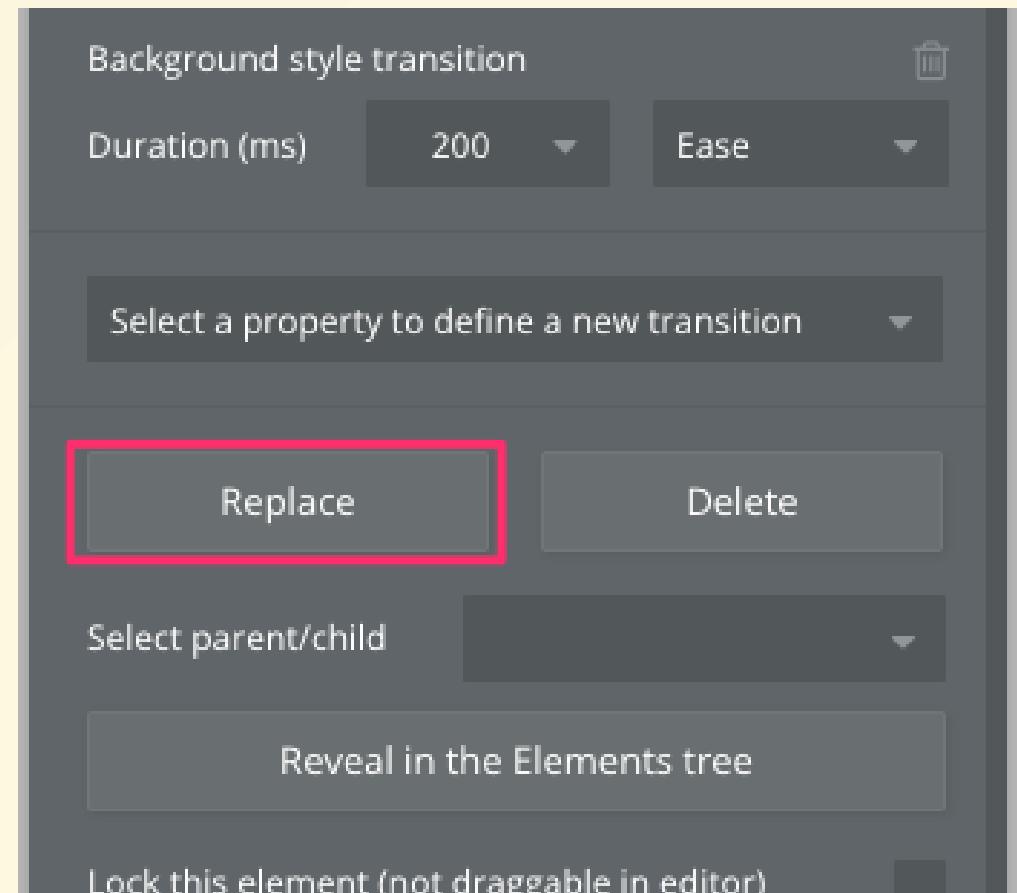


Setting up a link to the pet registration page

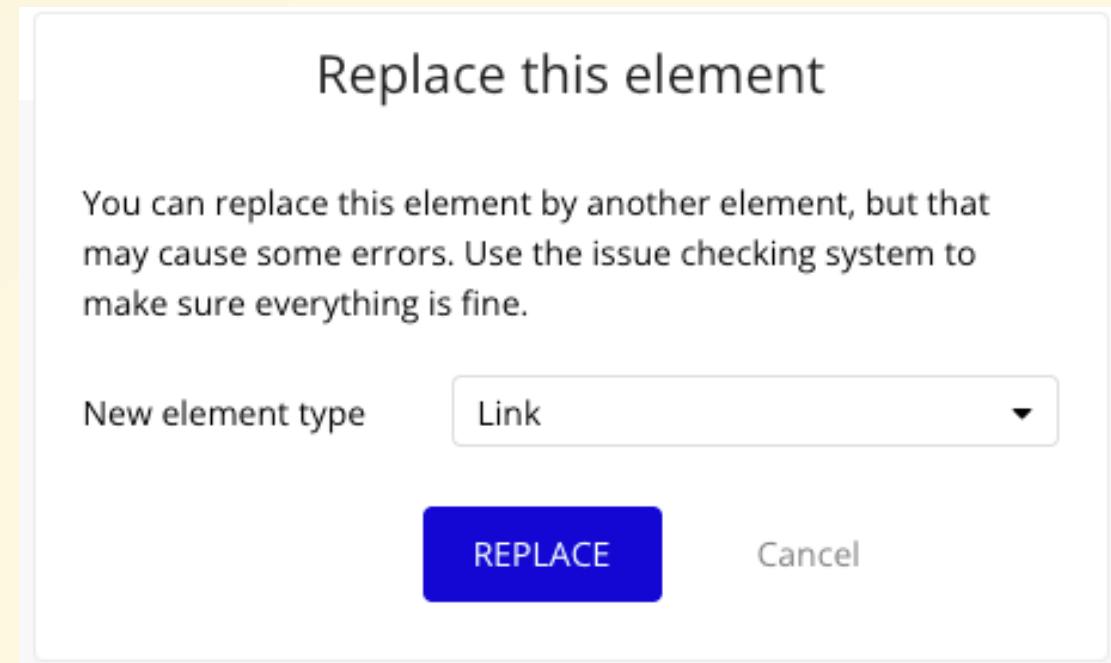
- First, let's set up a link to the pet registration page.



- 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

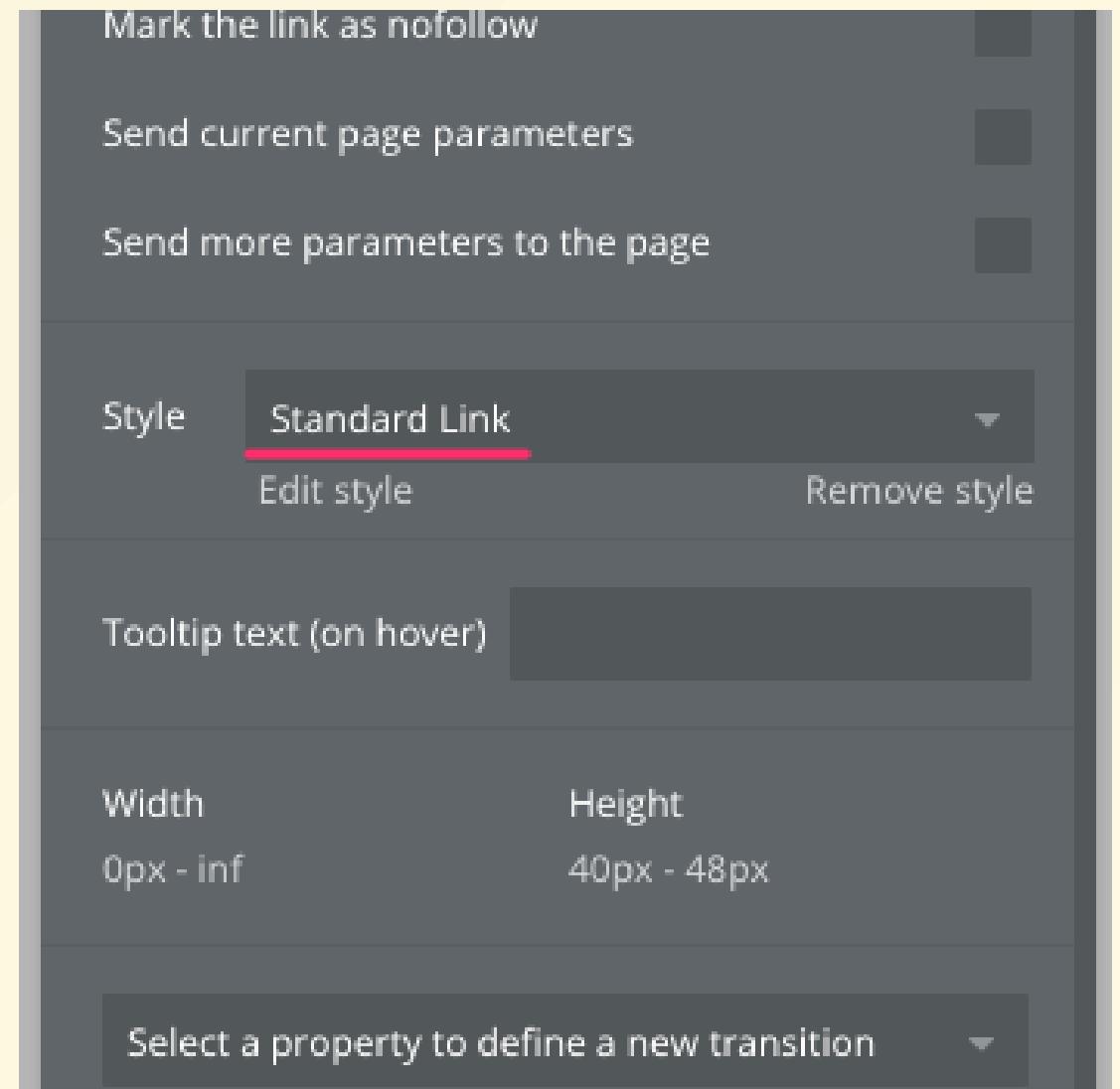


- This function changes the type of the currently selected element.
- In this case, we will replace the Text element with a Link element.
- Select "Link" in the New element type and click REPLACE

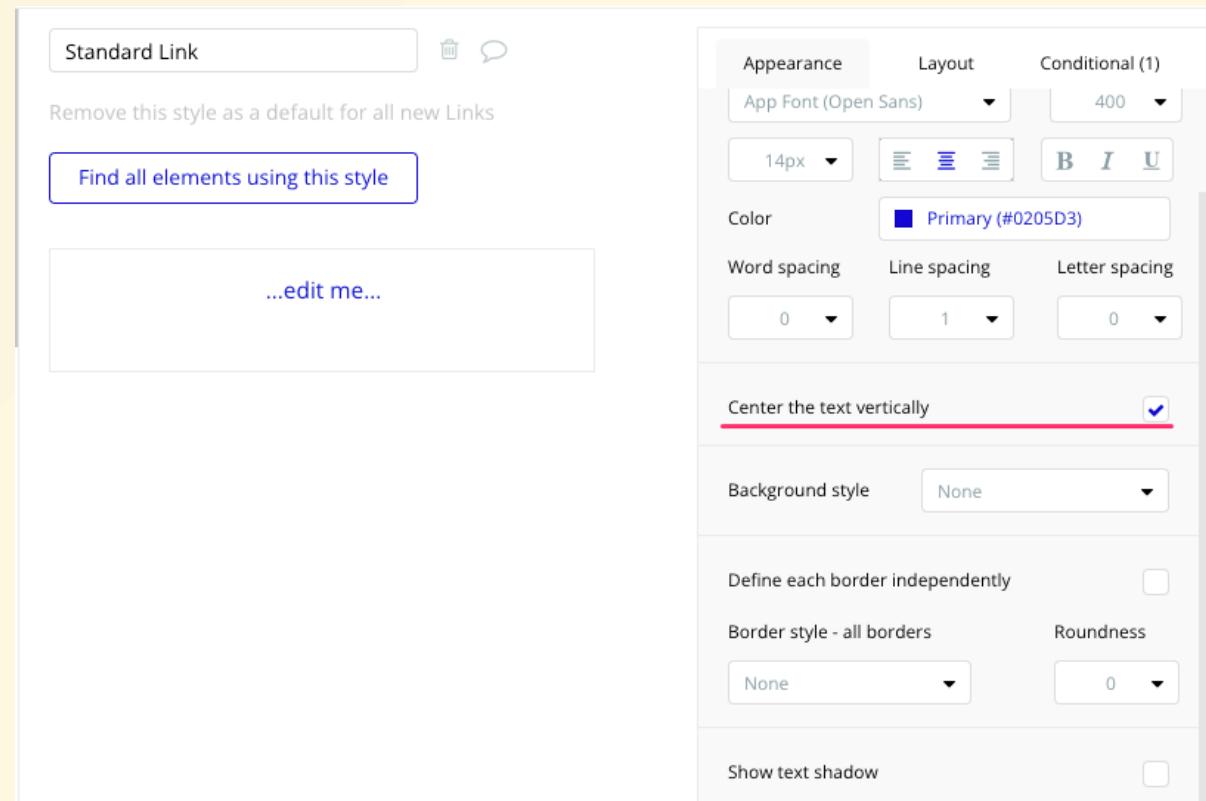


- The element type is now replaced by Link.
- However, the `Style` and `Conditional` elements that were set on the original element are still there, so we will change them.

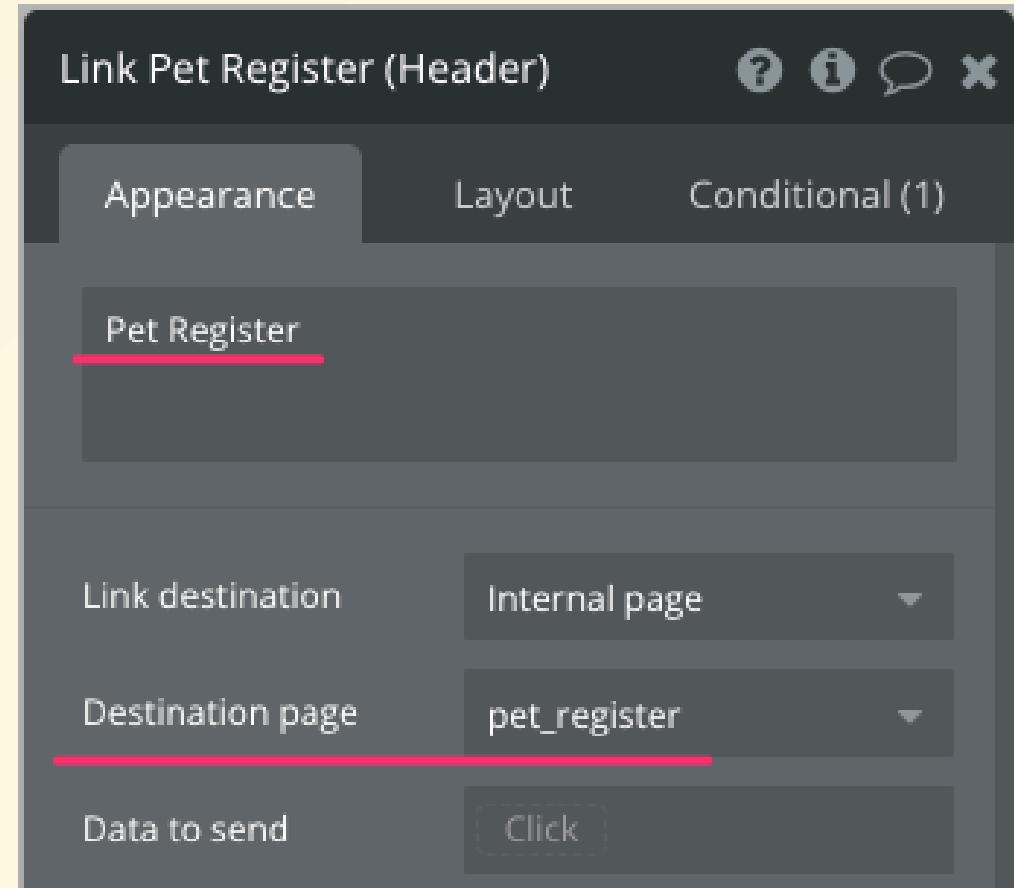
- First, change the **Style** to **Link**
- Select "Standard Link" for **Style** from the element's edit popup.
- If you leave it as it is, the text in the element is top-aligned, so center it vertically.



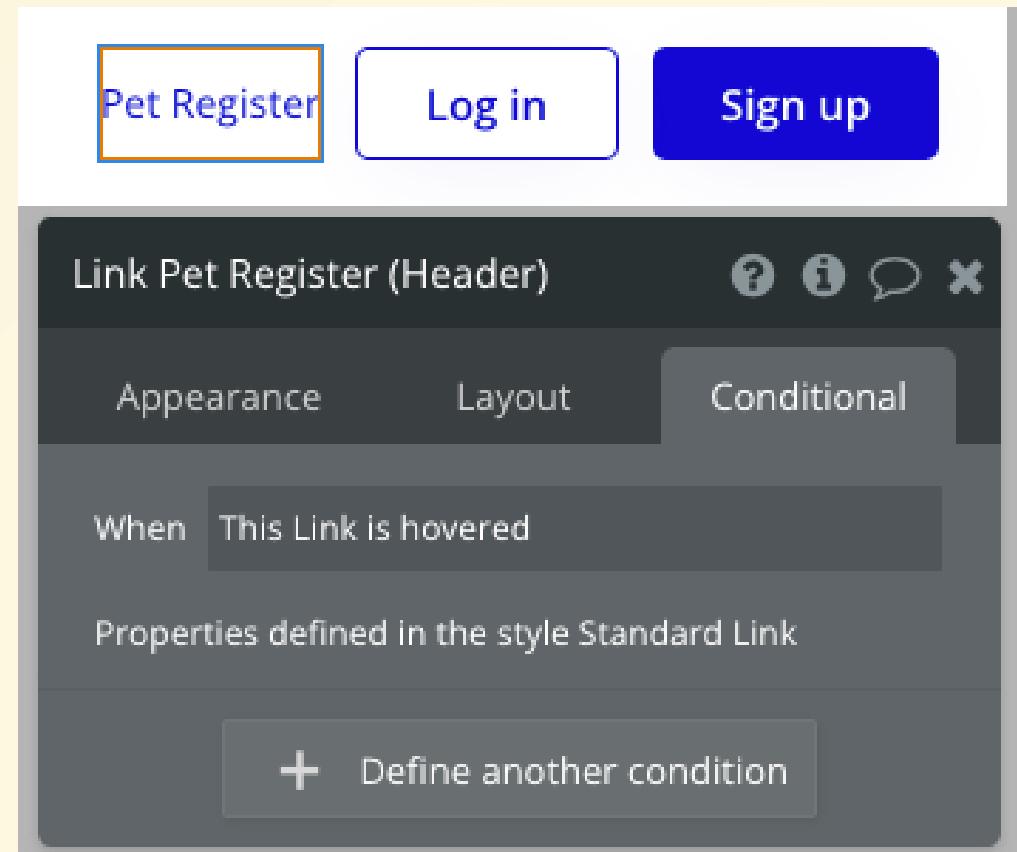
- Under "Standard Link" click on "Edit style".
- This will take you to the Styles tab, where you can edit the style of the Standard Link.
 - Now the text of the link will be vertically centered.



- 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.



- Style has been changed and the **Conditional** content has been removed
- If it is as shown in the capture, it is OK.



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



Creating Signup / Login

- Now that the header is ready, we will create the user information signup and login that will be required in the future.



.b

Welcome to
Bubble

Name

Email

Password

Re-enter Password

[Sign Up](#)

[Already have an account? Log In](#)

A screenshot of a Bubble application's sign-up page. On the left, there's a sidebar with the logo ".b" and the text "Welcome to Bubble". The main area contains five input fields: Name (with "John Doe" typed in), Email (with "johndoe@gmail.com" typed in), Password (with "*****" as placeholder), Re-enter Password (with "*****" as placeholder), and a blue "Sign Up" button. At the bottom, there's a link "Already have an account? Log In". To the right of the form is a large, fluffy, light brown rabbit sitting on a light-colored surface.

- Here are two things to do
 1. prepare a box to register users
 2. set up signup and login on the first index page
- First, we will prepare the box for user registration.

1. prepare a box for registering users

- In fact, the user box is already prepared from the beginning.
- Select User in the Data types of Data.
- There is already an email field, but there is no user name field, so add a type of text .

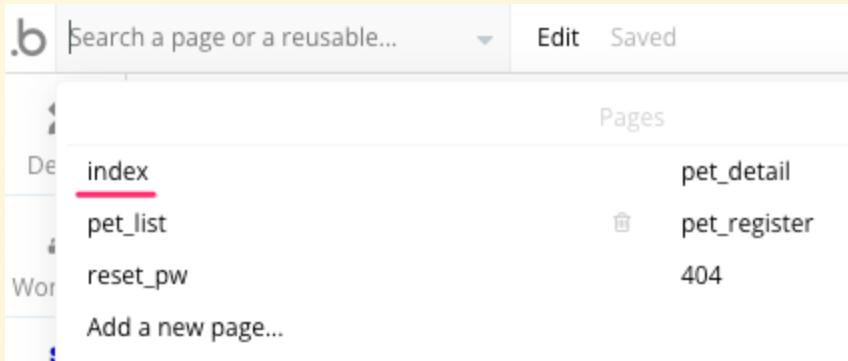
Create a new field

| | |
|--|-------------------------------------|
| Field name | <input type="text" value="Name"/> |
| Field type | <input type="text" value="text"/> ▾ |
| This field is a list (multiple entries) <input type="checkbox"/> | |
| CREATE Cancel | |

- Incidentally, the password field is missing, but it actually exists!
- This is only because Bubble doesn't show it to the public for data protection reasons, but behind the scenes, there is a password field, where the password entered by the user is encrypted and stored.

2. Set up signups and logins on the first index page

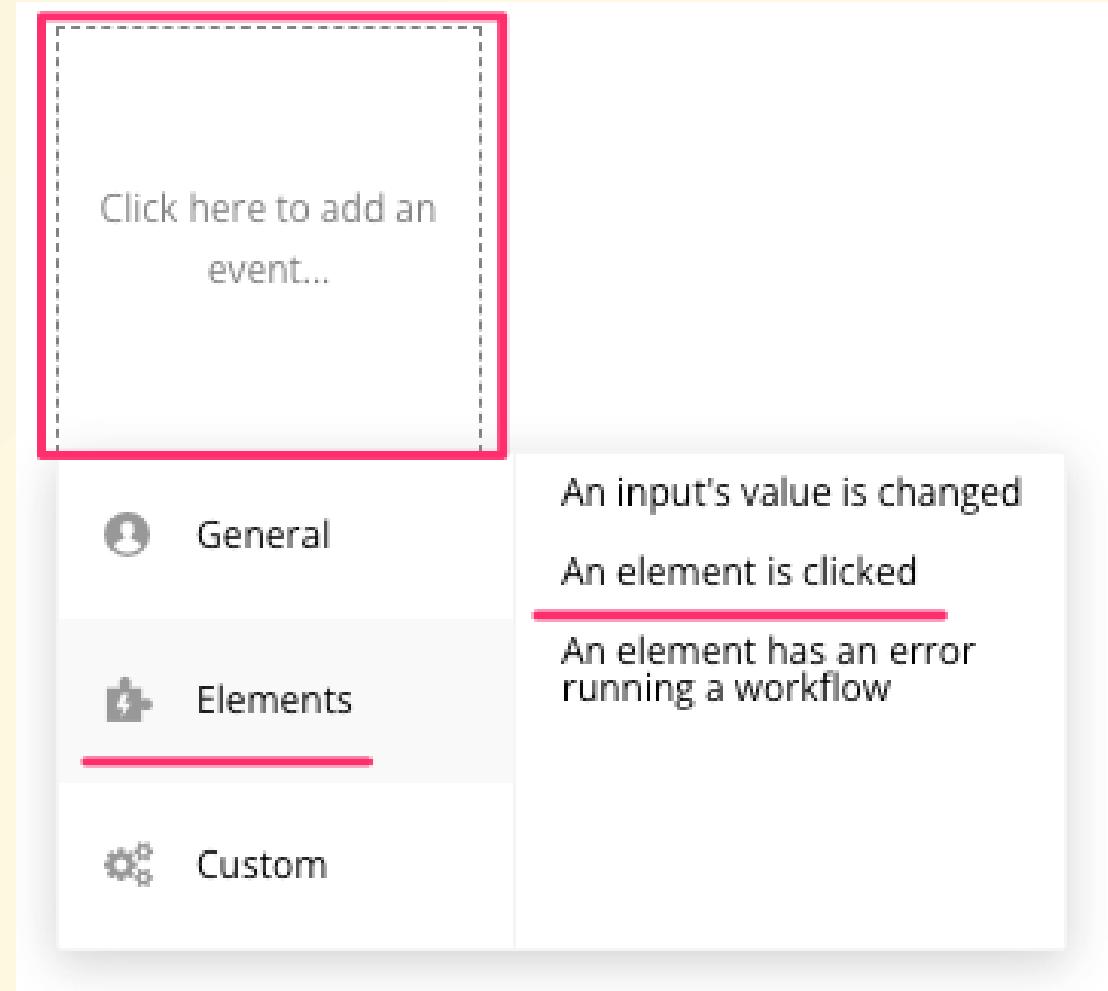
- Now that the user boxes are ready, it is time to set up the workflow on the index page
- Select "index" from the list of screens in the upper left corner



- This time we will set it up from the Workflow tab

First, sign up (user registration) to set up

- Click here to add an event...
- click
- Select **Elements** -->
An element is clicked
since the workflow will be
the same as when you click
the **Sign up / Sign in**
button already on the index
page.

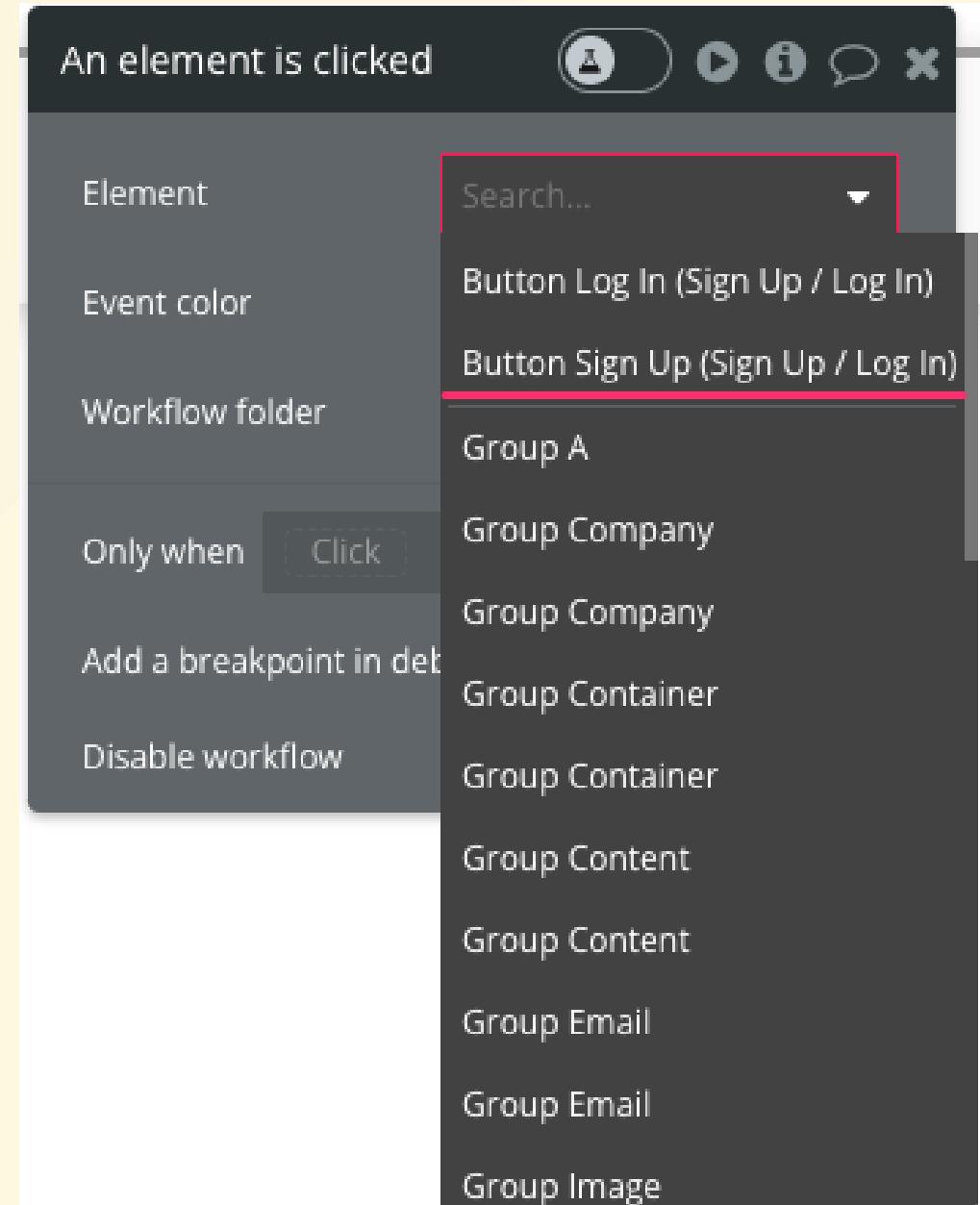


- A detail popup will appear, first select the element

Button Sign Up (Sign Up / Log In)

.

- This means the workflow when the Sign Up button is pressed.



- Next, we will set up the contents of the workflow. I am sure you can do it, right?
- Please try to set it up based on what you have learned so far.
- Note that user operations are special, so please select the appropriate menu from **Account**, not **Data (Things)**

Click here to add an action...

Recommended next action

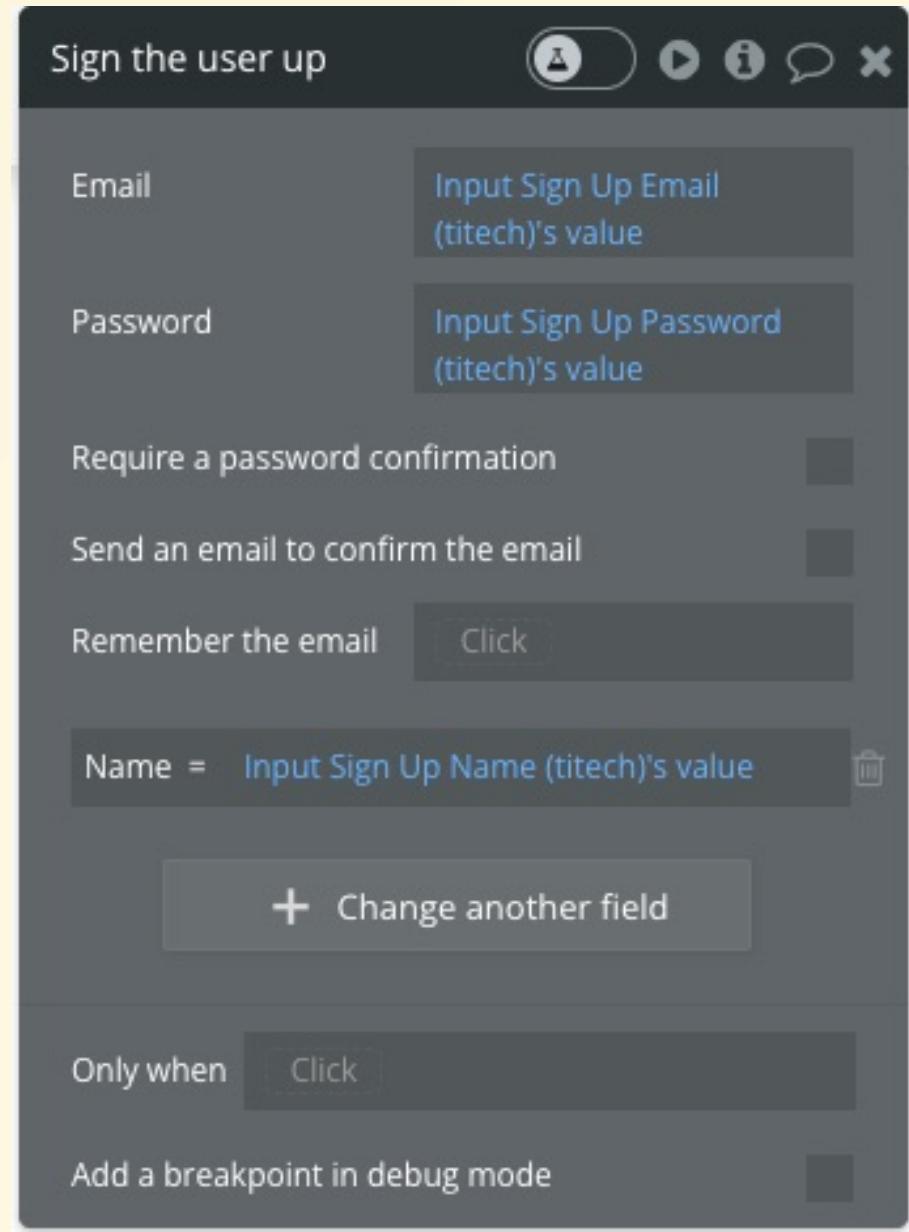
Send email

Search for an action...

| Action Category | Action Description |
|-----------------|--|
| Account | Sign the user up Log the user in Signup/login with a social network Log the user out Update the user's credentials Make changes to current user Send confirmation email Send password reset email 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 |
| Navigation | |
| Data (Things) | |
| Email | |
| Payment | |
| Analytics | |
| Element Actions | |
| Plugins | |
| Custom Events | |

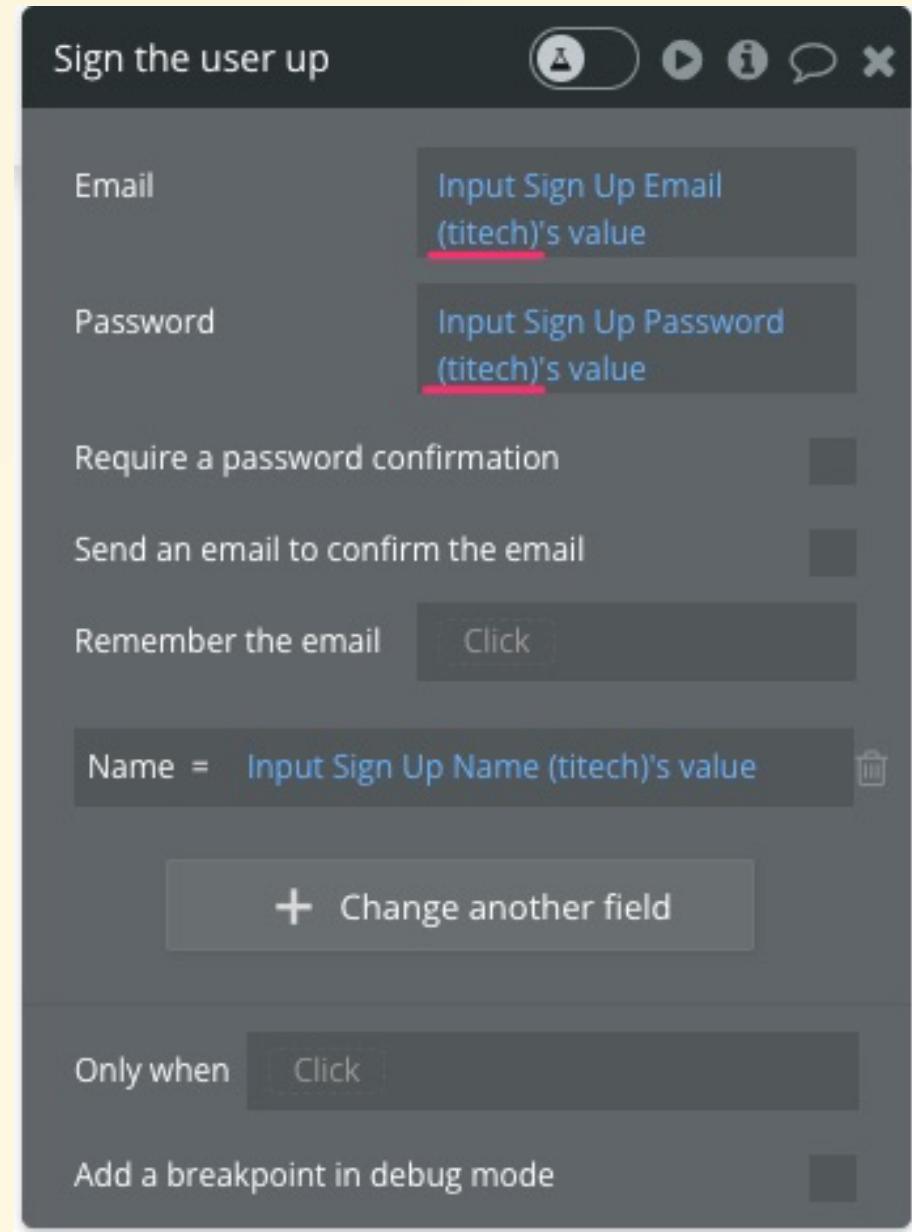


- The menu is literally
Sign the user up!
- And here's what you need
to set up
 - Email: Input Sign Up Email (titech)'s value
 - Password: Input Sign Up Password (titech)'s value
 - Name: Input Sign Up Name (titech)'s value



- Since there are many Input elements, it was difficult to find the element you were looking for, so we added (titech) to the end of the element name.

This makes it convenient to filter by titech when setting up 



- Let's preview it in action
- Fill in the required information on the Sign up screen and press **Sign Up**.
- Then, select **All Users** from the App data in the Data tab, and if the user information you just entered is registered, you are good to go. 

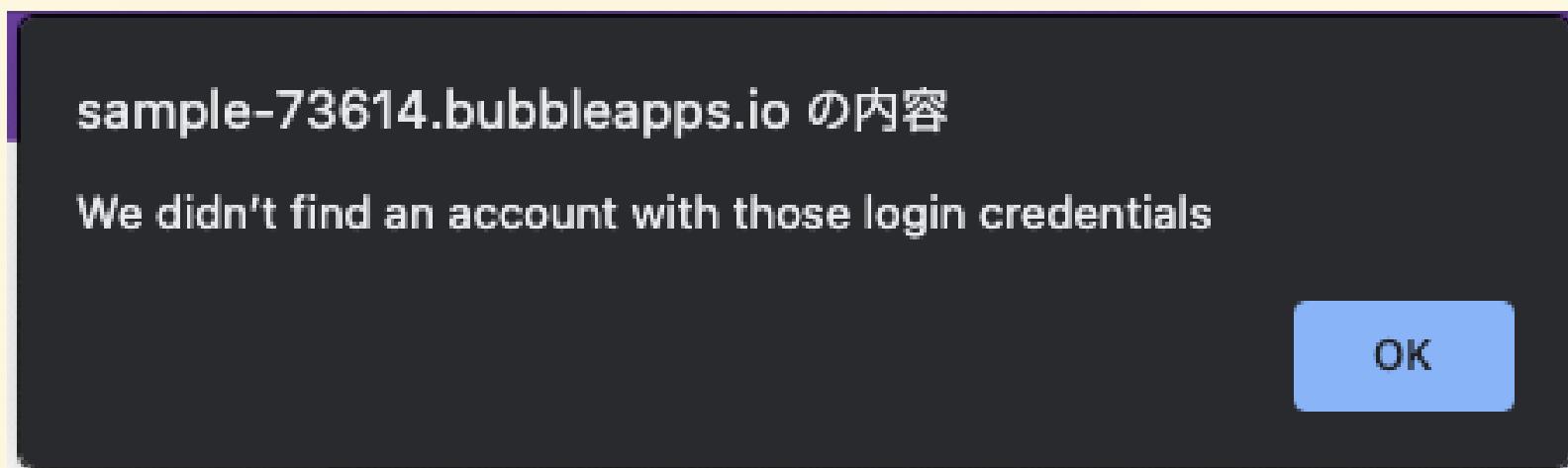
Then let's set up your login!

- The concept is the same as in Sign Up earlier, so let's do it!



- The action to choose is `Account --> Log the user in.`.
- Then select the email address and password to use when logging in.
 - Again, since there are so many elements, I added `(titech)` to the end of the target element to identify it.

- Let's preview it.
- Enter the user's email address and password that you entered when signing up earlier, and confirm that Log In is enabled.
- Press the Log In button, and if the following dialog does not appear, the login was successful!



- Now that we have the user registration and login out of the way, we can proceed to 

Control Login / Logout buttons based on login status

- Next, we will prepare the login and logout buttons
- Select header under Reusable elements from the top left corner

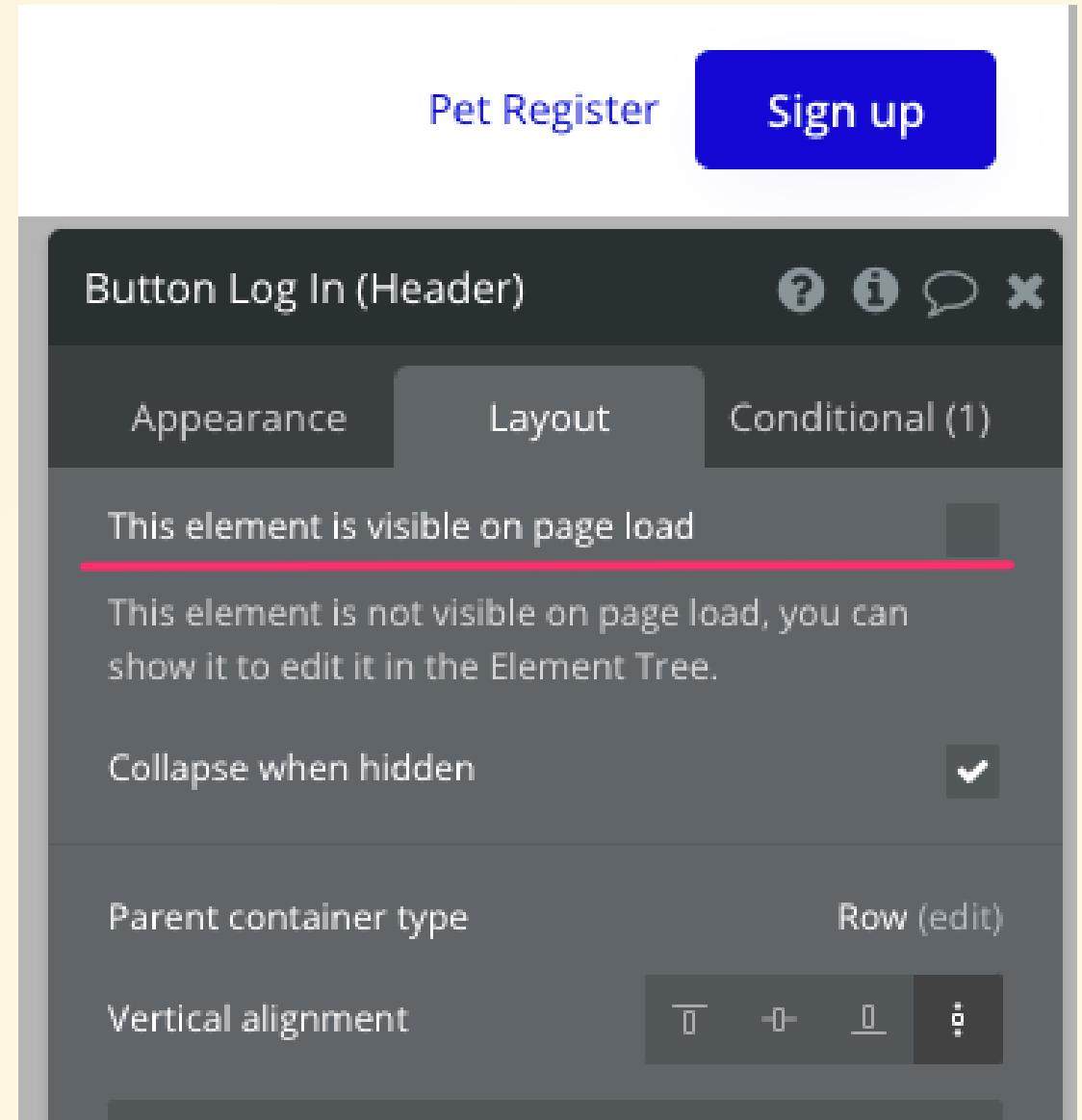


- There are four things to do here
 1. display the login button only when the user is "not logged in"
 2. make the login button go to the login screen (index) when it is pressed
 3. change the signup button to a logout button and display it only when the user is already logged in
 4. When the logout 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.

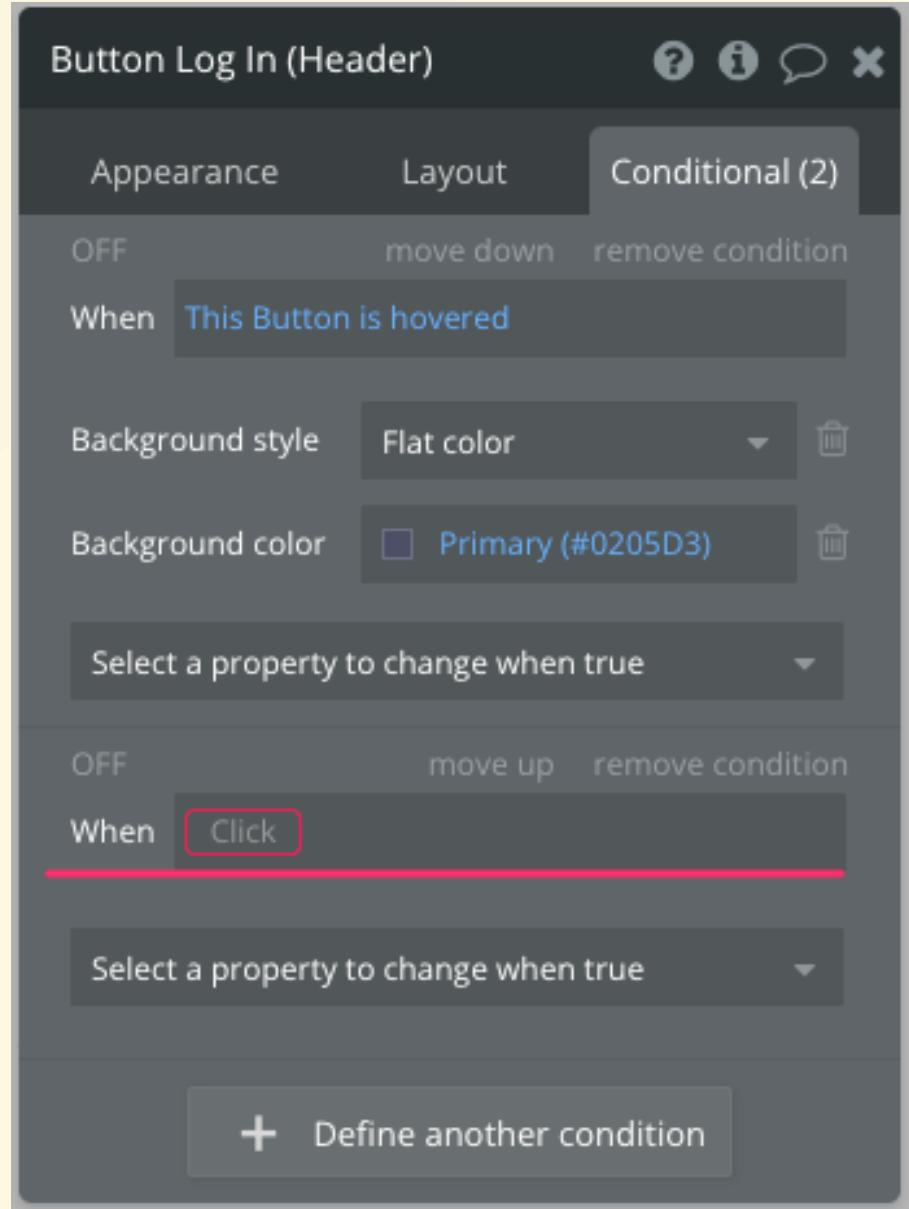
1. The login button is "displayed only when the user is not logged in."

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

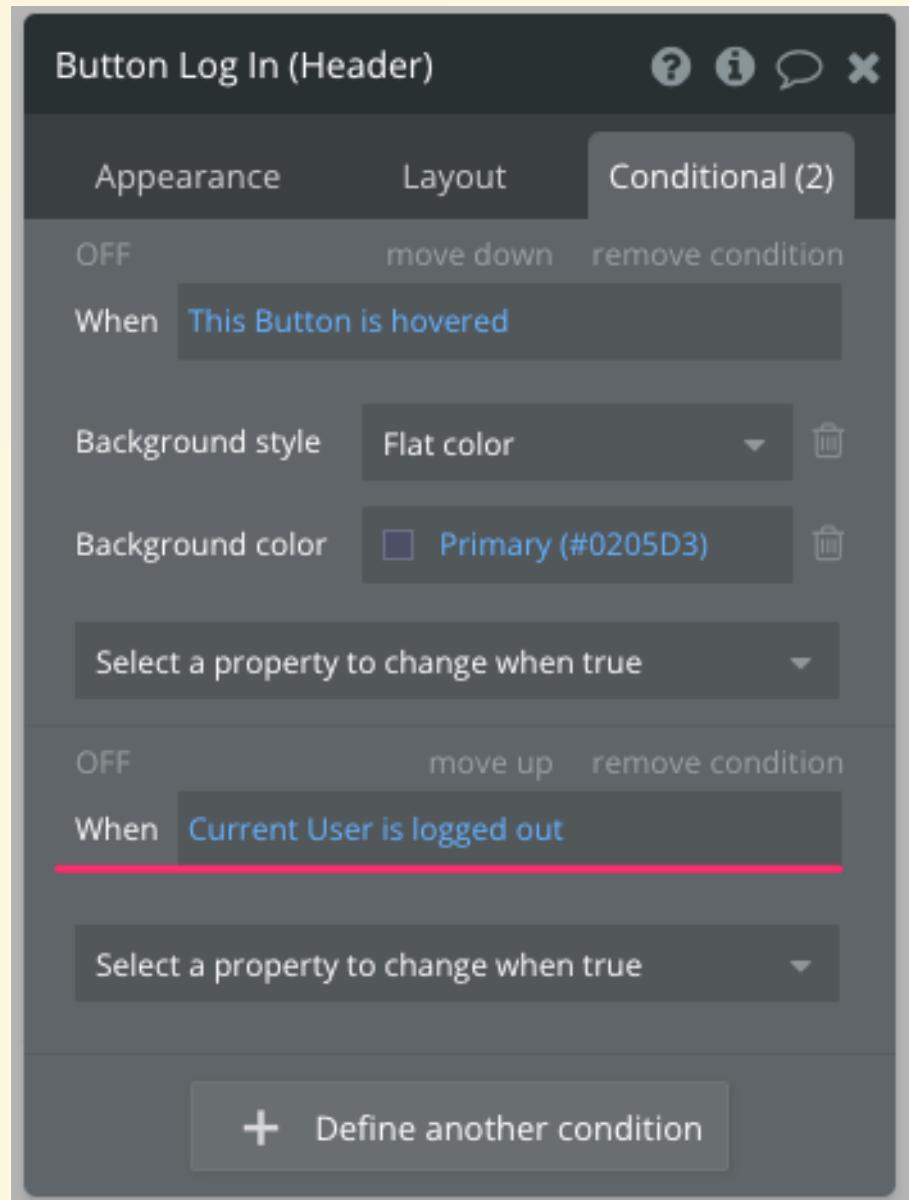
- First, display the Edit Login Button dialog
- On the Layout tab, uncheck `This element is visible on page load
- This will make this element invisible on page load.



- 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".



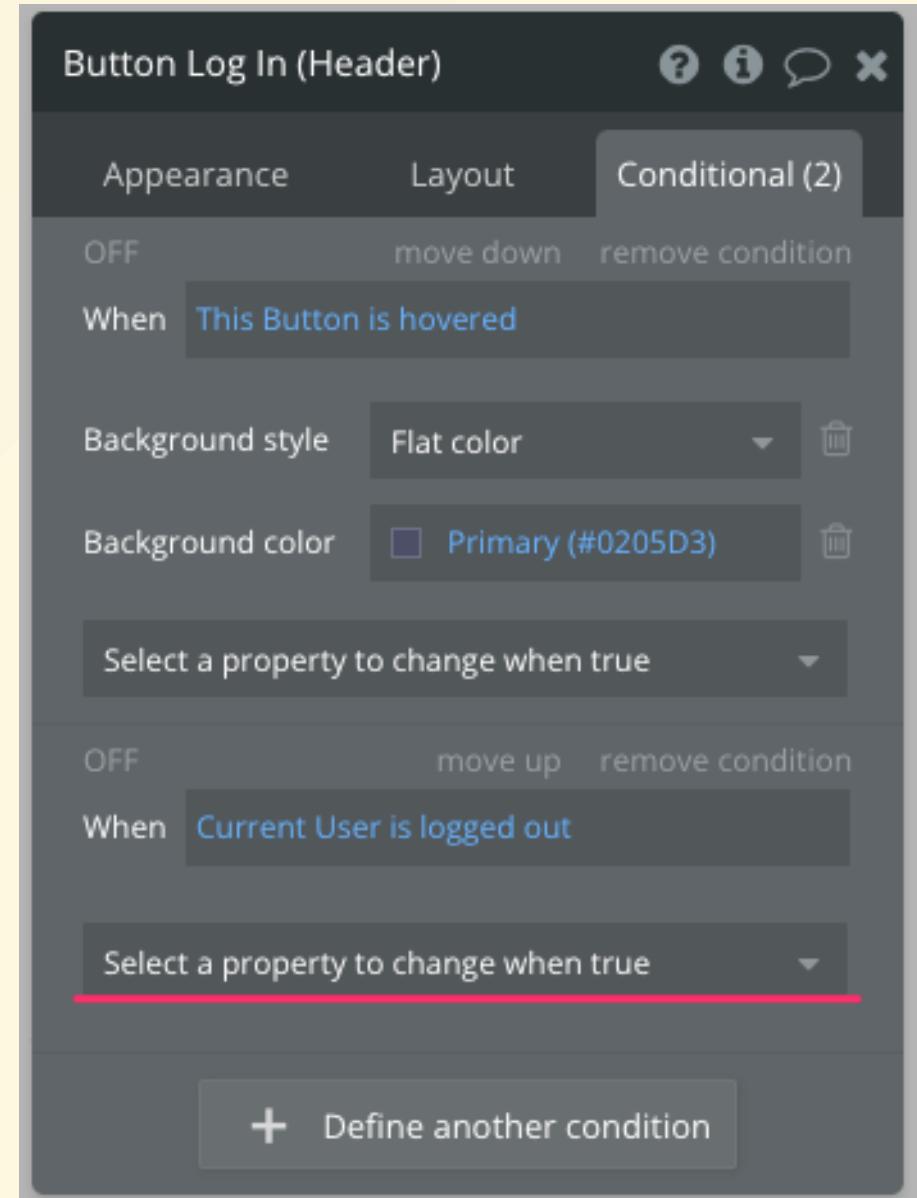
- It looks like this.
 - "Current User is logged out"



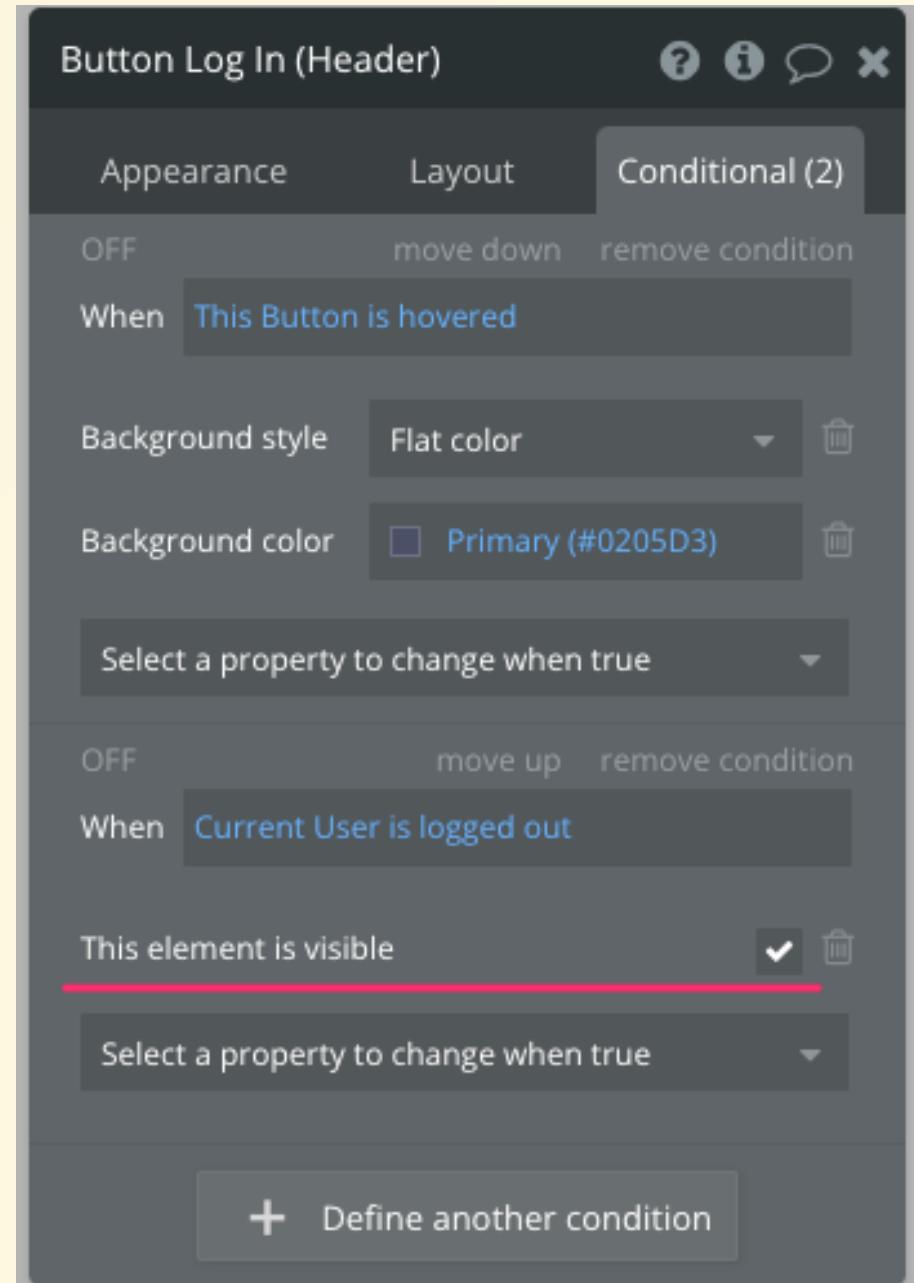
- Then, select the process to be performed when the When condition is met with Select a property to change when true

!

- Do you know what to select?



- "This element is visible".
- Select this, check the box, and you are good to go!
- Now you have the following ready to go
 - The login button should be hidden.
 - If the conditional condition is that the user is currently not logged in, the login button is displayed.

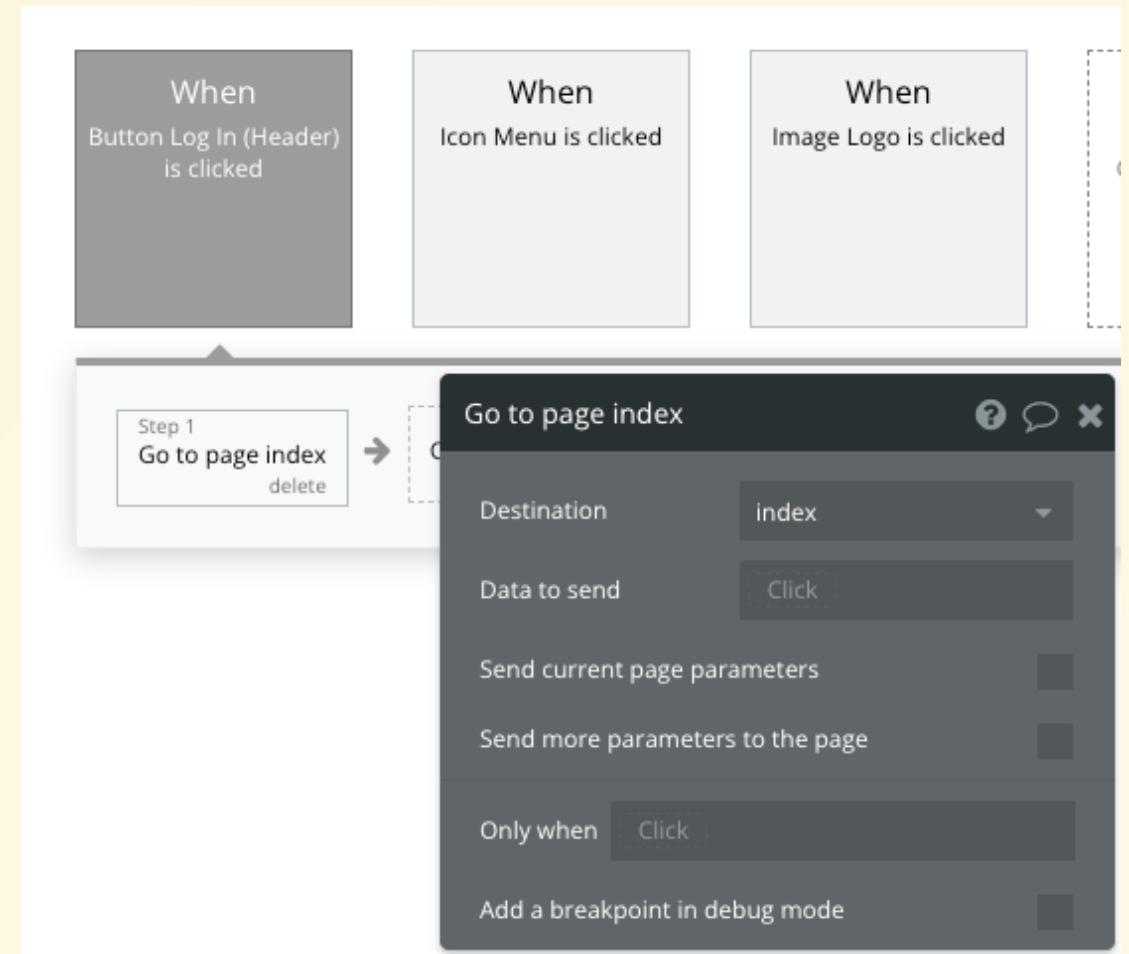


2. Pressing the "Log in" button takes the user to the login screen (index).

- Next, let's set up a workflow that takes you to the login screen when you press the "Log in" button.
- Since this is a simple screen transition, let's set it up for everyone!



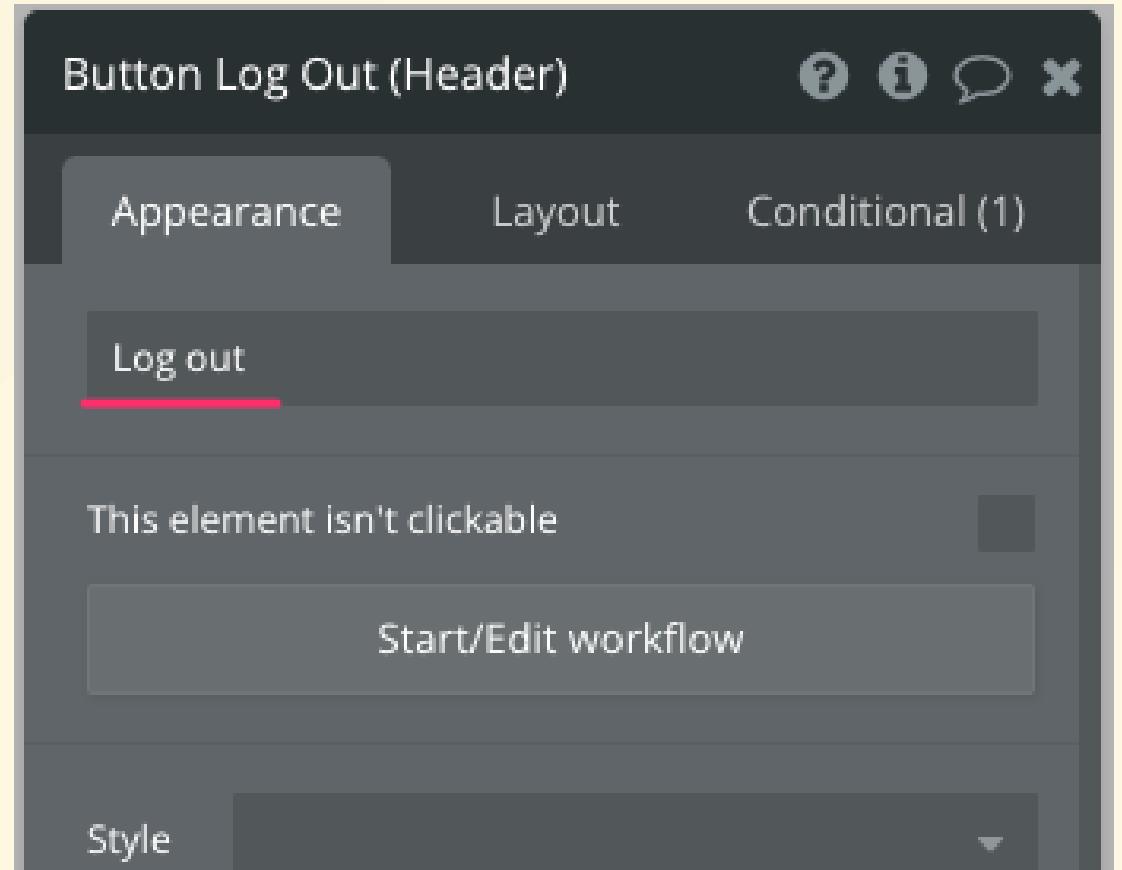
- The workflow to set up is Go to page
- Select the index of the login screen as Destination and you are good to go.



- 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

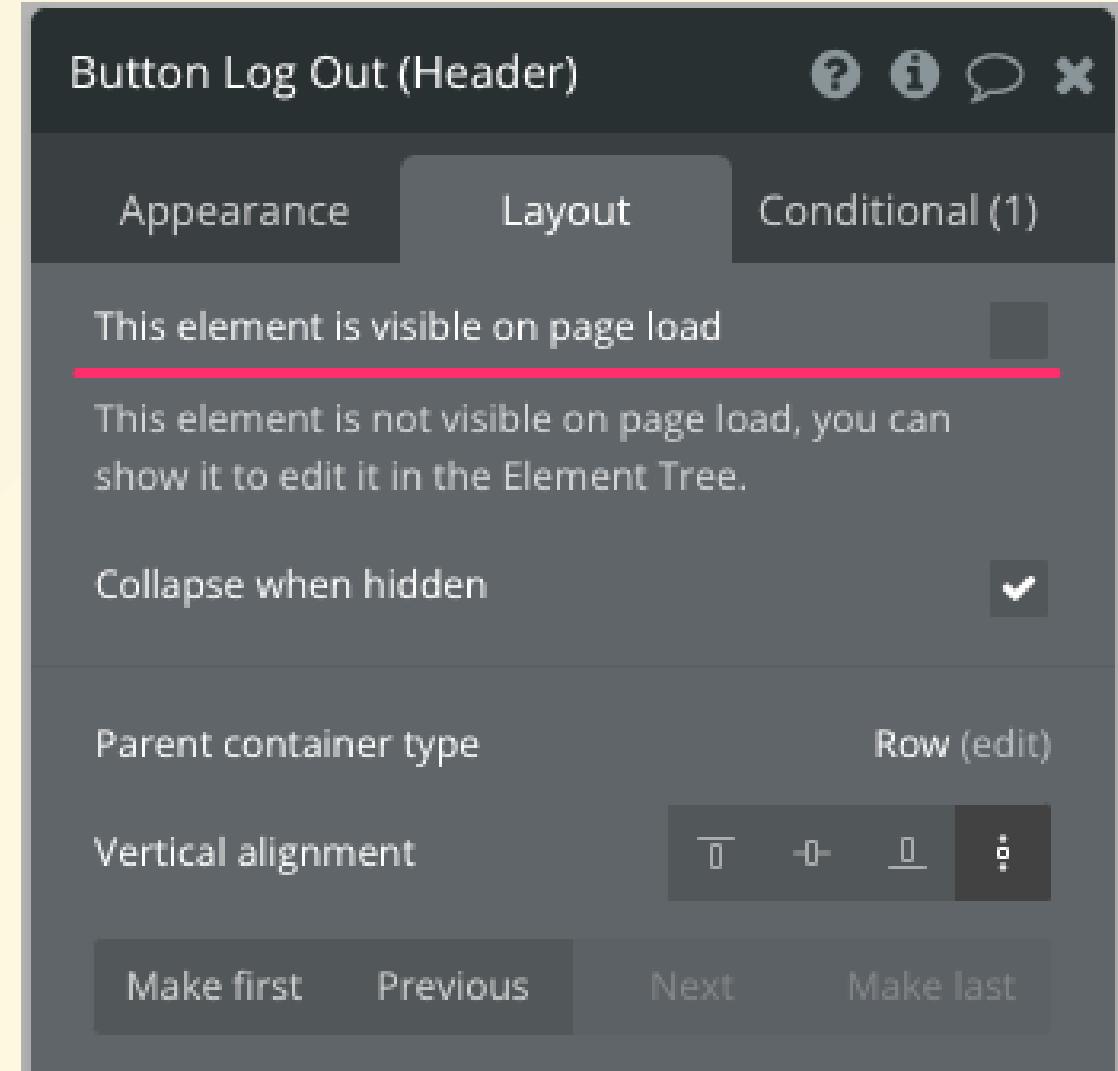
- First, change the label of the "Sign up" button to "Log out"



- 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

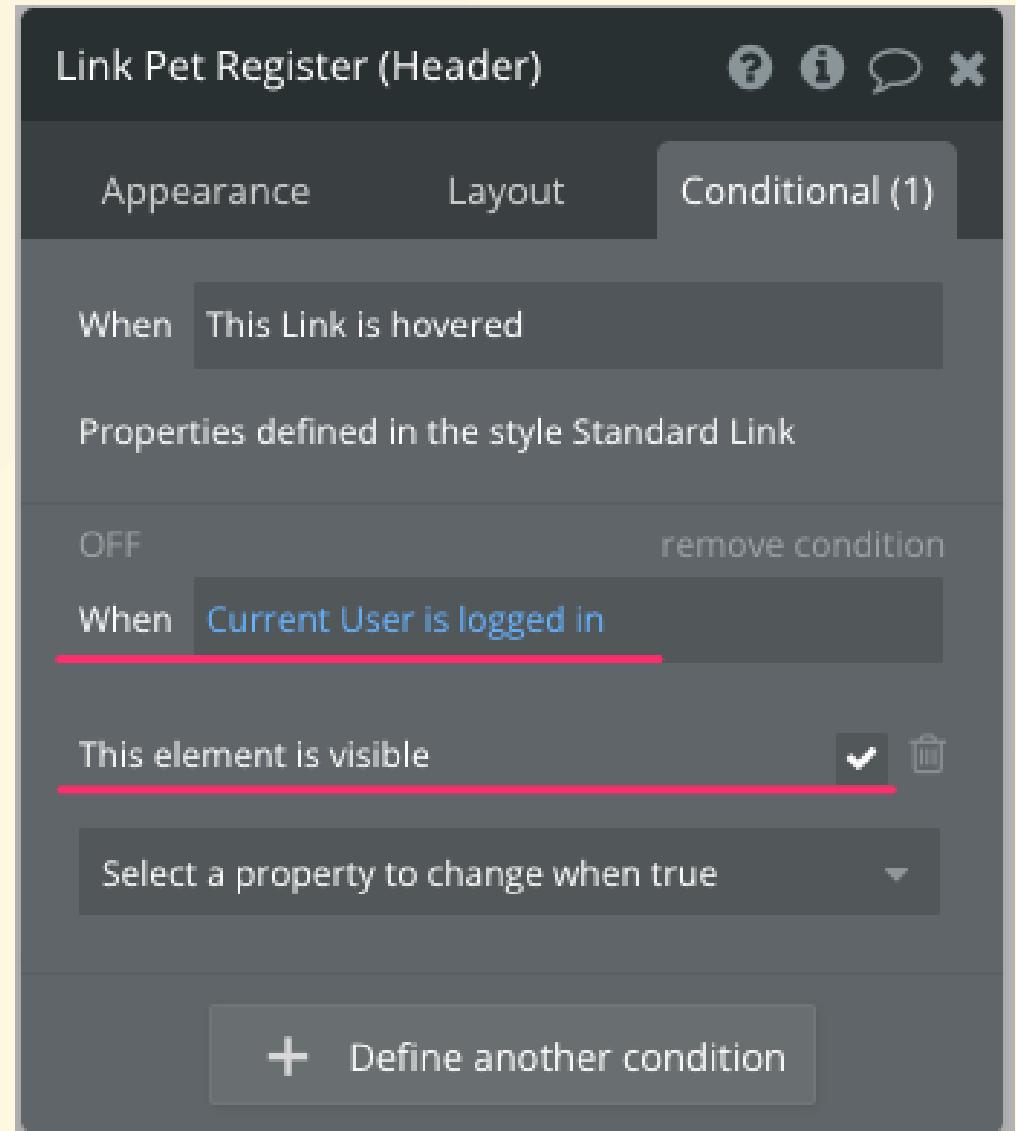
+ Define another condition

- 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)

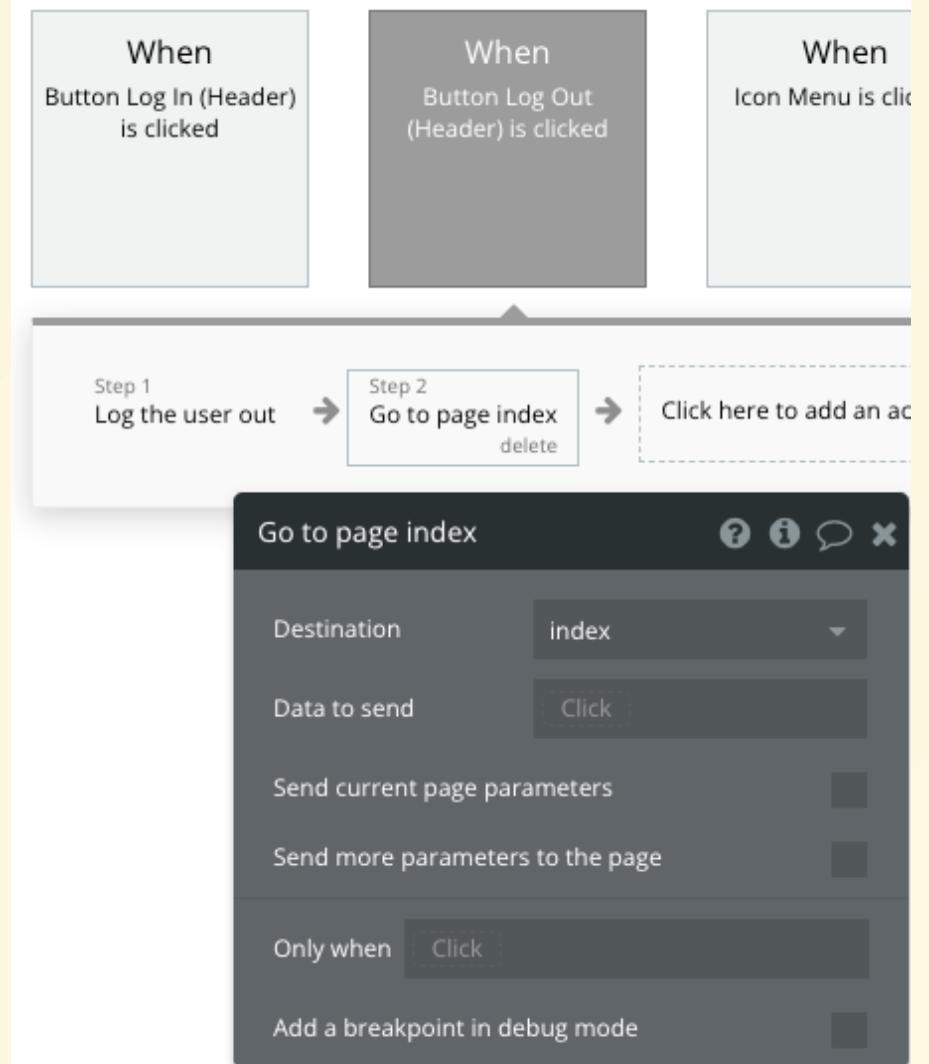
- Finally, set up the workflow when the "Log out" button is pressed.
- Let's set up the workflow when the "Log out" button is pressed!

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

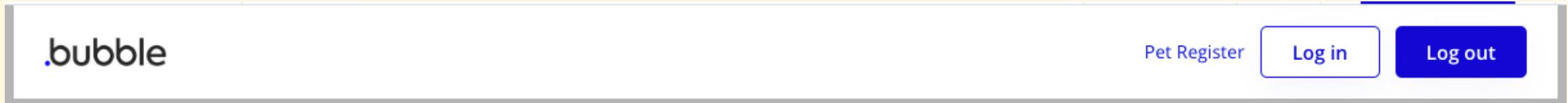
| Search for an action... | |
|---|-------------------------------------|
|  Account | Sign the user up |
|  Navigation | Log the user in |
|  Data (Things) | Signup/login with a social network |
|  Email | Log the user out |
|  Payment | Update the user's credentials |
|  Analytics | Make changes to 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

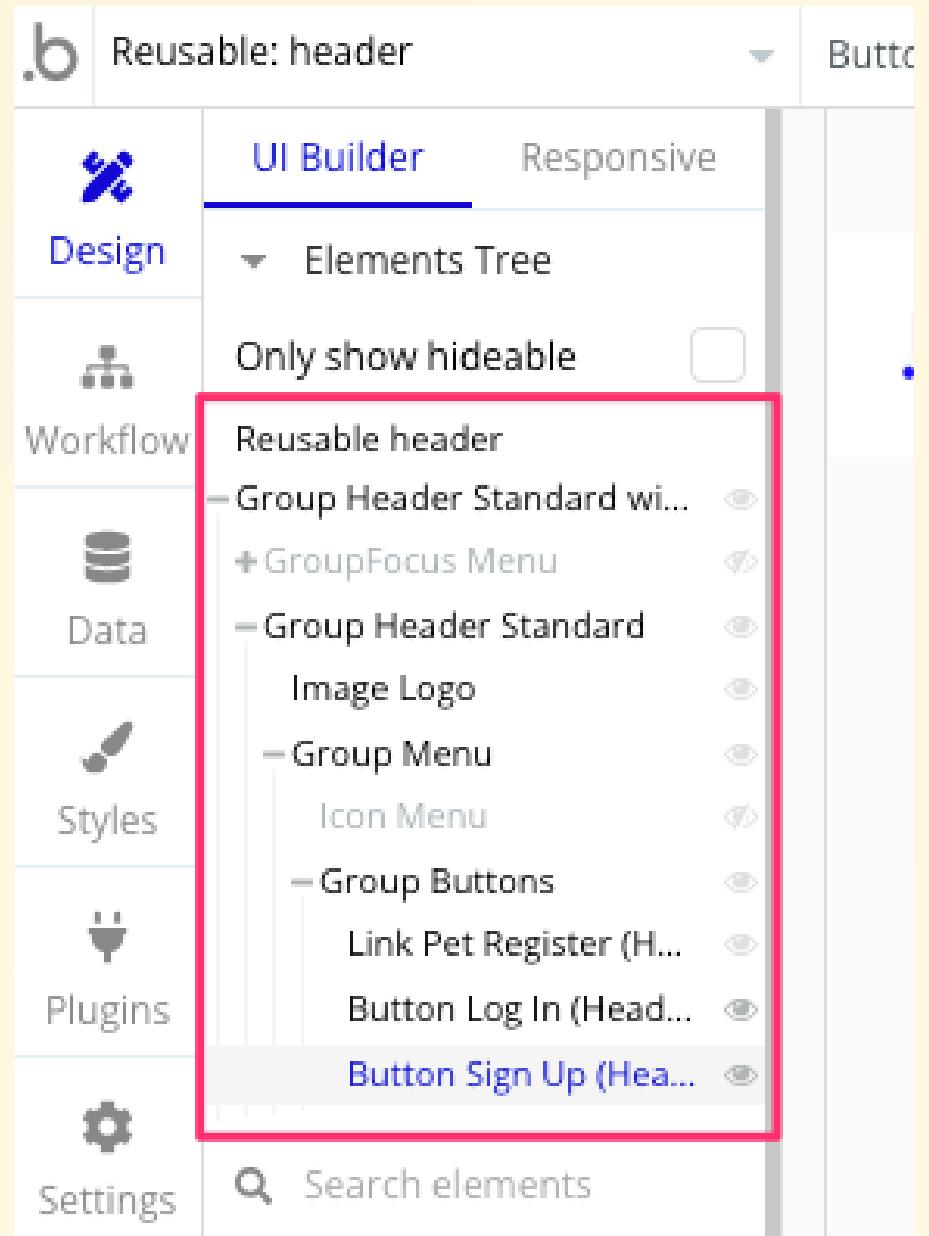
- Link to pet registration screen
- Control Login / Logout buttons based on login status
 1. the login button is displayed only when the user is "not logged in"
 2. move to the login screen (index) when the login button is pressed
 3. change the signup button to the logout button and display it "only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

By the way...

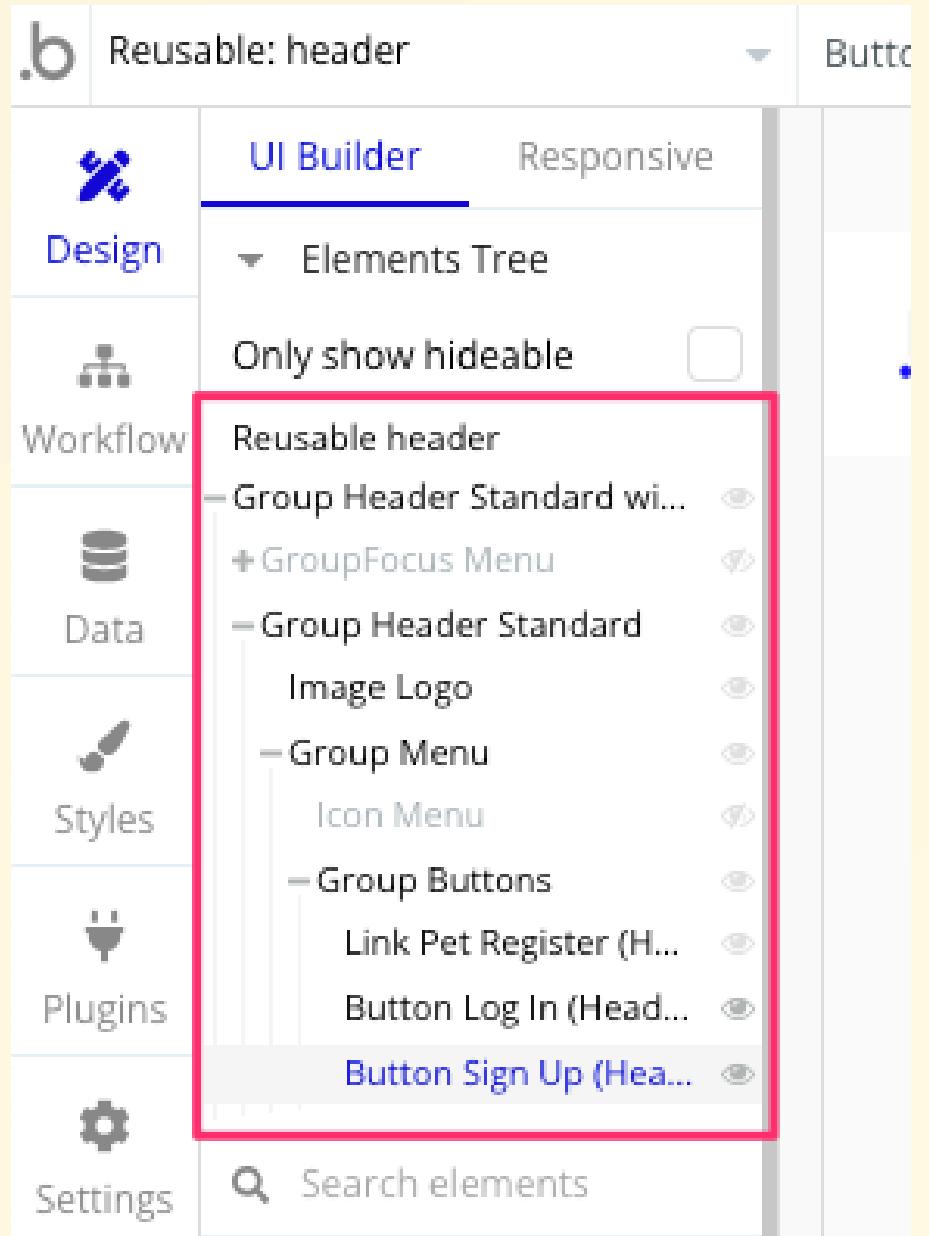
- I assume that the "Log in" and "Log out" buttons are not visible in your header part.
- This is because they are initially hidden in the settings, but there is a way to display the elements even in that state.



- 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.

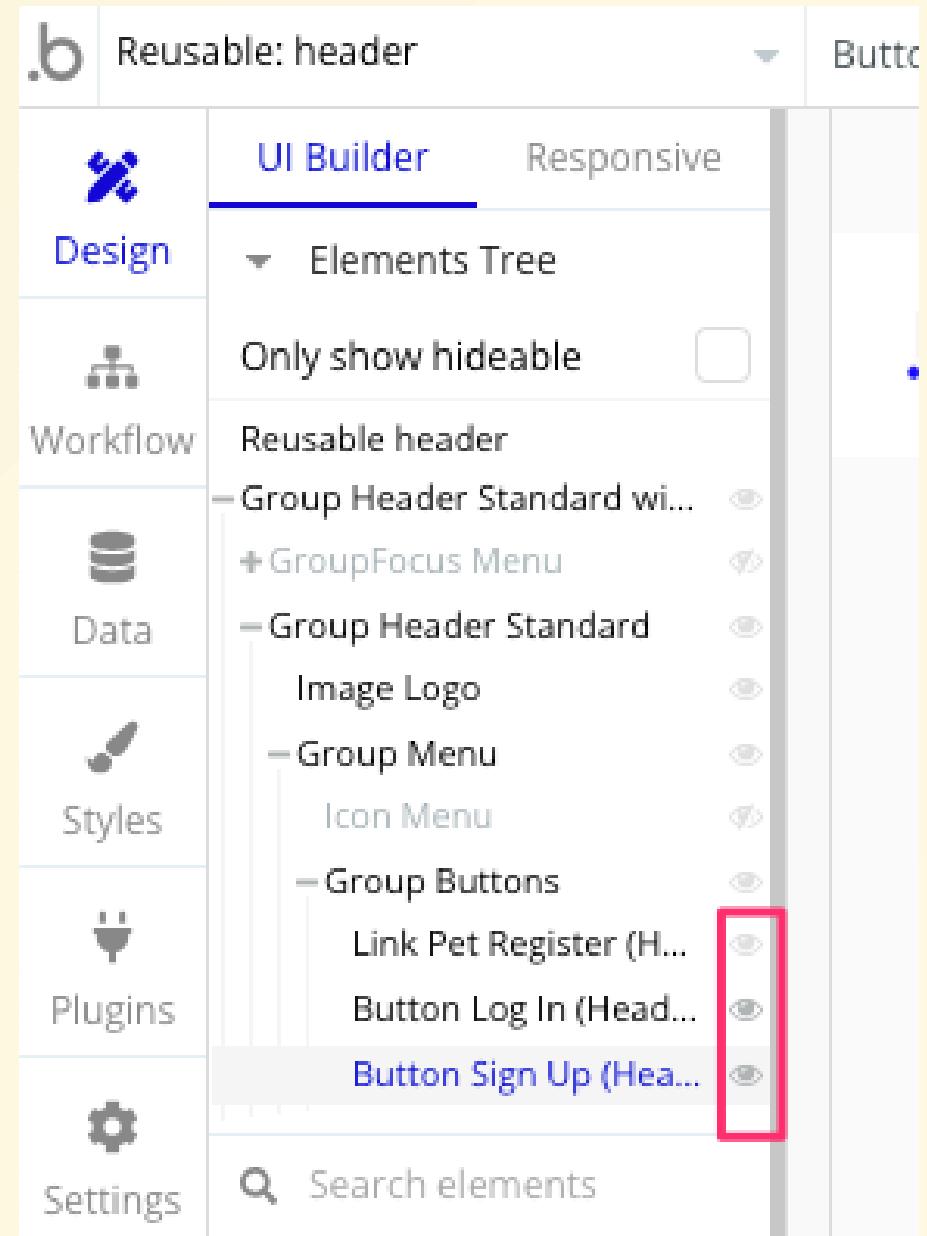


The screenshot shows the UI Builder interface with the title "Reusable: header" at the top. On the left, there's a sidebar with icons for Design, Workflow, Data, Styles, Plugins, and Settings. The "UI Builder" tab is selected, indicated by a blue underline. Below it is a section titled "Elements Tree". A checkbox labeled "Only show hideable" is present. A red box highlights a list of UI components:

- Reusable header
- Group Header Standard wi...
- + GroupFocus Menu
- Group Header Standard
- Image Logo
- Group Menu
- Icon Menu
- Group Buttons
- Link Pet Register (H...
- Button Log In (Head...
- Button Sign Up (Hea...

 At the bottom of the tree view is a search bar with the placeholder "Search elements".

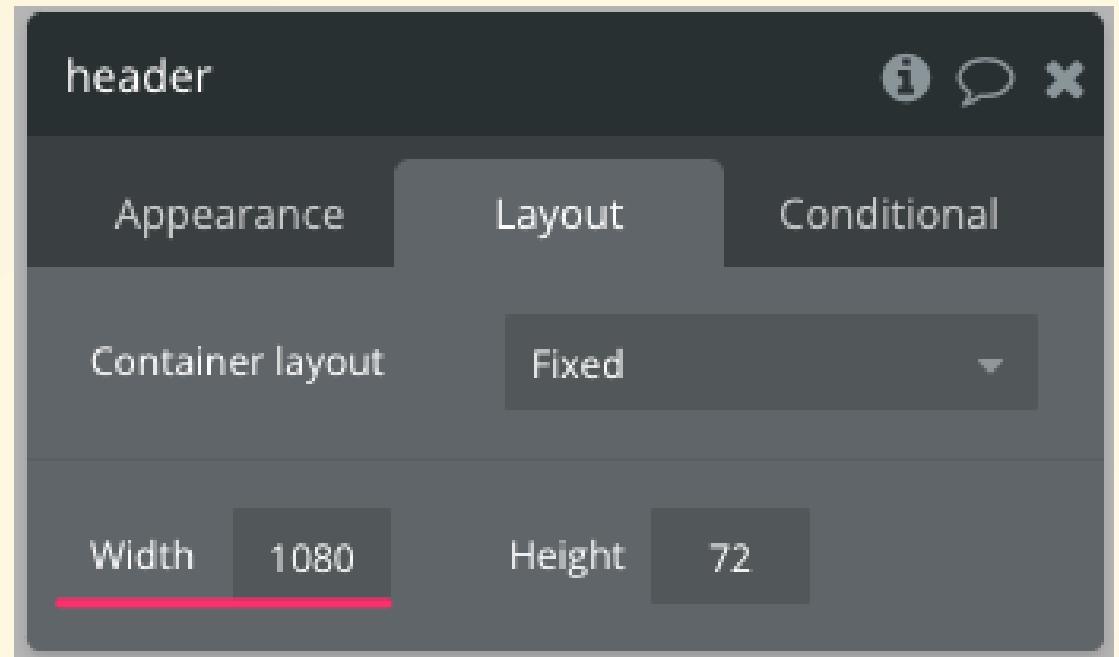
- Then, among the elements at the very front (bottom) of the panel, there is an "eye" icon to the right of the "Log in" and "Log out" (Sign in) buttons: 
- This has an ON and OFF status, ON means it is displayed on the right panel, OFF means it is not displayed on the right panel.



- This time, turn off the "eye" icon on the "Log in" and "Log out" (Sign in) buttons and turn them on again to make the buttons visible on the right panel.
- (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.)



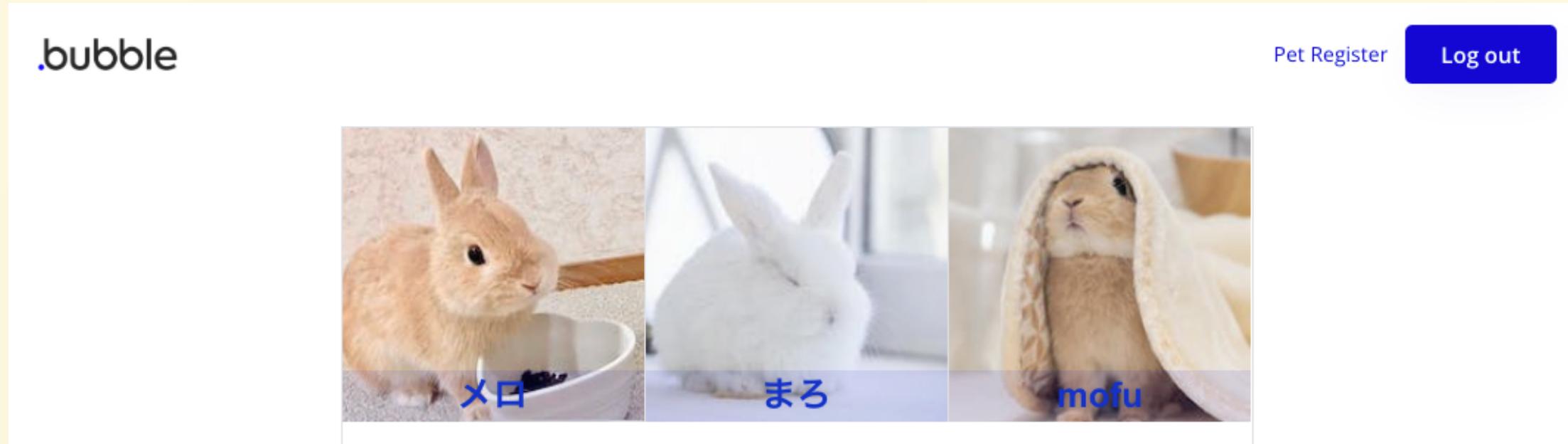
- Finally, to match the width of the current header component with the width of the screen, select the following element from the **Elements tree** and change the value of **Width** to "1080".
 - "Reusable header"
 - "Group Header Standard with Menu"



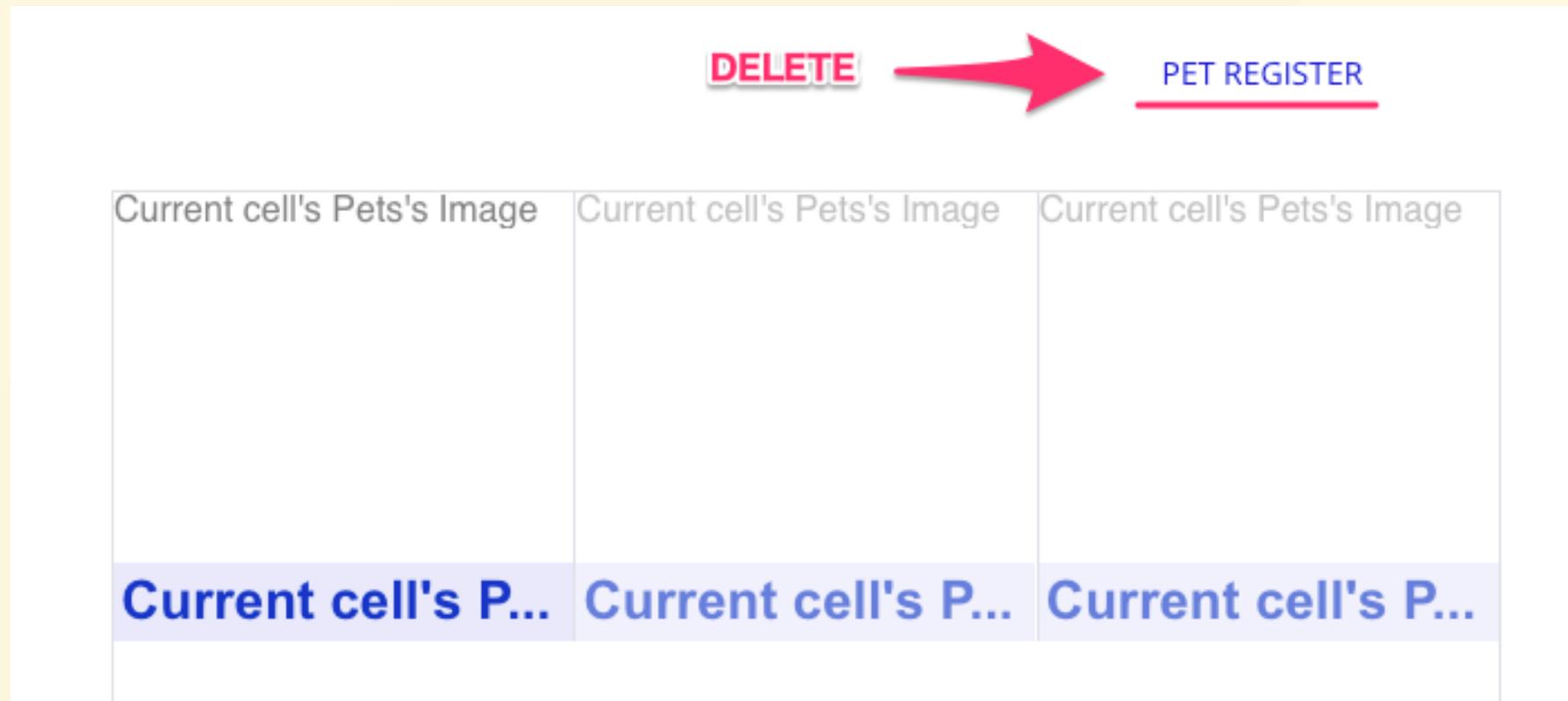
- If you can no longer see the "Pet Register" link or the login button because you have changed the `Width`, follow the same procedure as before and click on the "eye" icon of the button or other element in the `Elements tree` and turn on the display.

Let's incorporate it into each screen.

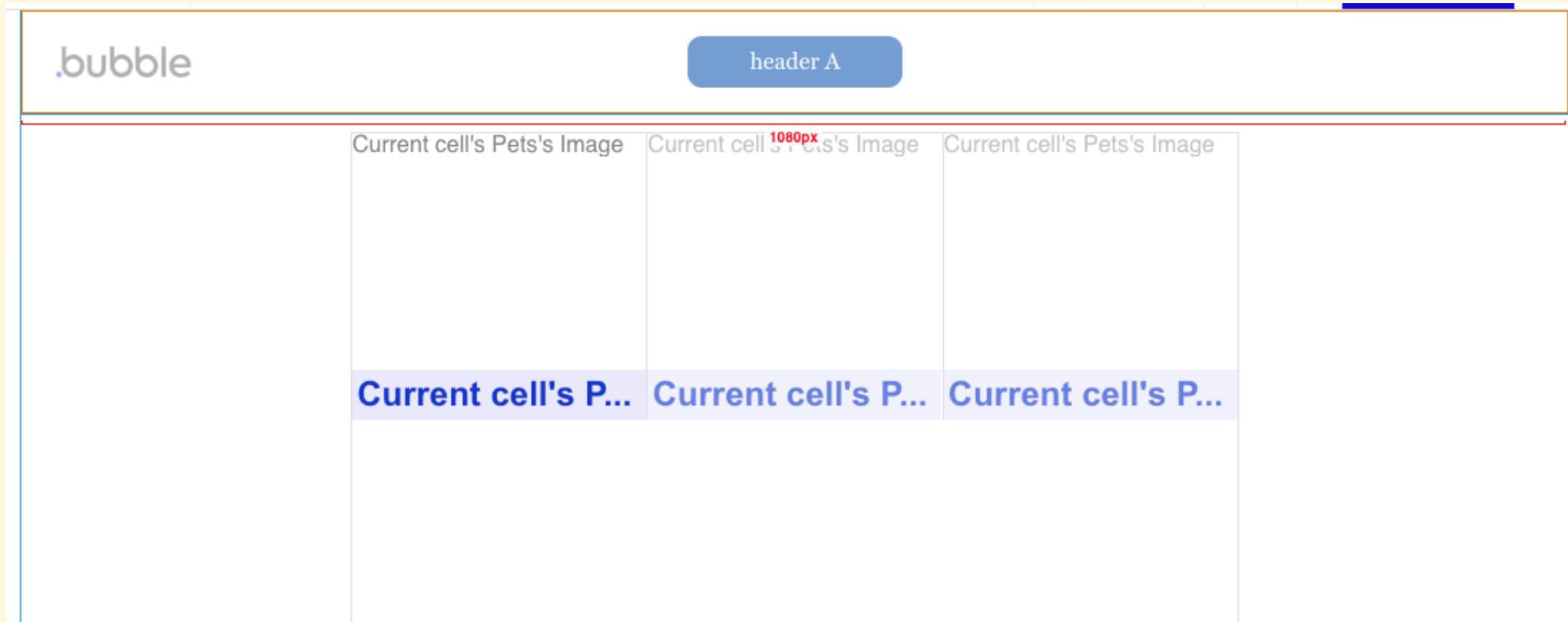
- Now that the header component is ready, let's first incorporate it into the Pets List screen.



- 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!



Let's preview it now



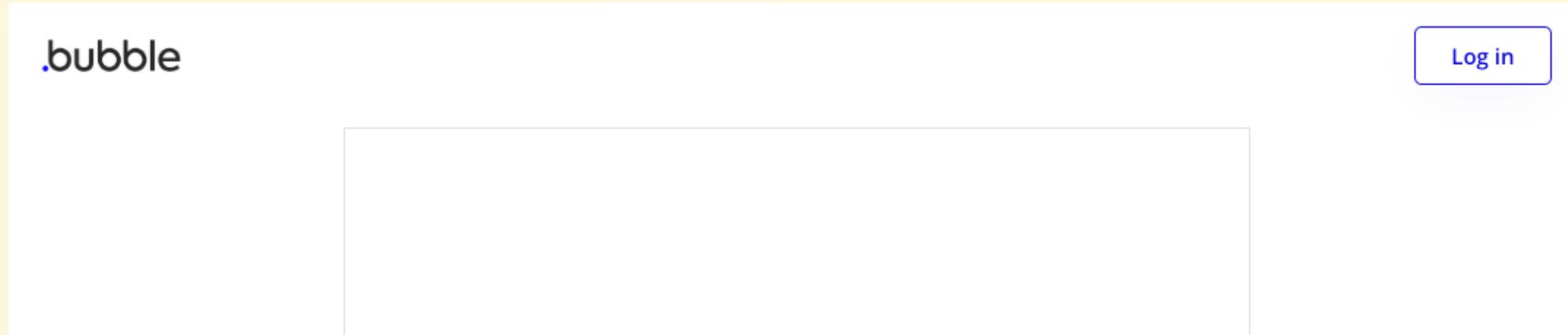
If you are logged in

- 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 screens, 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.

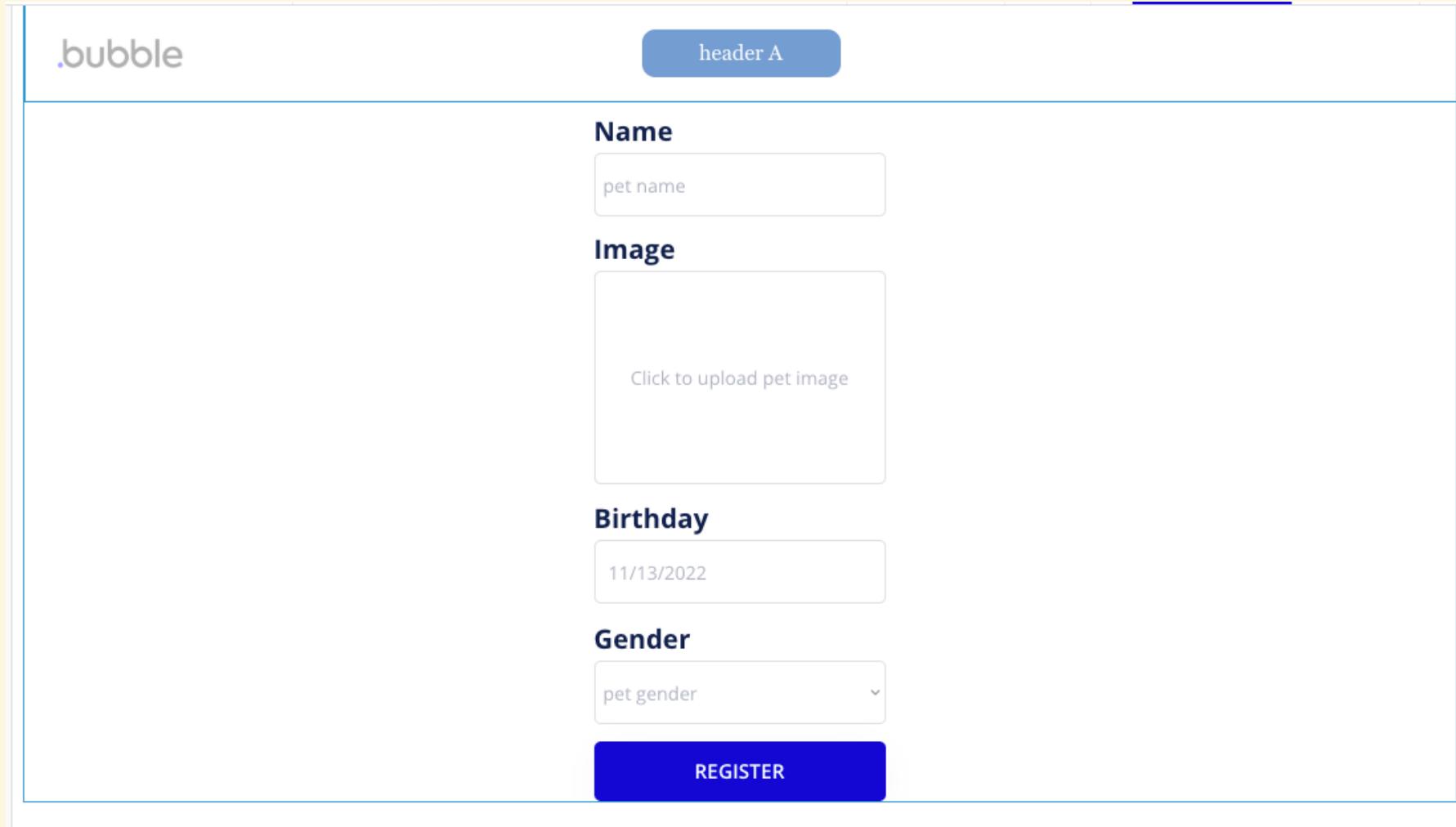
Let's Try!! 🔥



- It's easy for everyone 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 (Sign Up / Log In) is clicked"
 - "When Button Sign Up (Sign Up / Log In) is clicked"

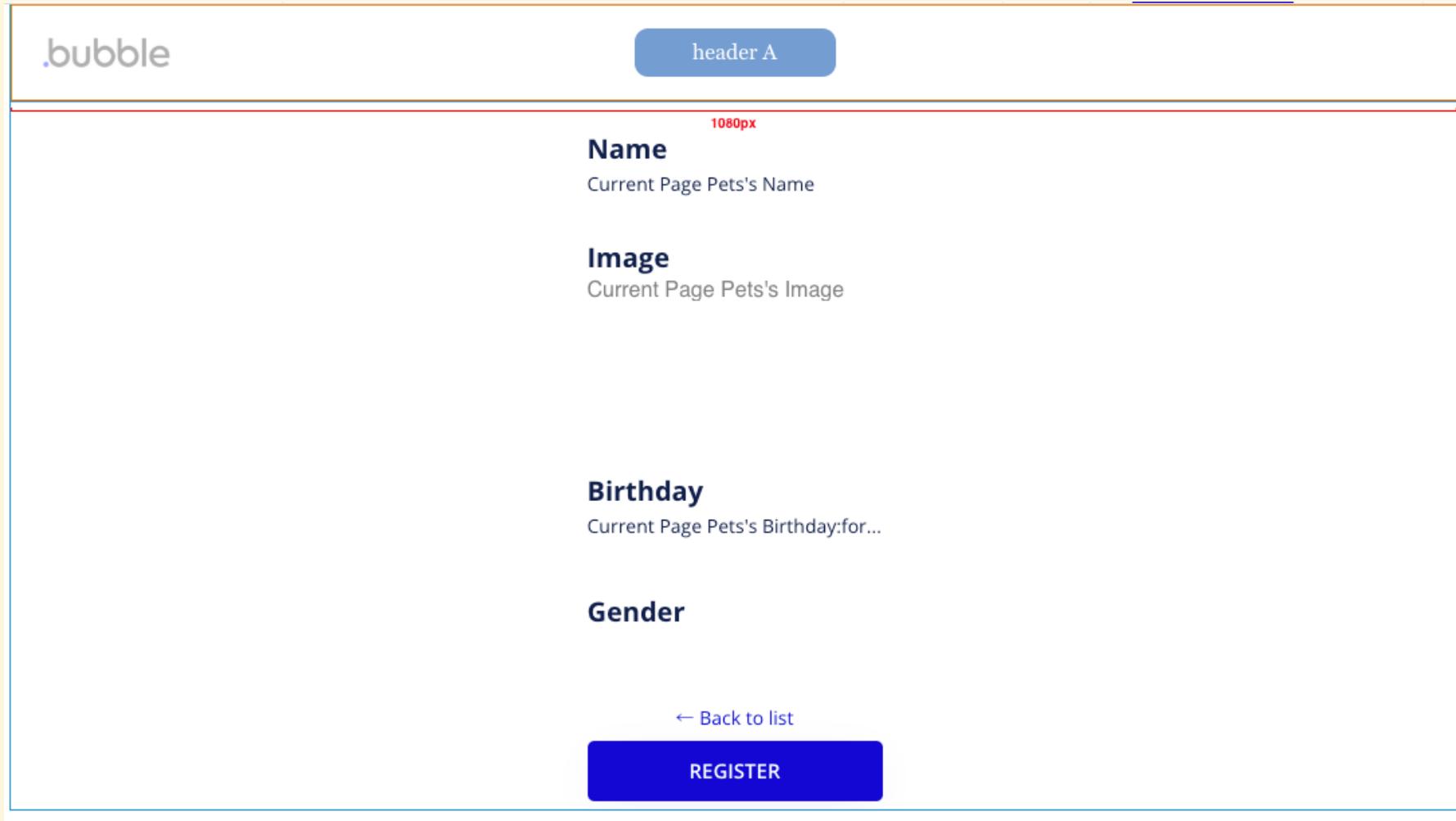
- Preview the page to make sure that the user will be taken to the Pets List page when he/she presses the Login button and the Sign Up button, respectively!
- If the preview is OK, let's place the common header components on the other two screens as well.
 - Pet Registration Screen
 - Pets Detail Screen

- 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 with placeholder "pet name"), "Image" (button with placeholder "Click to upload pet image"), "Birthday" (text input with placeholder "11/13/2022"), and "Gender" (dropdown menu with placeholder "pet gender"). At the bottom center is a large blue button labeled "REGISTER".

• Pet Details Screen



- Let's also preview the registration and detail screens

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

A screenshot of a pet registration form. At the top right are 'Pet Register' and 'Log out' buttons. The form fields include 'Name' (placeholder 'pet name'), 'Image' (placeholder 'Click to upload pet image'), 'Birthday' (placeholder '11/13/2022'), and 'Gender' (placeholder 'pet gender'). A large blue 'REGISTER' button is at the bottom.

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.

- Pet weight entered from the screen
 - WeightKg: number
- Tie which pet's weight.
 - Pet: Pets

The screenshot shows the Backstage application interface for managing custom data types. The left sidebar includes tabs for Design, Workflow, Data (which is selected), Styles, Plugins, Settings, and Logs. The main area has tabs for Data types, Privacy, App data, Option sets, and File manager. Under the Data types tab, there is a section for "Custom data types". A table lists existing data types: "Pet" (Publicly visible), "PetWeightLogs" (selected and highlighted in red, Publicly visible), and "User" (Privacy rules applied). The "PetWeightLogs" row shows its type name as "PetWeightLogs" and contains fields for "pet" (type "Pets") and "WeightKg" (type "number"). Other fields listed include "Creator" (User, Built-in field), "Modified Date" (date, Built-in field), "Created Date" (date, Built-in field), and "Slug" (text, Built-in field). A "Create a new field" button is at the bottom of the list.

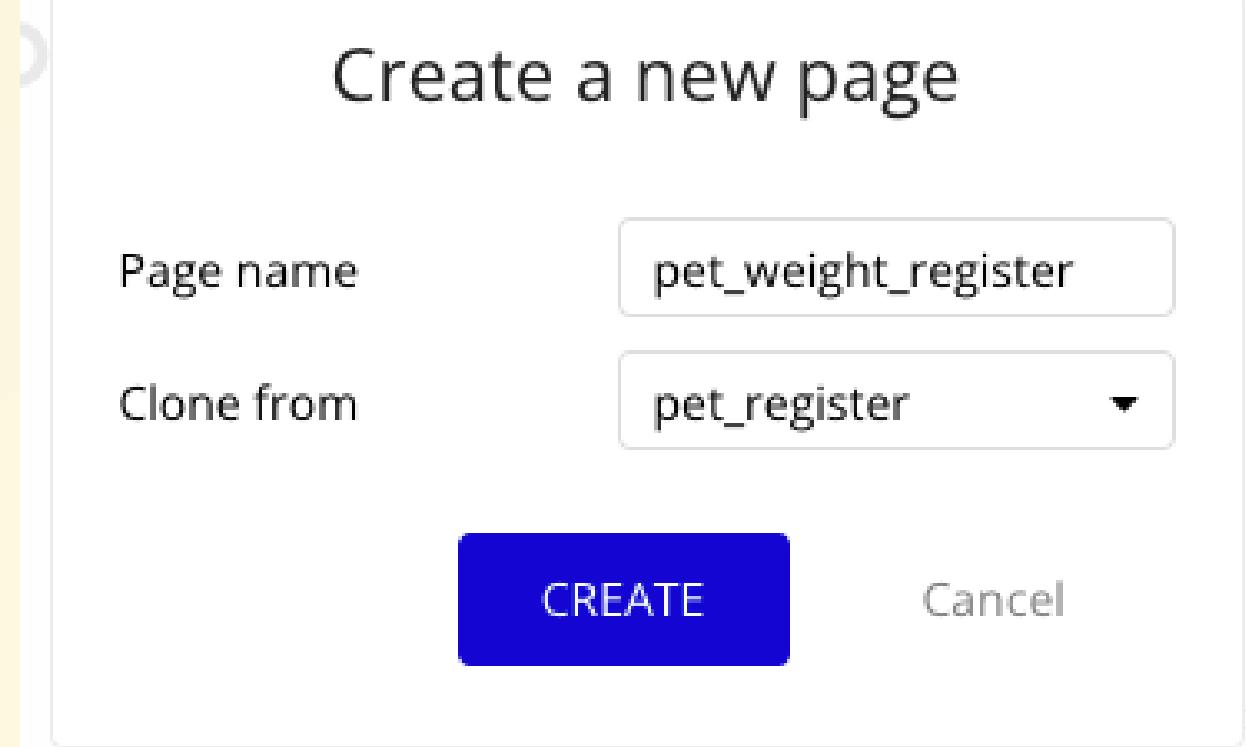
- Now that the database is ready, let's set up the screens!
- First, let's set up a new page for pet weight management.
- From **Add a new page**, select **pet_weight_register** as the page name and **pet_register** as the copy source, and **CREATE**

Create a new page

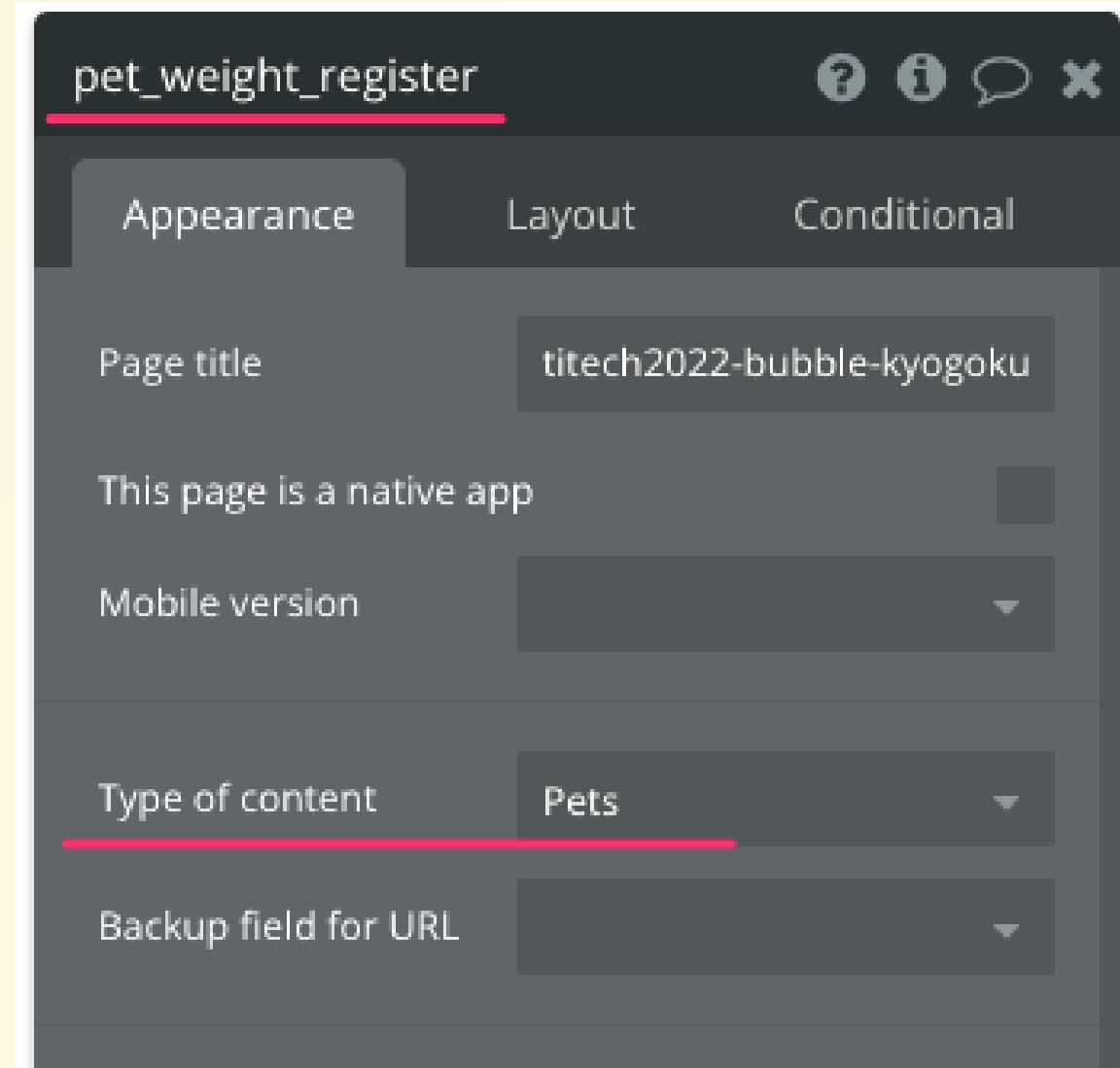
Page name

Clone from ▾

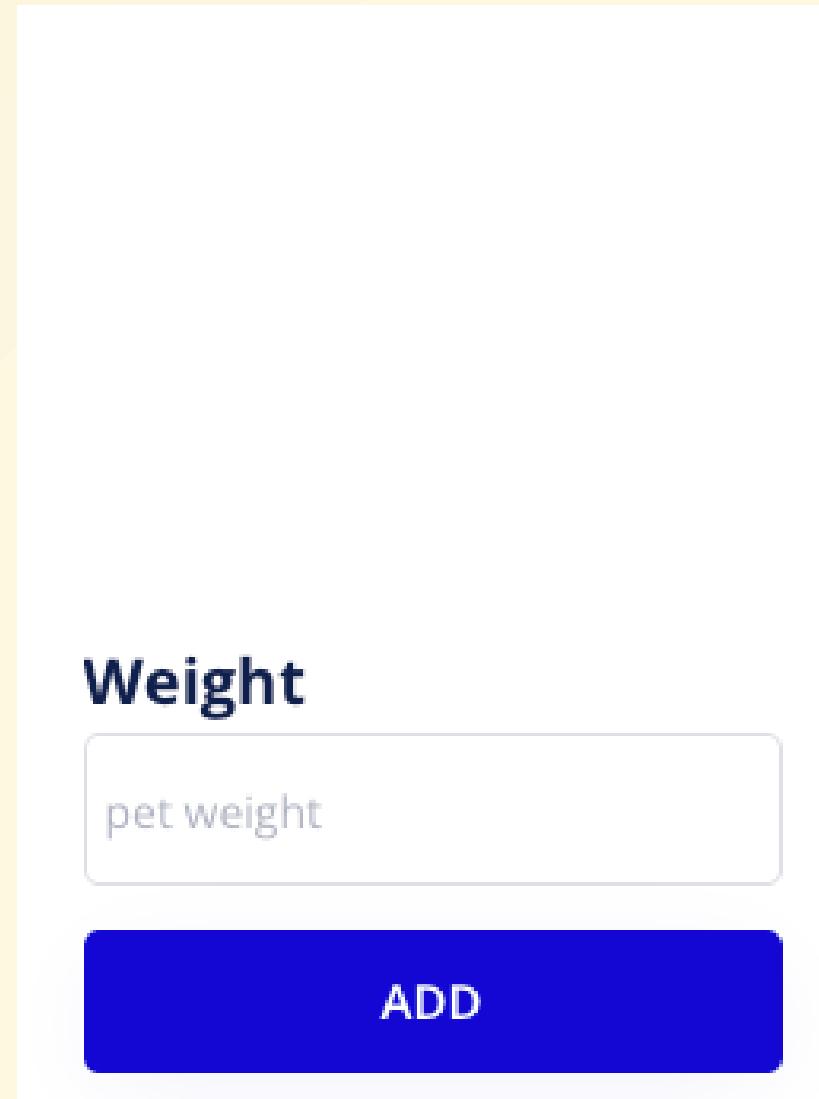
CREATE Cancel



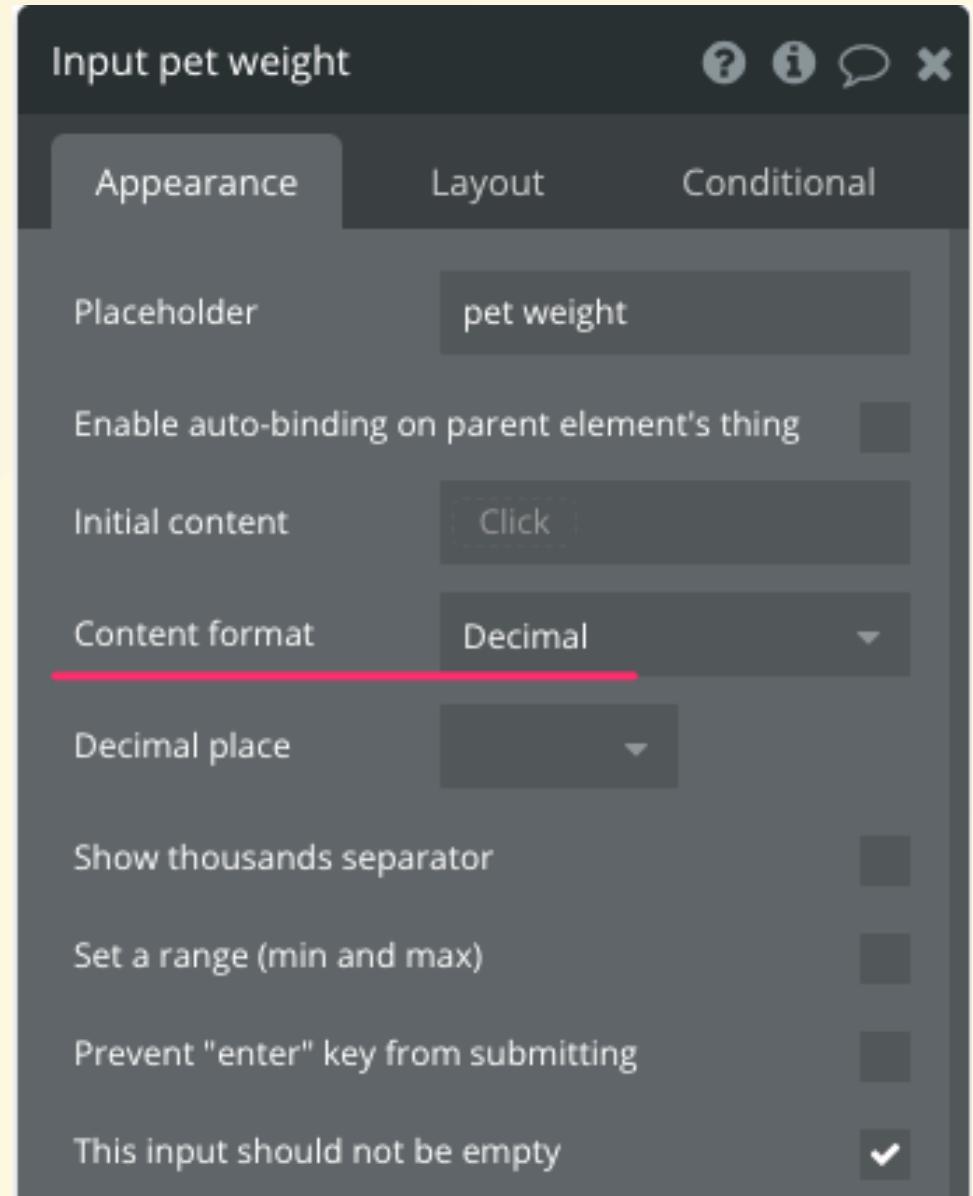
- This screen is also supposed to be accessed from the pet details, so from the advanced settings dialog for the `pet_weight_register` screen, specify "Pets" for the Type of content.



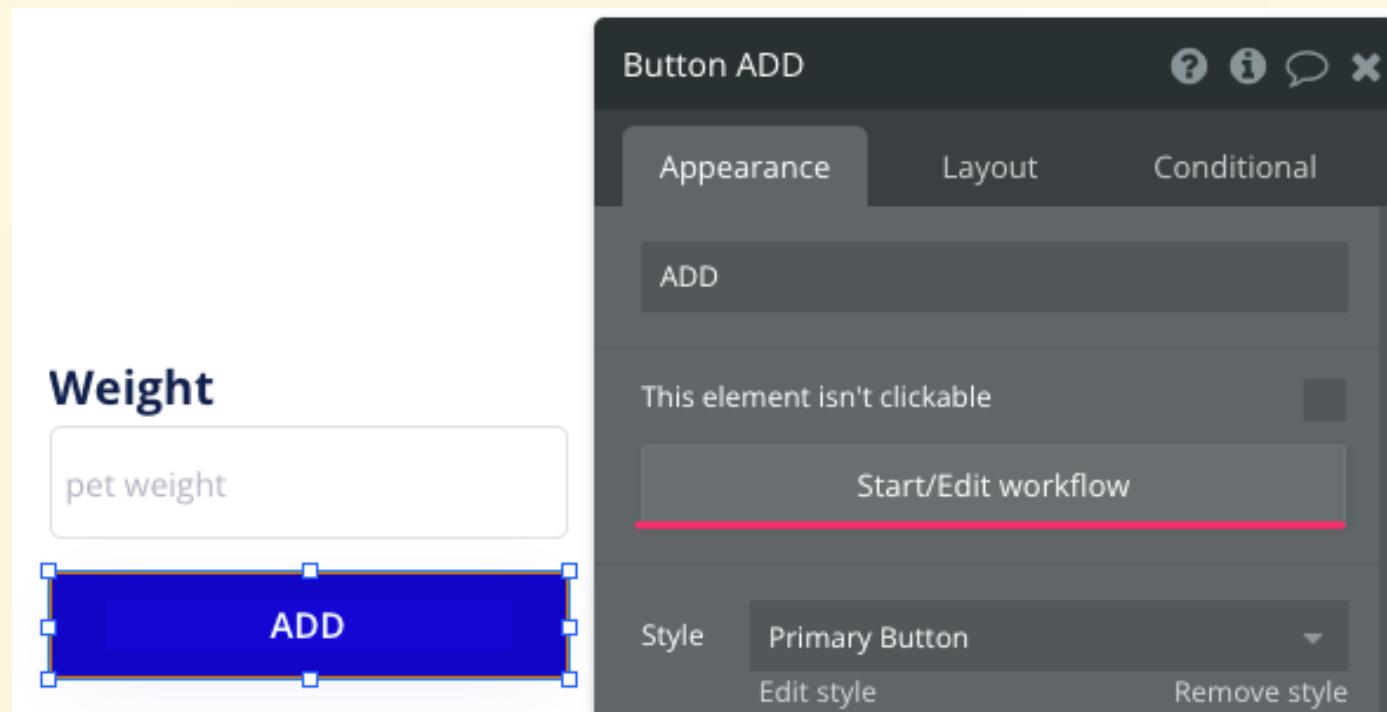
- The elements of the original pet registration screen are still there, so we will place the elements as shown in the reference image while deleting the unnecessary ones.
- However, the graph drawing area is not placed at this time.



- 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 Edit workflow from the Advanced dialog



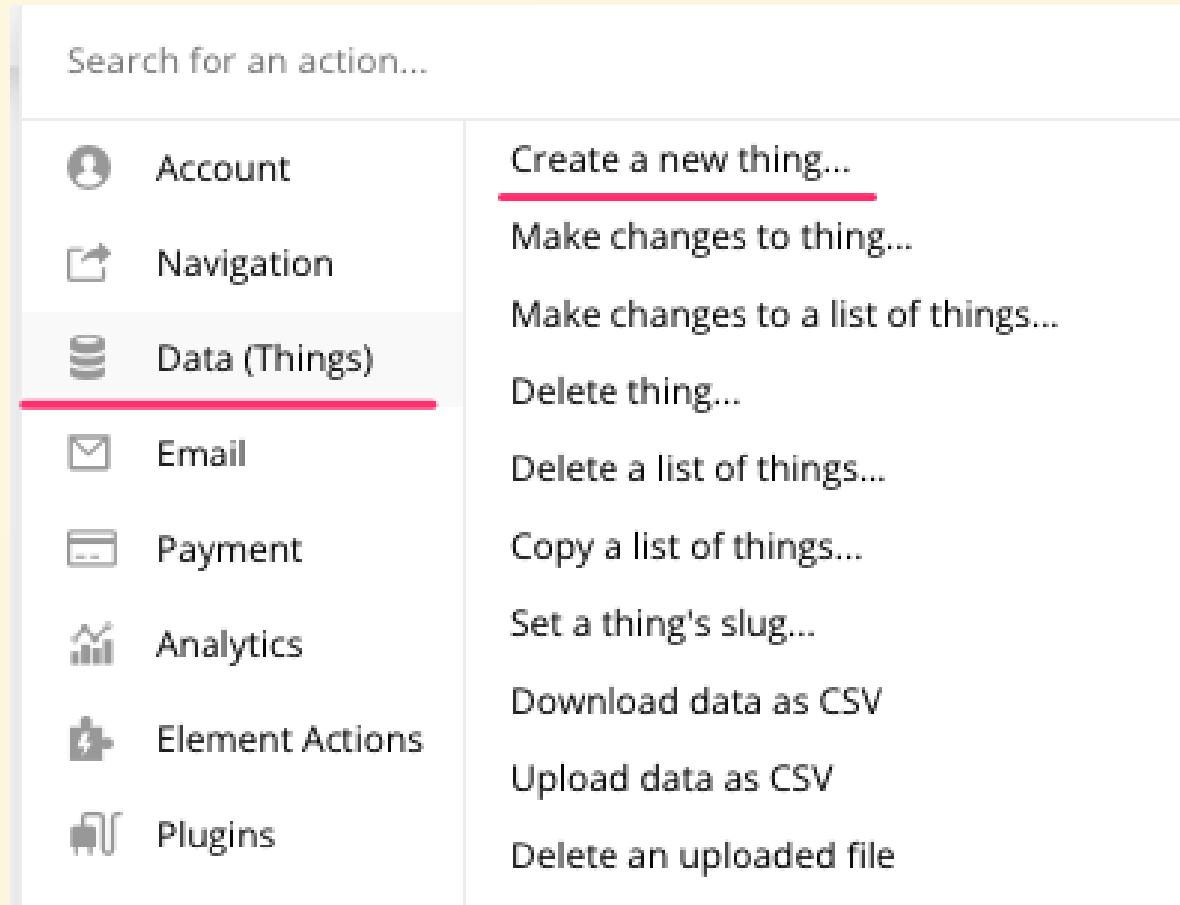
- Then, for `When Button ADD is clicked`, some behaviors are already set!
- This is because we copied `pet_register` when we created the page, so the workflow settings are inherited from that time.
- This time, we will define a new behavior, so we will delete the existing workflow.
- When you select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



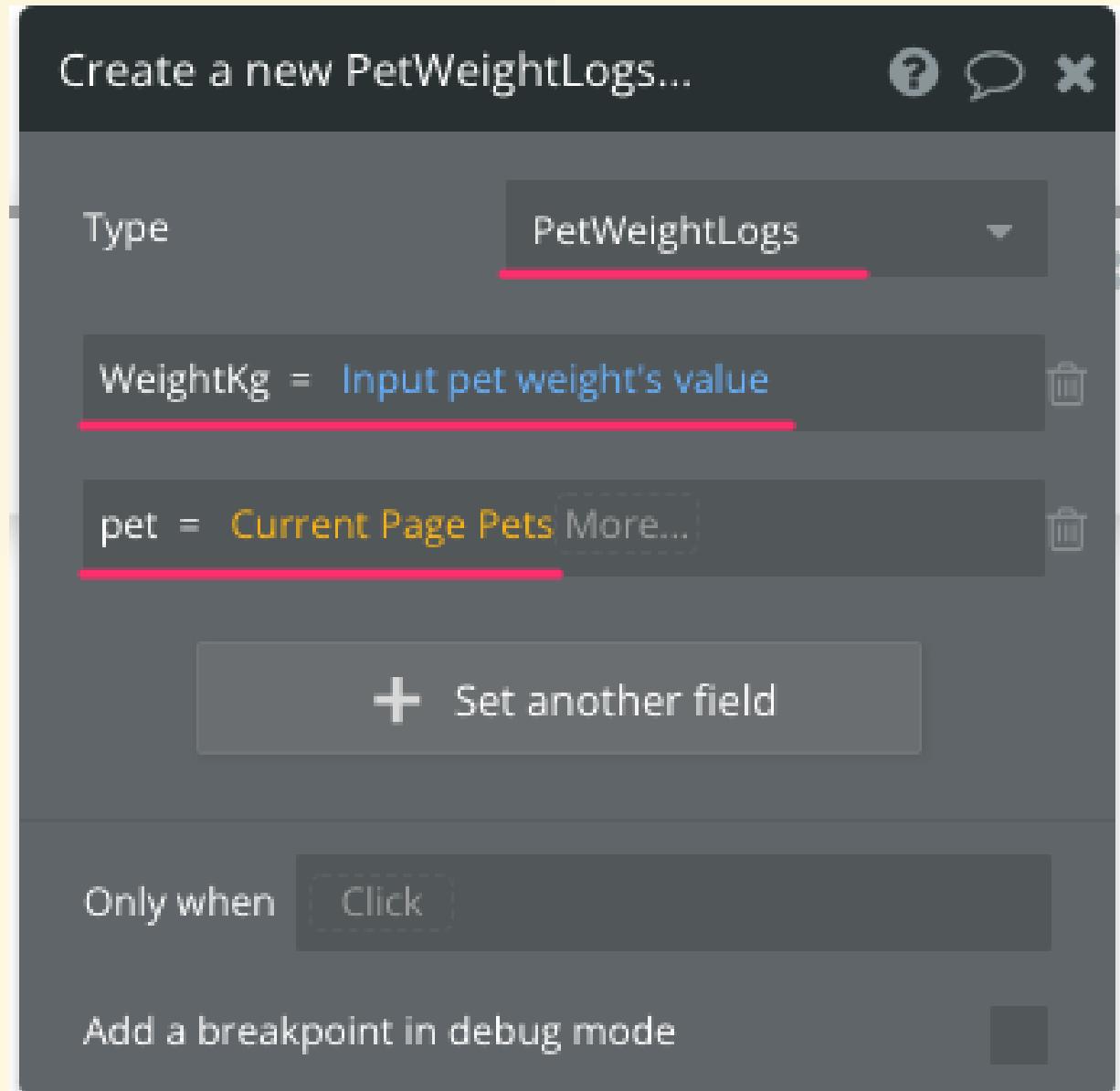
- Now let's set up the workflow.
- The first thing we want to do is "save the weight values entered on the screen to the PetWeightLogs we prepared earlier"!
- Please try to set this up based on the contents of the lecture so far!
- The explanation is on the next page, so let's do that first!



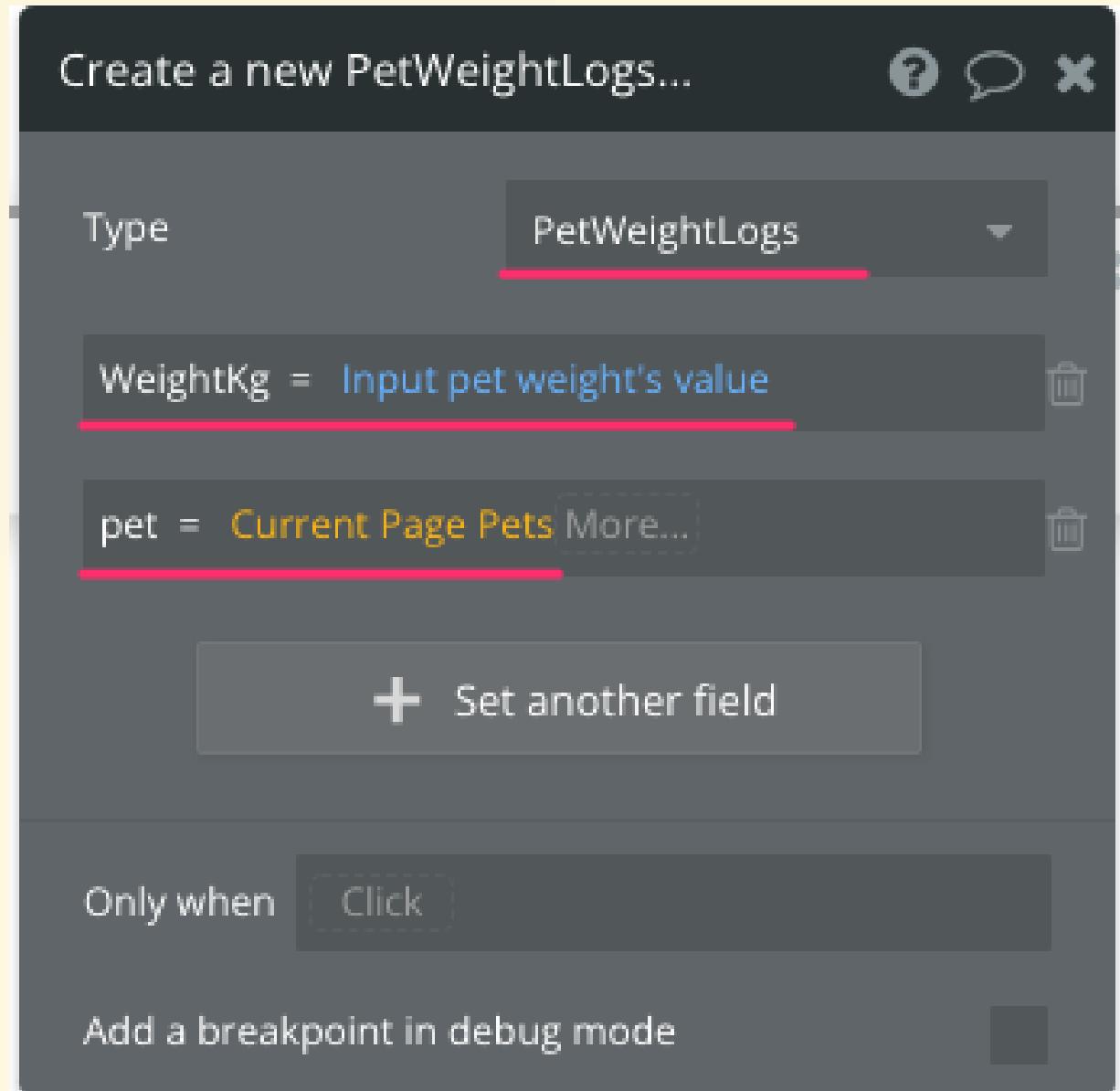
- The first action to set up is **Data(Thing)** --> **Create a new thing...**



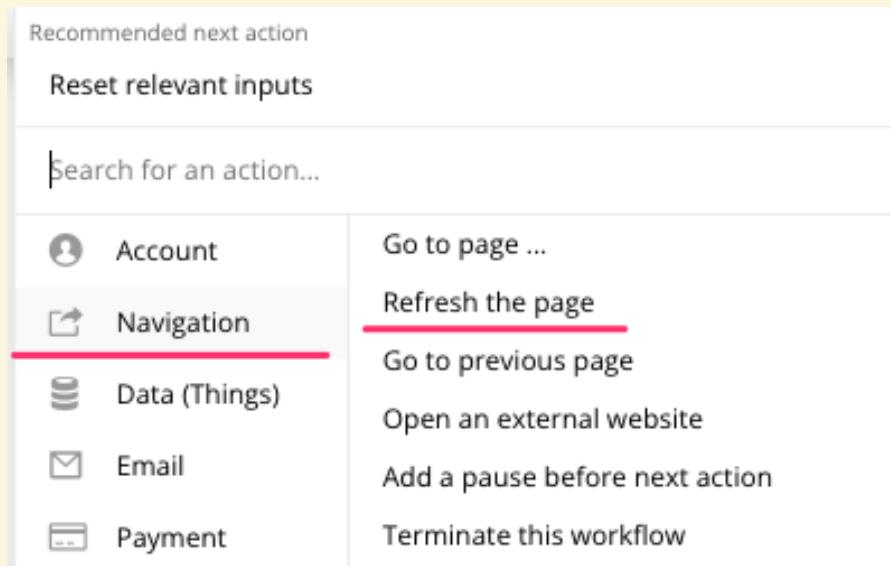
- Then, specify the type and fields to register
- Since we are registering the pet's weight, the Type is PetWeightLogs .



- 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 currently displayed on the screen.



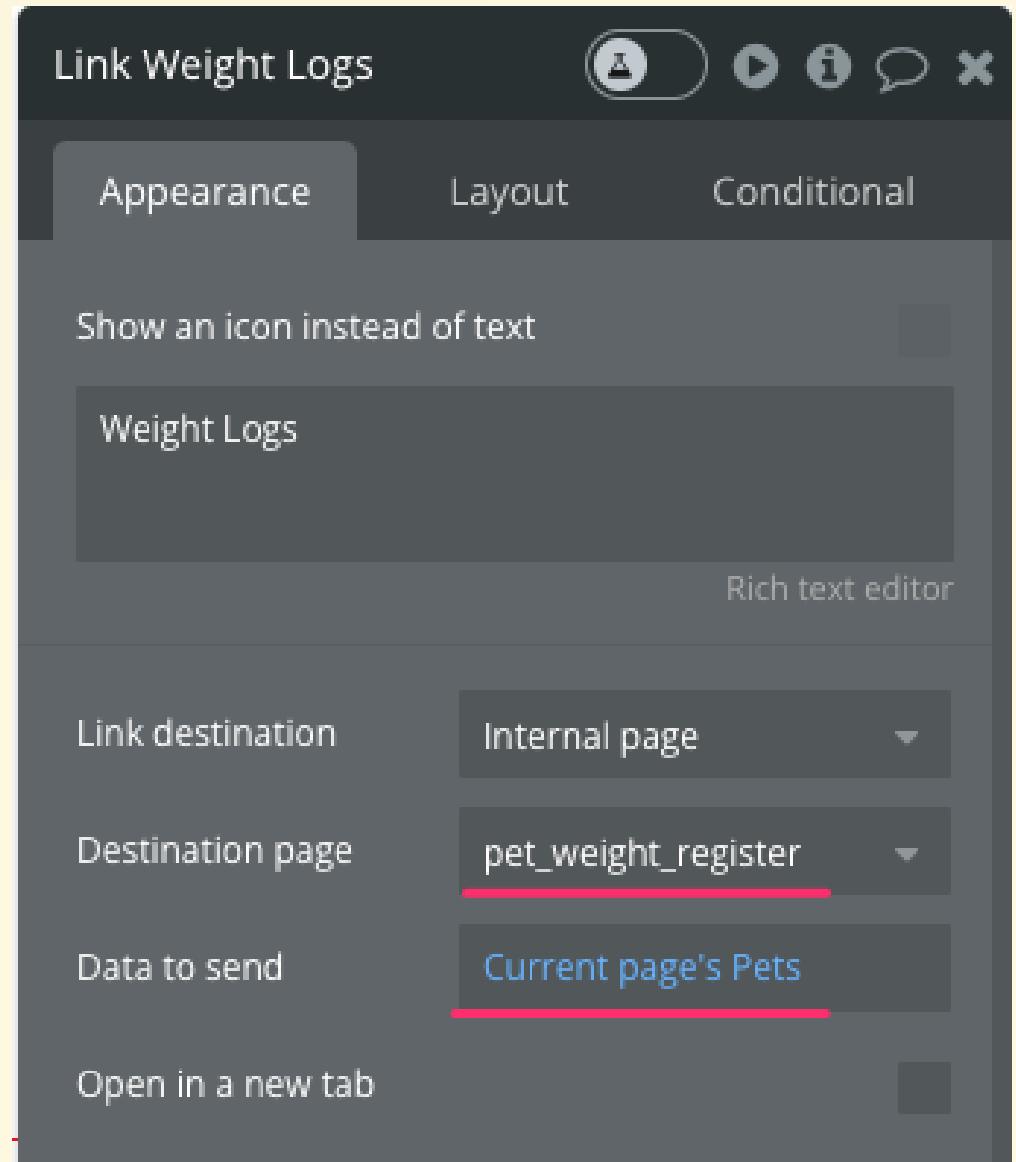
- We want to redraw the screen after registration to refresh the graph to be displayed later, so let's define the behavior to redraw the screen after registration
- The action for manipulating the screen is **Navigation**.
- If you look at the submenu, you will find a suitable action **Refresh this page**.



- 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.

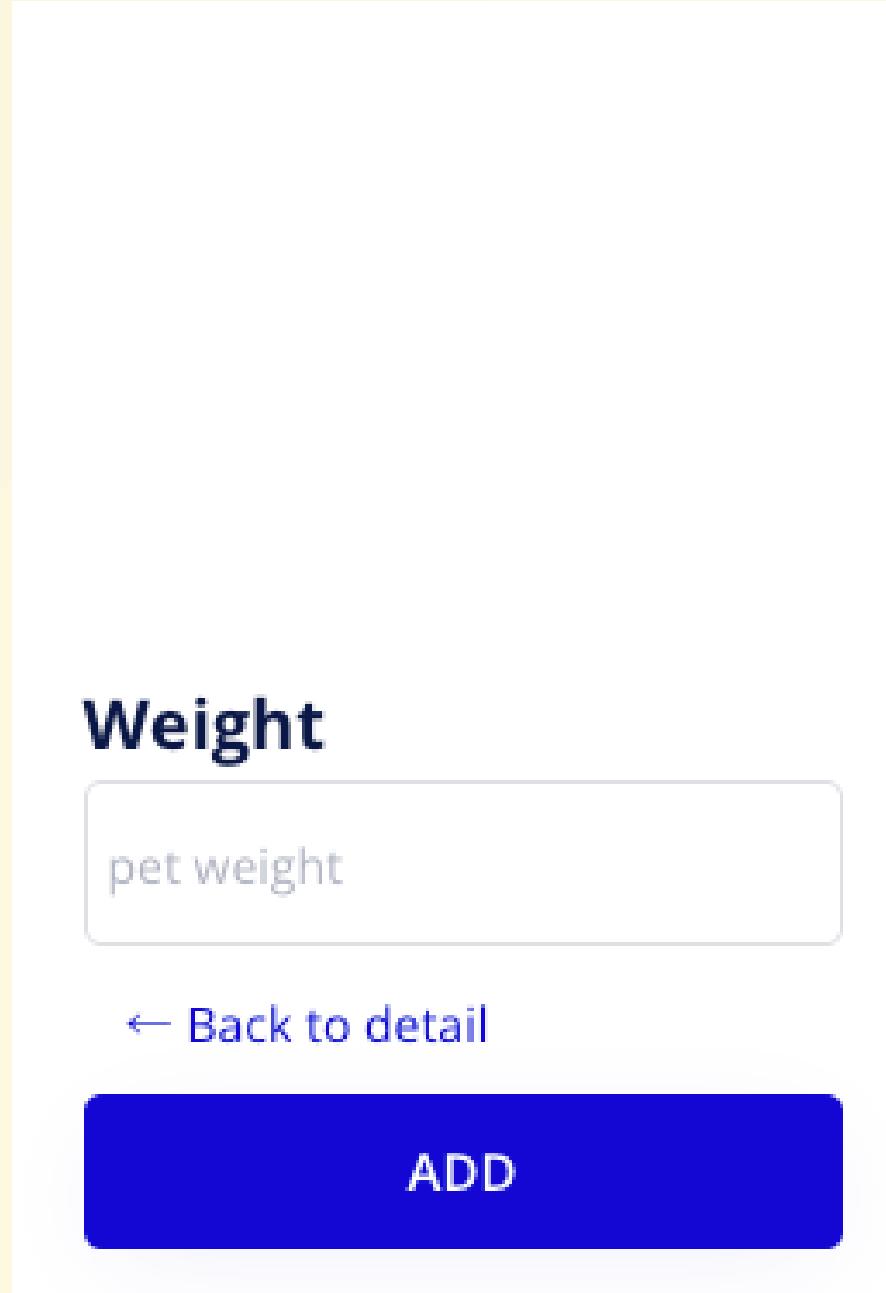
- There are two points
 1. set the destination to Destination page
 2. Give the link destination the pet information of the current page.
- The following page shows an image of the setup, so let's start by setting it up yourself!

- Like this!



- 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 (which is selected), Styles, Plugins, and Settings.
- Top navigation:** Buttons for Data types, Privacy, App data (selected), Option sets, and File manager.
- Header:** Database views, Application data - All PetWeightLogs - Development version, Copy and restore database, and Switch to live database.
- Tool buttons:** New view, Primary fields, Search, 2 entries (displaying 2), New entry, Delete (0), Upload, Modify, Export, and Bulk.
- Search bar:** Search view names or data types...
- View title:** All Pets.
- Table:** A grid showing two entries of pet weight logs. The columns are Pet, WeightKg, Created Date, Modified Date, Slug, and Created By. The data is as follows:

| Pet | WeightKg | Created Date | Modified Date | Slug | Created By |
|-------------------------|----------|-----------------------|-----------------------|------|------------------|
| 3554x124158377302949890 | 12.7 | Nov 13, 2022 11:29 pm | Nov 13, 2022 11:29 pm | | test@example.com |
| 3554x124158377302949890 | 12.3 | Nov 13, 2022 11:29 pm | Nov 13, 2022 11:29 pm | | test@example.com |

Let's display your pet's weight graphically

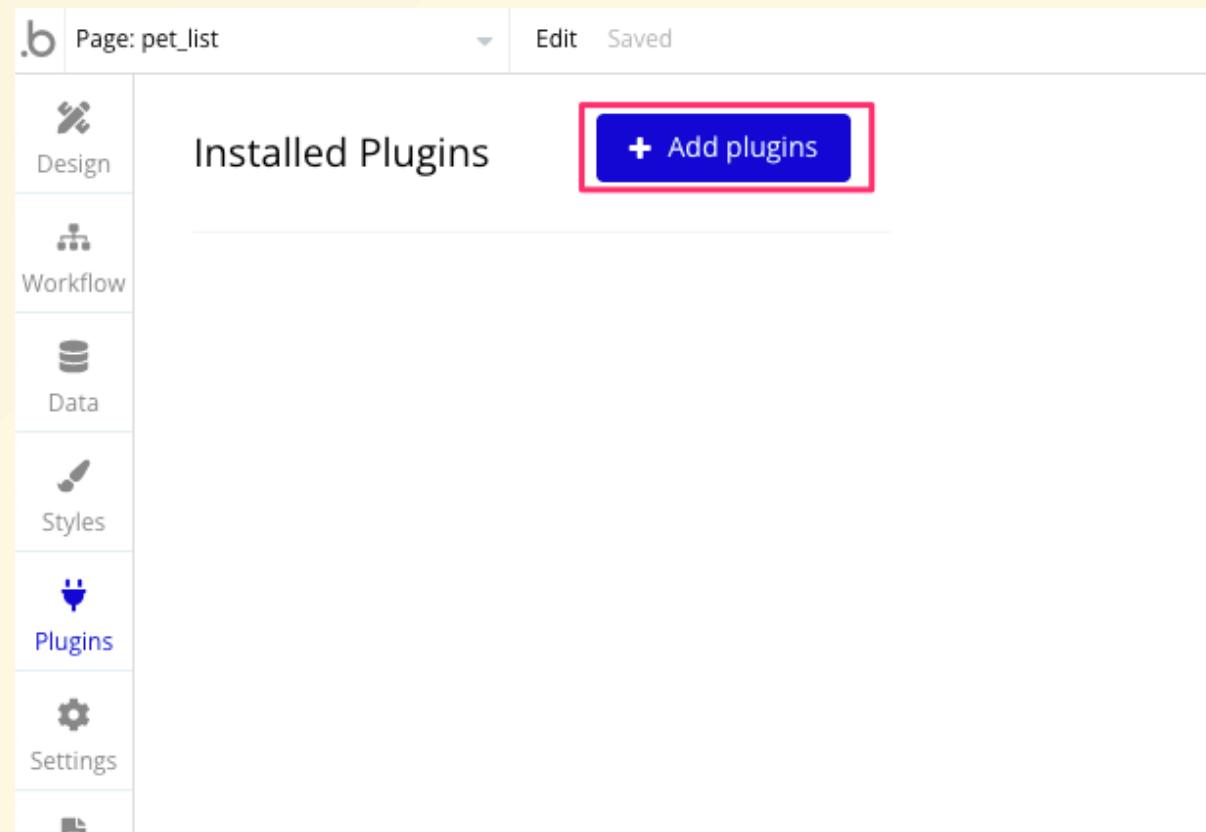
- Let's display the weight of your registered pets using a graph as in the case of Adalo



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

Install New Plugins

Filters

Sort by name usage rating date price

chart

Types (deselect all)

- Action
- Api
- Background Services
- Element
- Event
- Login Service

Chart Element

Add a chart element to your application. You can display line, bar charts or polar charts.

By Bubble 

100,107 apps

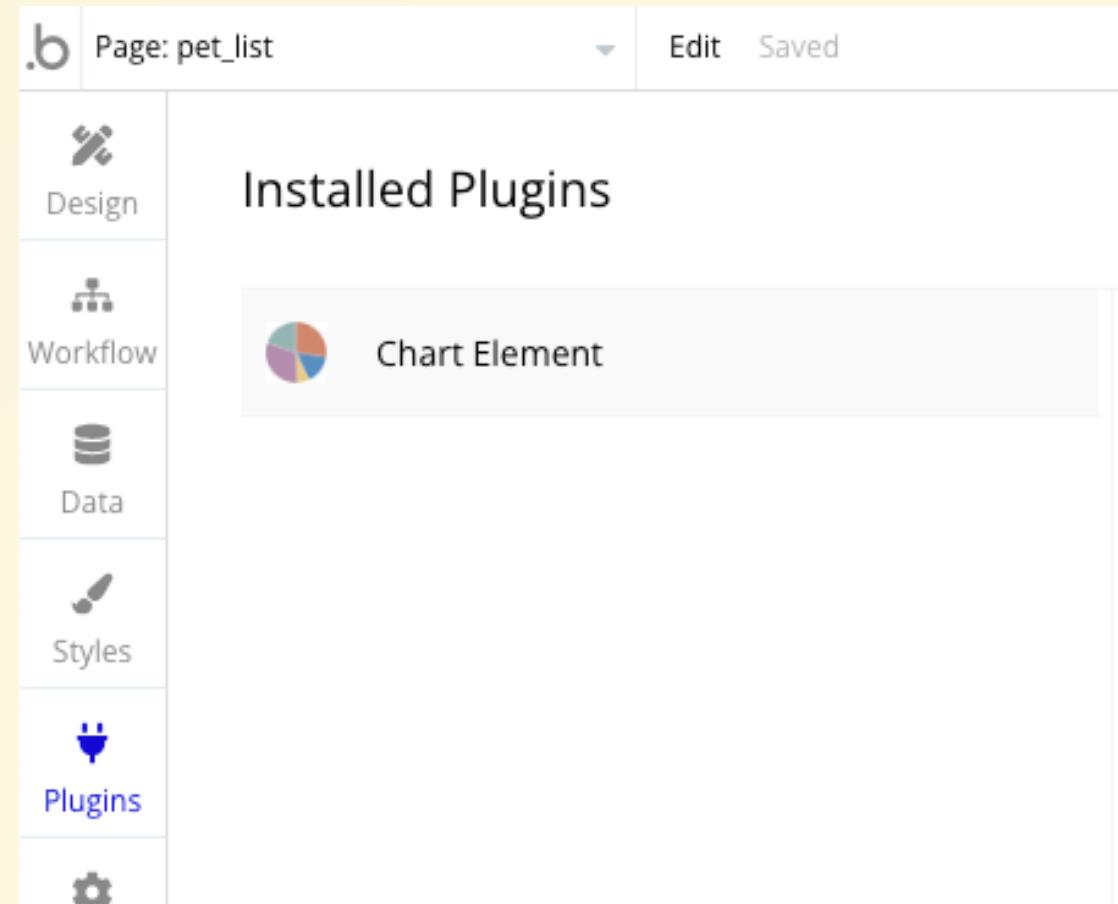
Install

amCharts Free

This plugin enables Bubble users to visualise data through the amCharts library. Visit www.amcharts.com to learn more about the library. Licenses are available for purchase which will remove the 'amC...

By wedgehog 

- 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, the ones 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

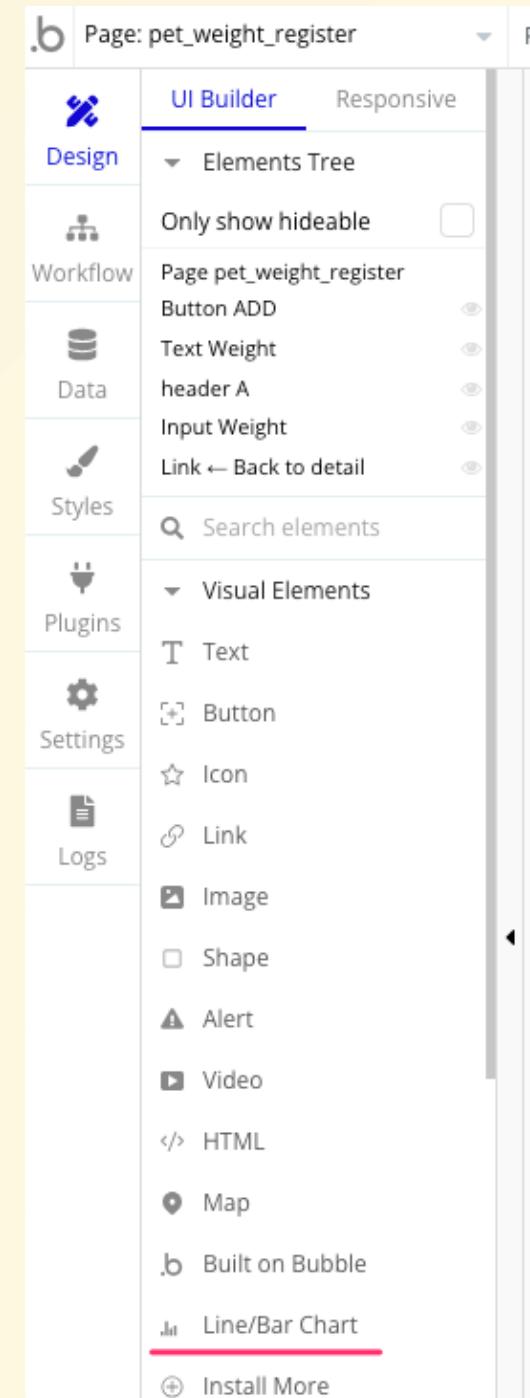
Plugin page

Buy \$190

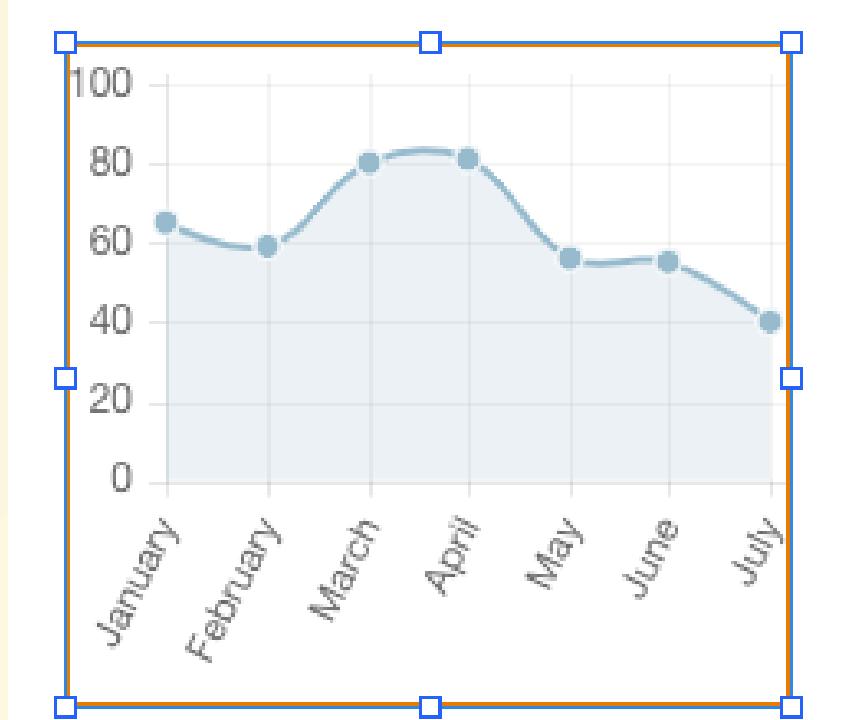
Subscribe

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



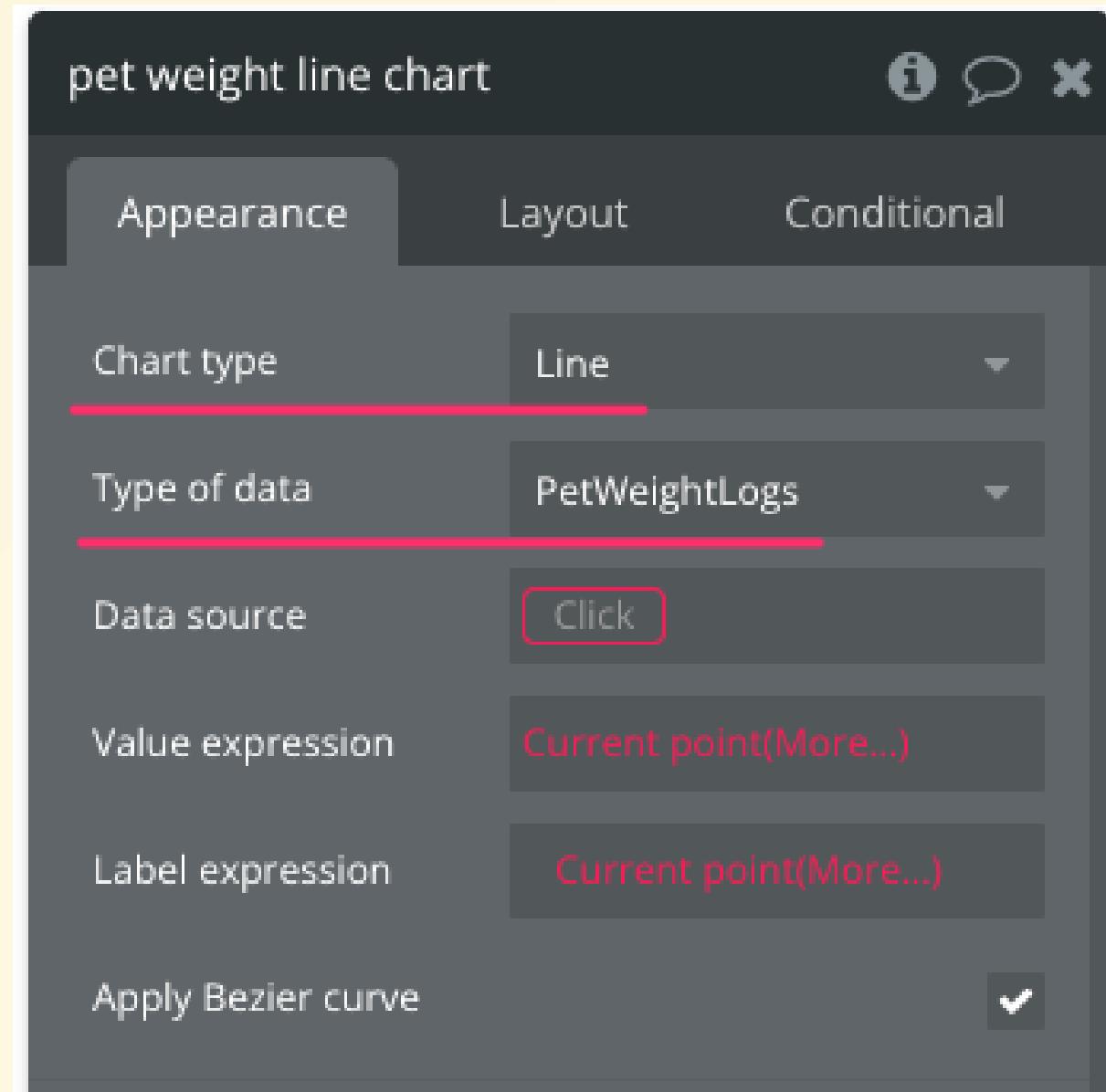
Weight

pet weight

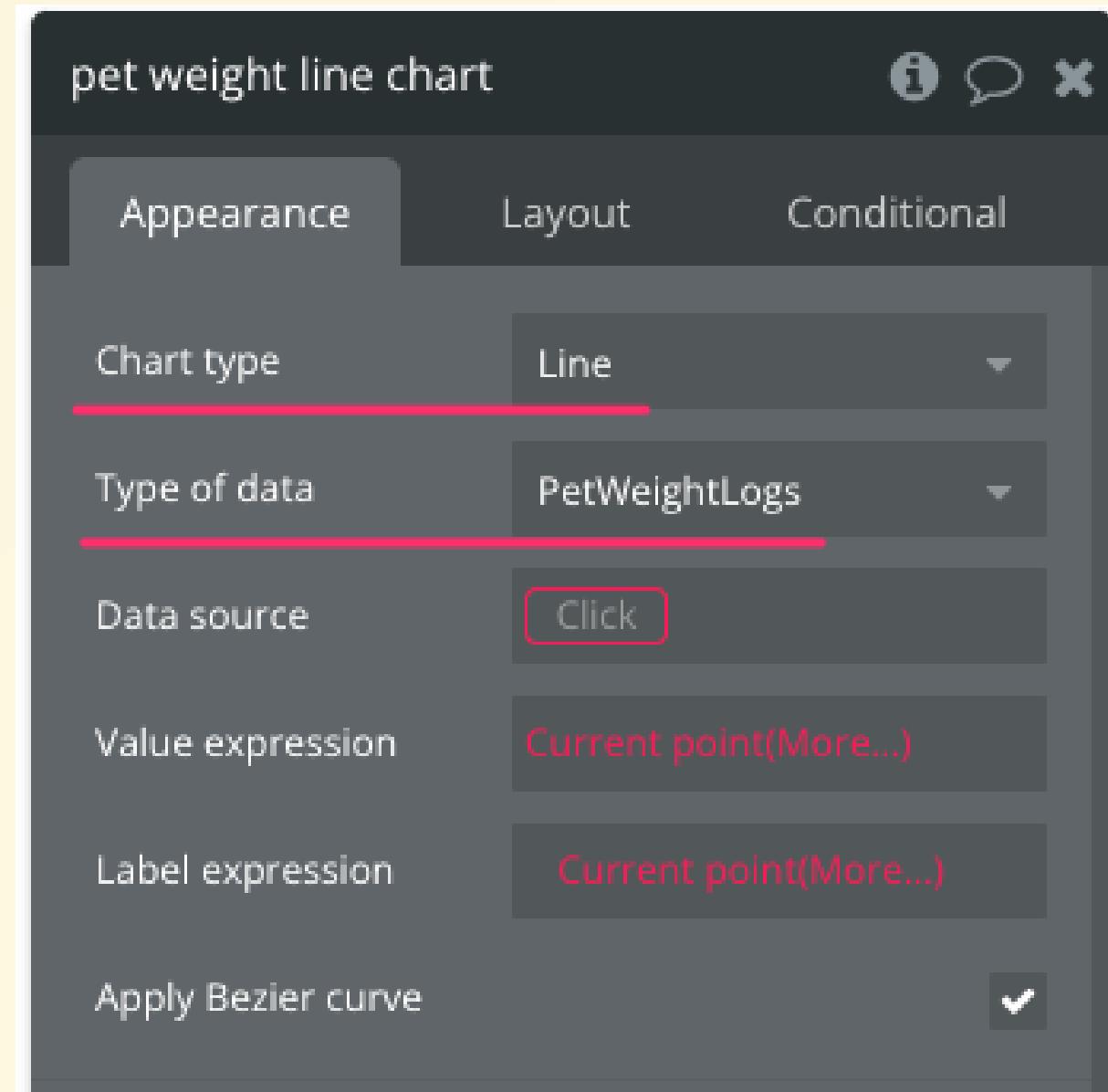
← Back to detail

ADD

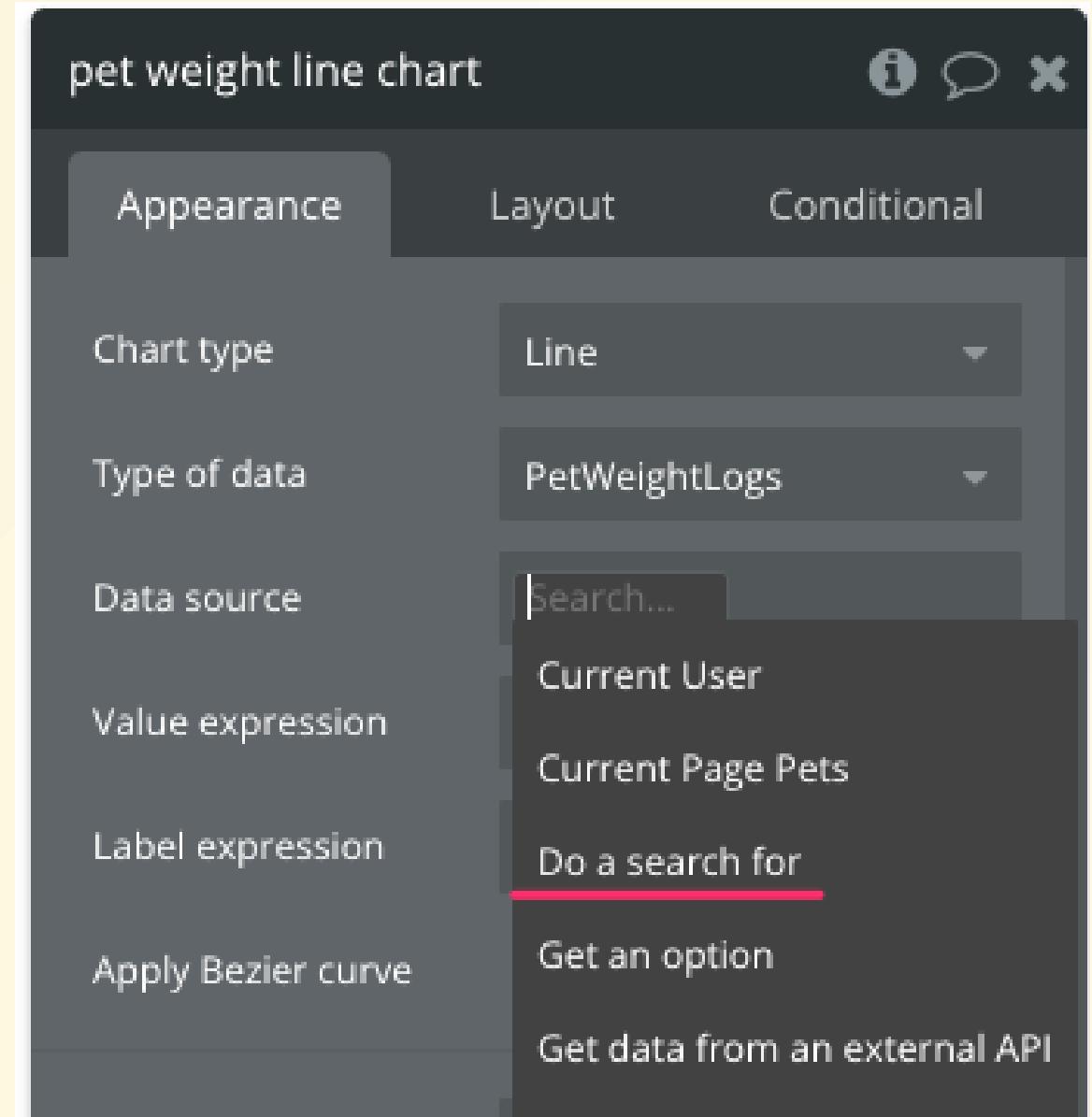
- After drawing, a dialog box for detailed settings will appear as before, so let's set each item!
- **Chart type** specifies the type of chart.
 - In this case, it is set to **Line** since it is a line chart.



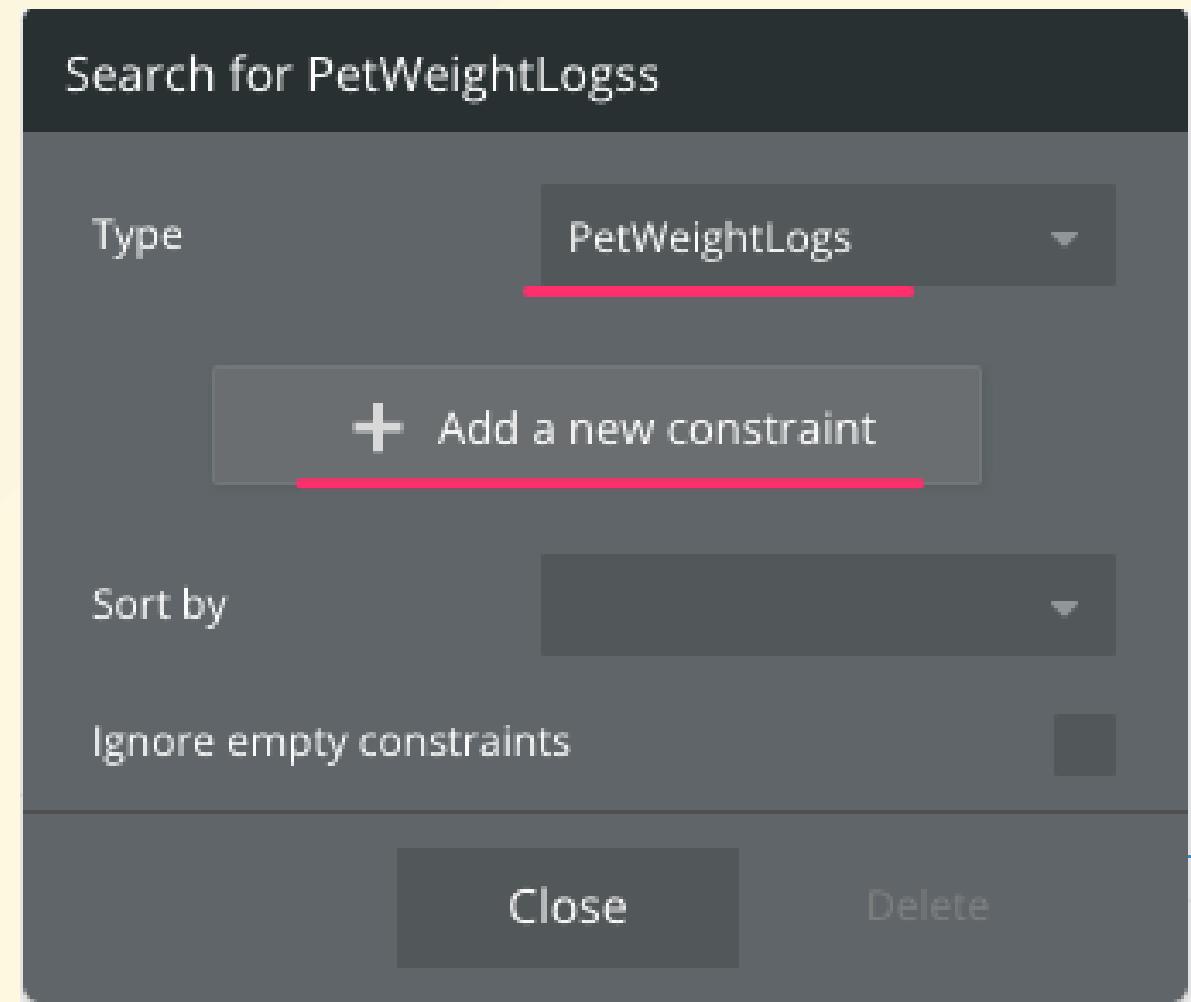
- Type of data specifies the type of data to be plotted in the graph.
 - In this case, PetWeightLogs is used to display the pet's weight.



- 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.



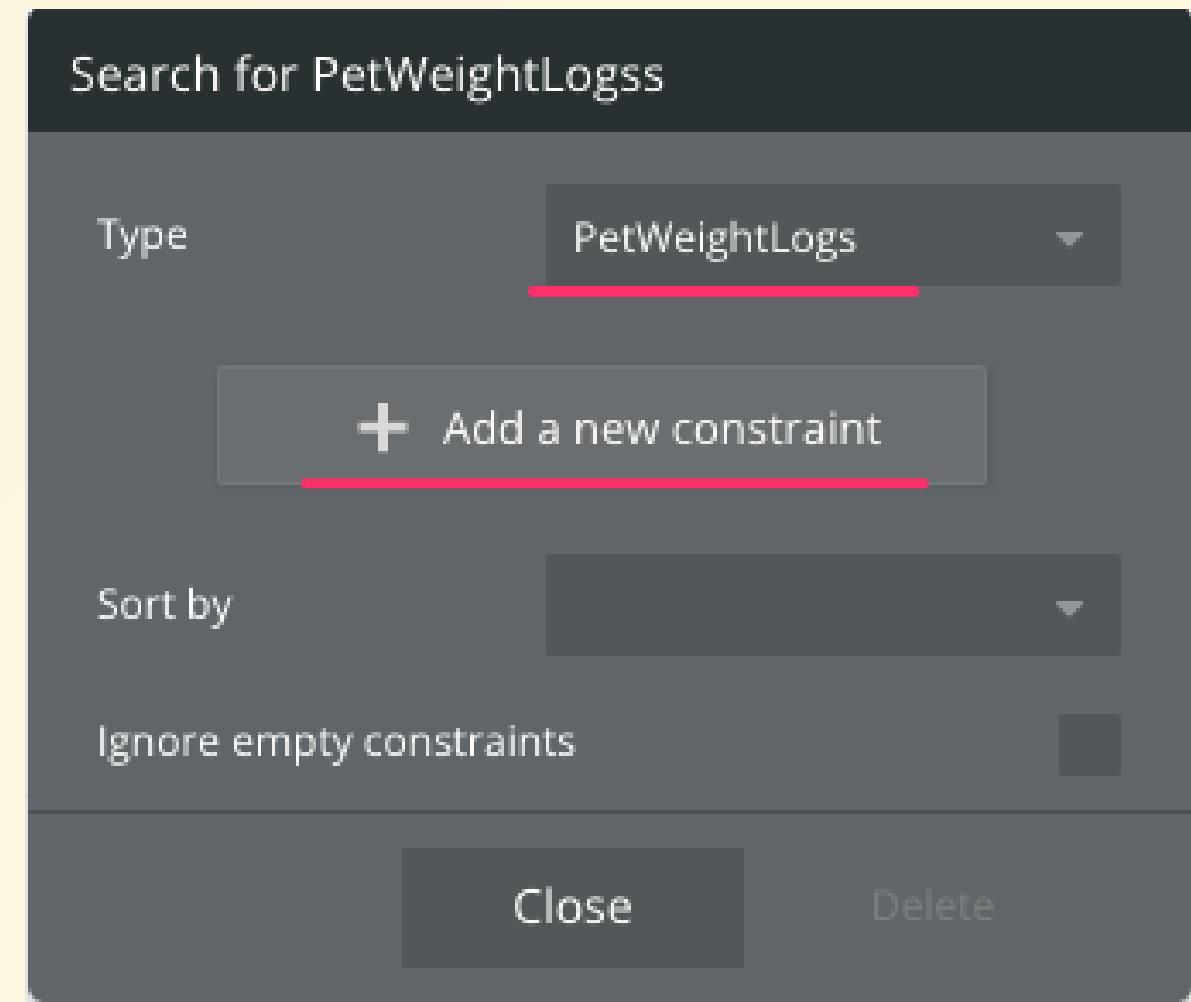
- A **Search for** dialog will appear.
 - Type `PetWeightLogs` is the type of the target data, so specify `PetWeightLogs``.



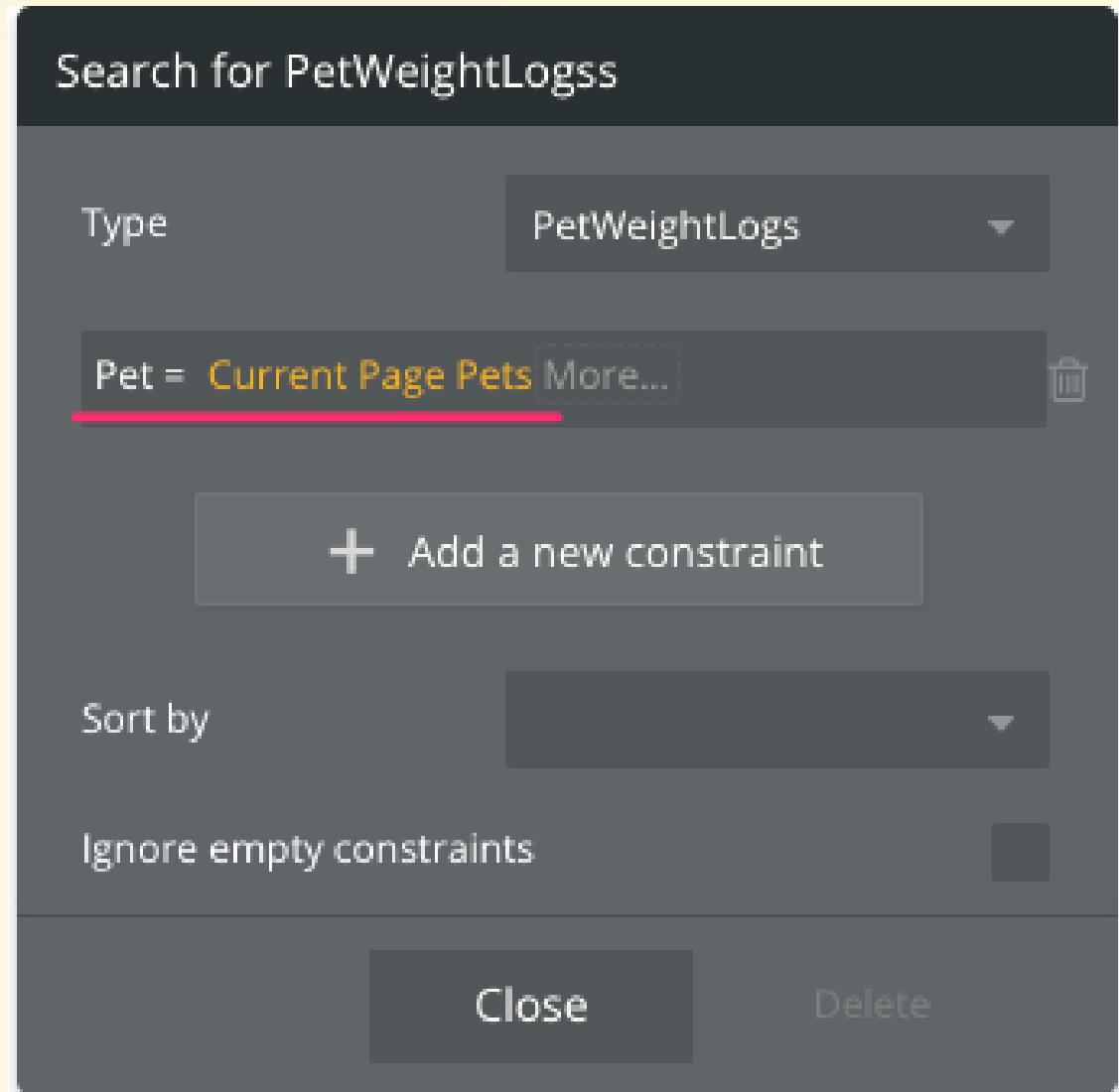
- This alone will display all the weight data for all pets, so we need to specify the criteria to narrow down the search.

- Click on

Add a new constraint to specify the criteria.

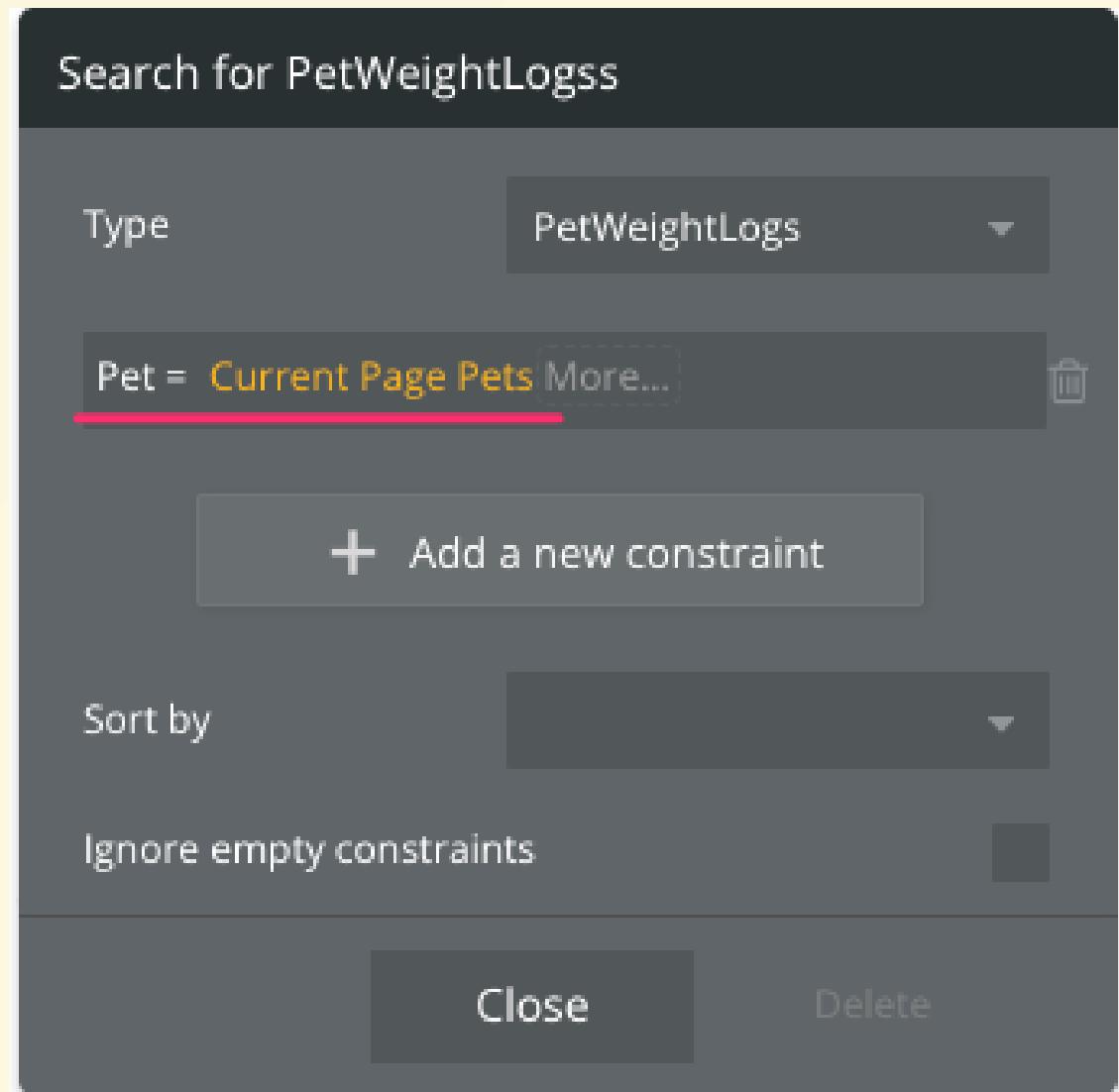


- First, click on "Click" to display the fields that the `PetWeightLogs` specified in Type has
- In this case, we will target the data of the `Pet` related to the `PetWeightLogs`, which is the same as the currently displayed pet.

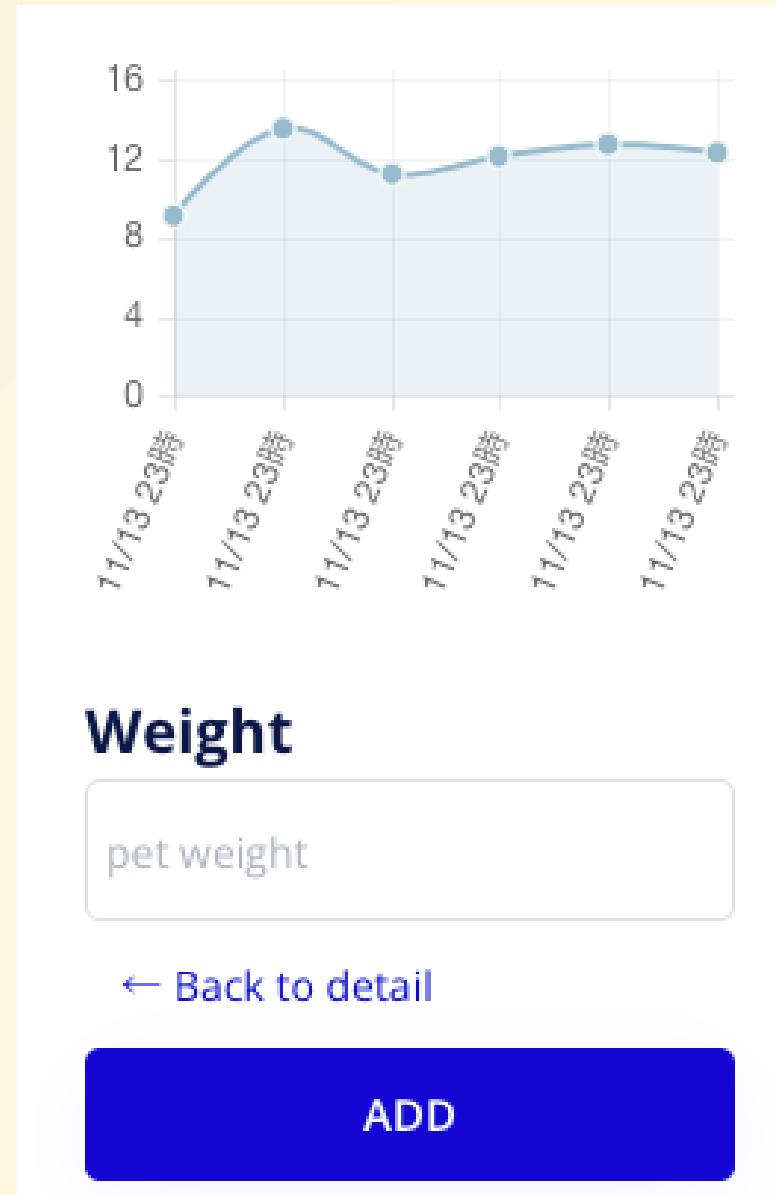


- Let's express this as a condition, which looks like this, so let's set it up

Pet = Current Page Pets



- Finally, specify the order of the data
 - For the presentation of 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).
- When you are done, 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.

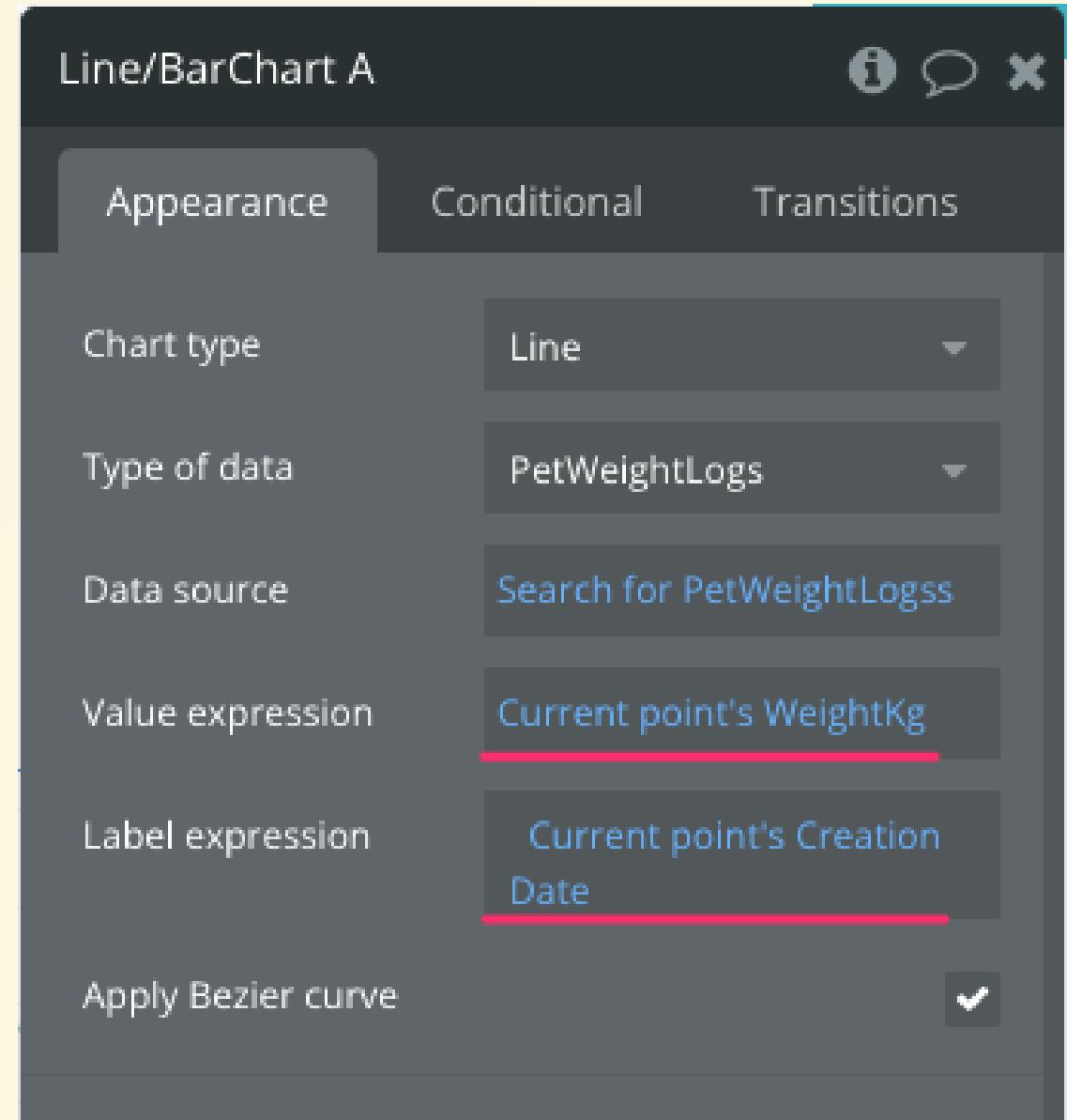
- This time,
Current point's
Creation Date

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 it!
- 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 constraints.

Left Screenshot: Search for PetWeightLogss

- Type:** PetWeightLogs
- Constraint:** pet = Current Page Pets
- Buttons:** Add a new constraint, Close, Delete

Right Screenshot: Group A

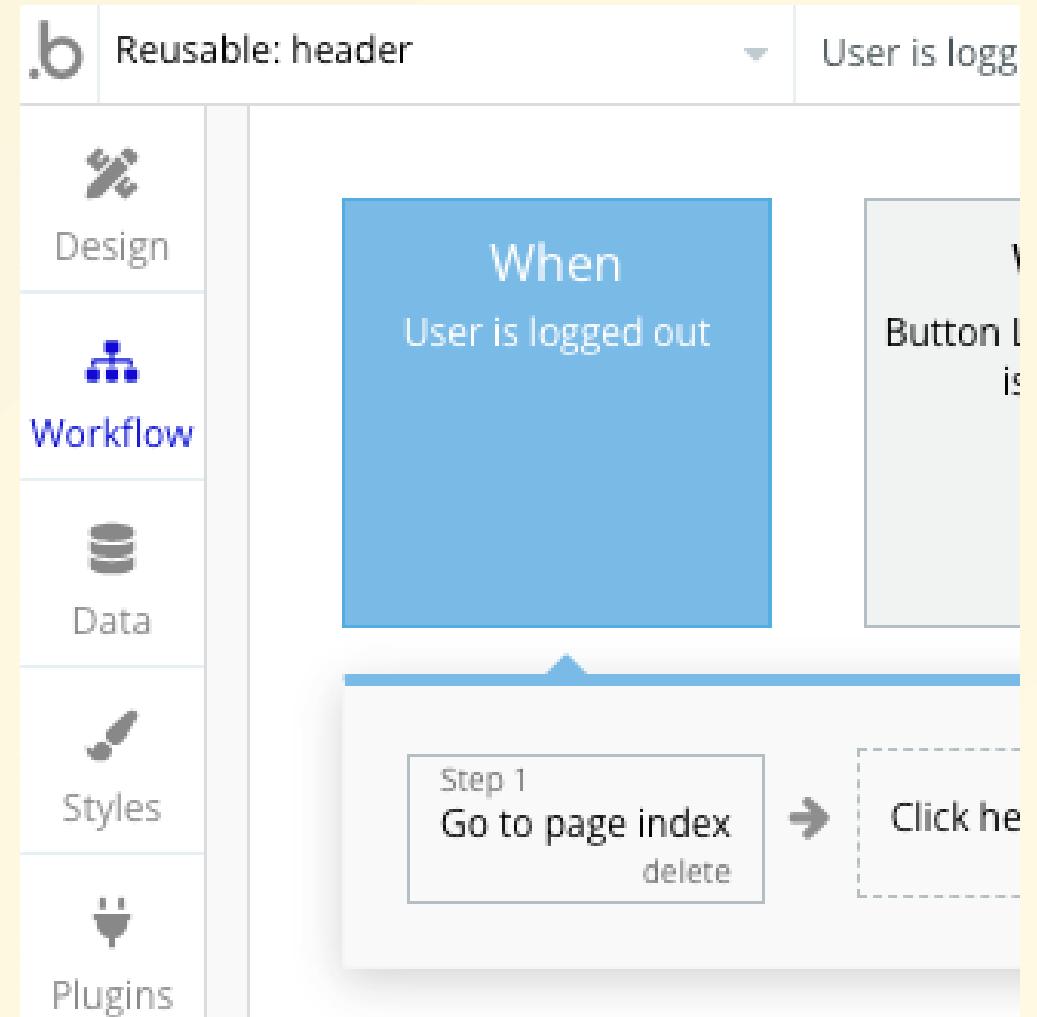
- Appearance:** OFF
- Conditional (1):** When Search for PetWeightLogss:count is 0
- Transitions:** remove condition
- Element Visibility:** This element is visible (checkbox checked)
- Action:** Select a property to change when true
- Buttons:** Define another condition

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
- You will find the answer (the settings) on the next page!

- For header parts
 - When: User is logged out
 - Go to page index
- Unlike other workflows, buttons are not triggered, so the colors have 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 points are as shown in the following sheet!
Please try to 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.) is 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 design of the screen and 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!