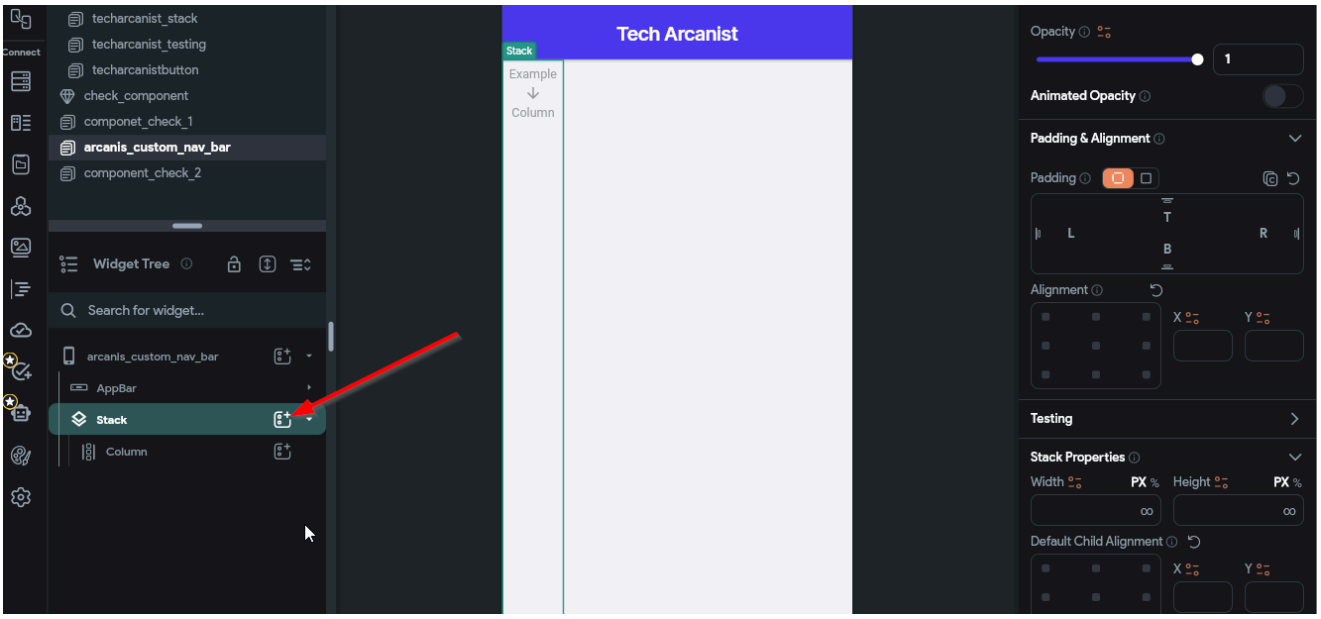
Custom Bottom Navigation Bar using Stack and Components in FlutterFlow

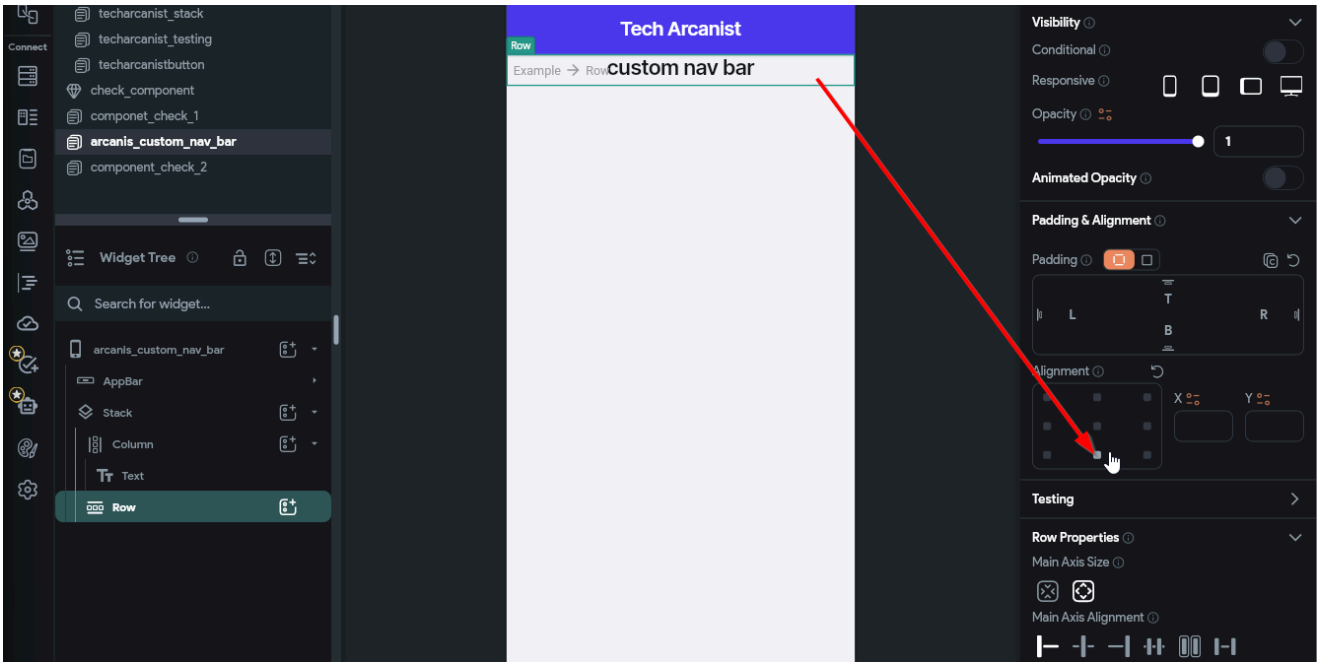
Now we will build a **custom navigation bar** instead of using the default navbar.

Step 1: Wrap Column in Stack



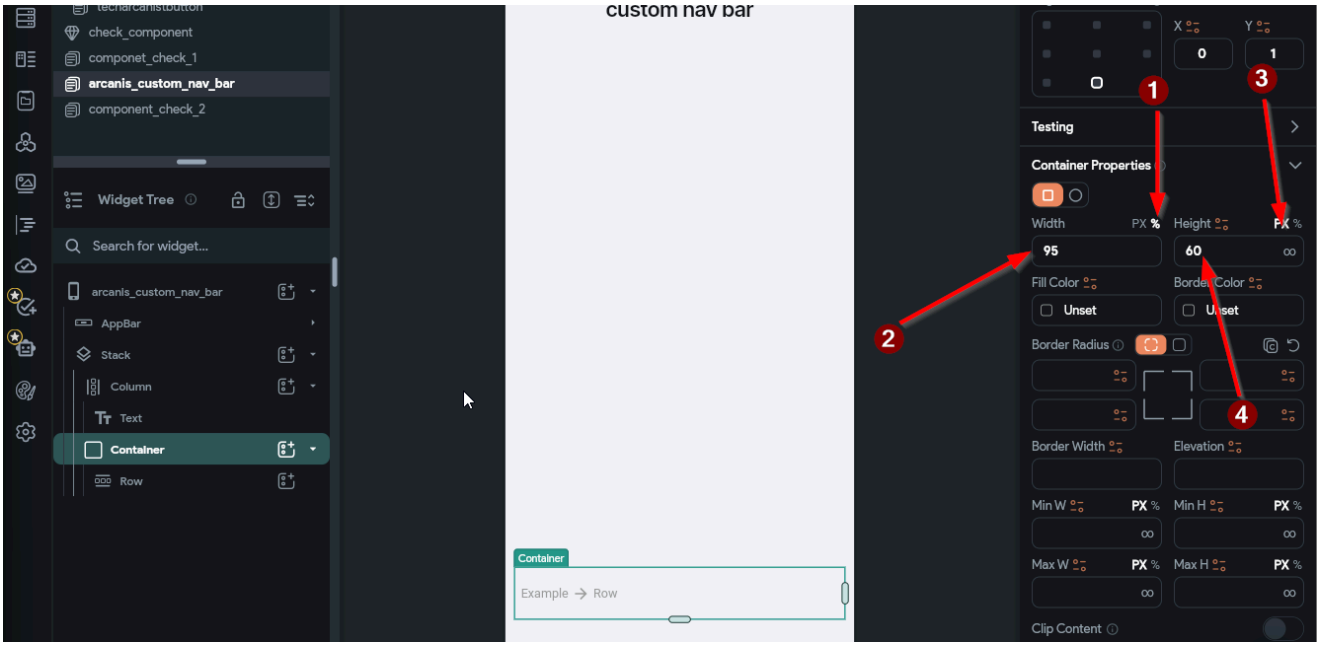
First, wrap the existing **Column** inside a **Stack**.
This allows us to place widgets on top of each other.

Step 2: Add Row inside Stack



Add a **Row** inside the Stack and move it to the bottom using the **Alignment** option.

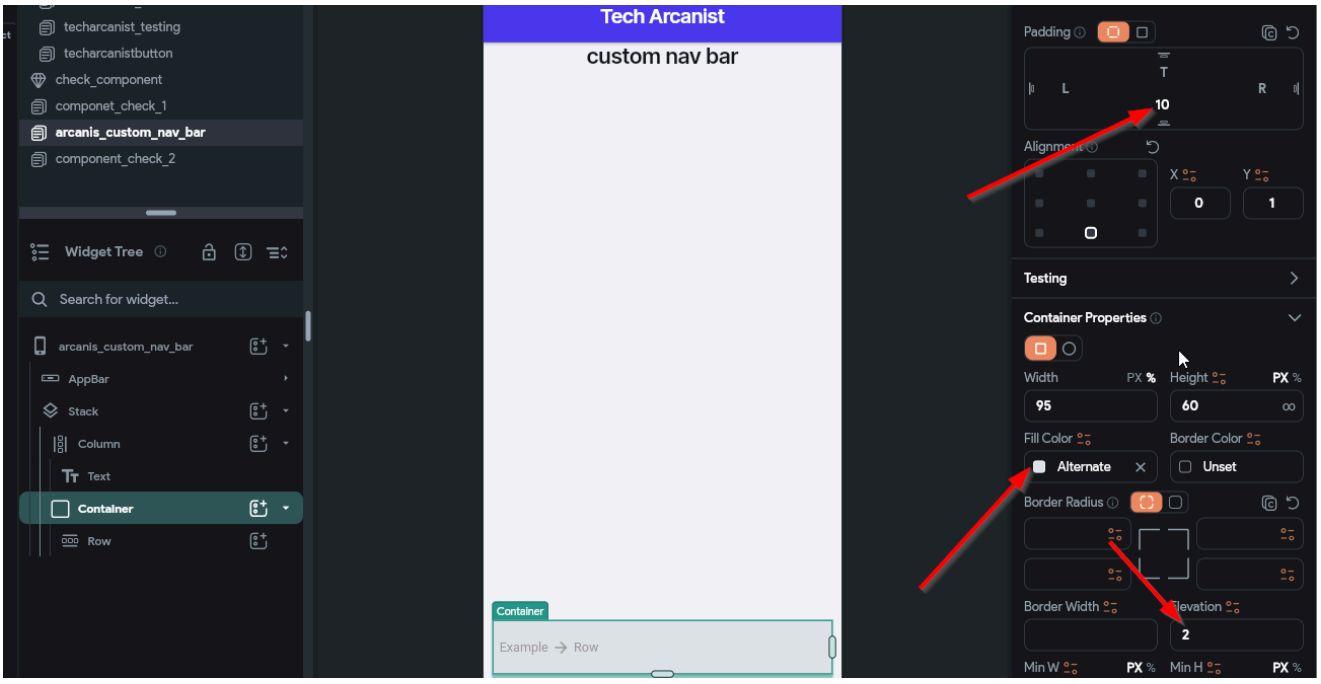
Step 3: Wrap Row in Container



Wrap the Row inside a **Container** and set:

- **Width**: 95% (use percentage option)
- **Height**: 65 px

Step 4: Style the Container



Apply the following:

- **Bottom padding**: 10
- Add a **light background color**
- Set **Elevation = 2**

This gives the navbar a floating effect.

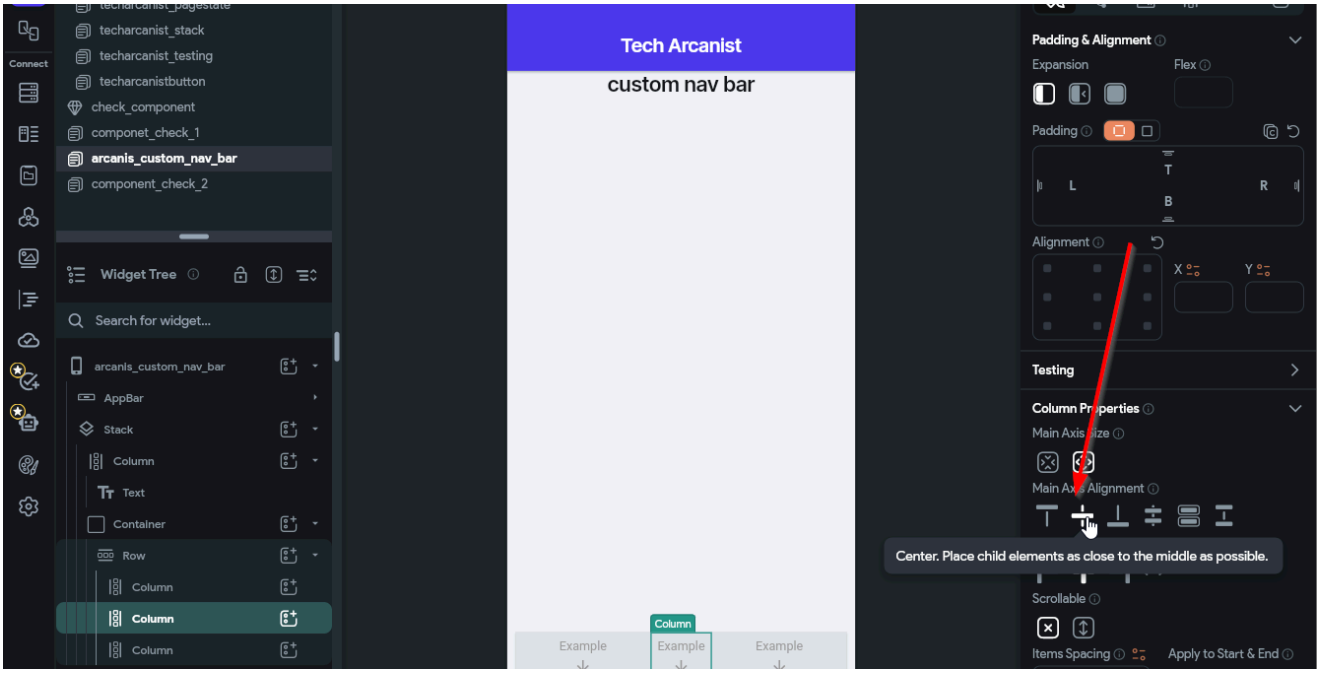
Step 5: Add Columns for Icons



Inside the Row:

- Add **3 Columns**
- Set Row **Main Axis Alignment = Space Evenly**

Step 6: Center Content in Columns

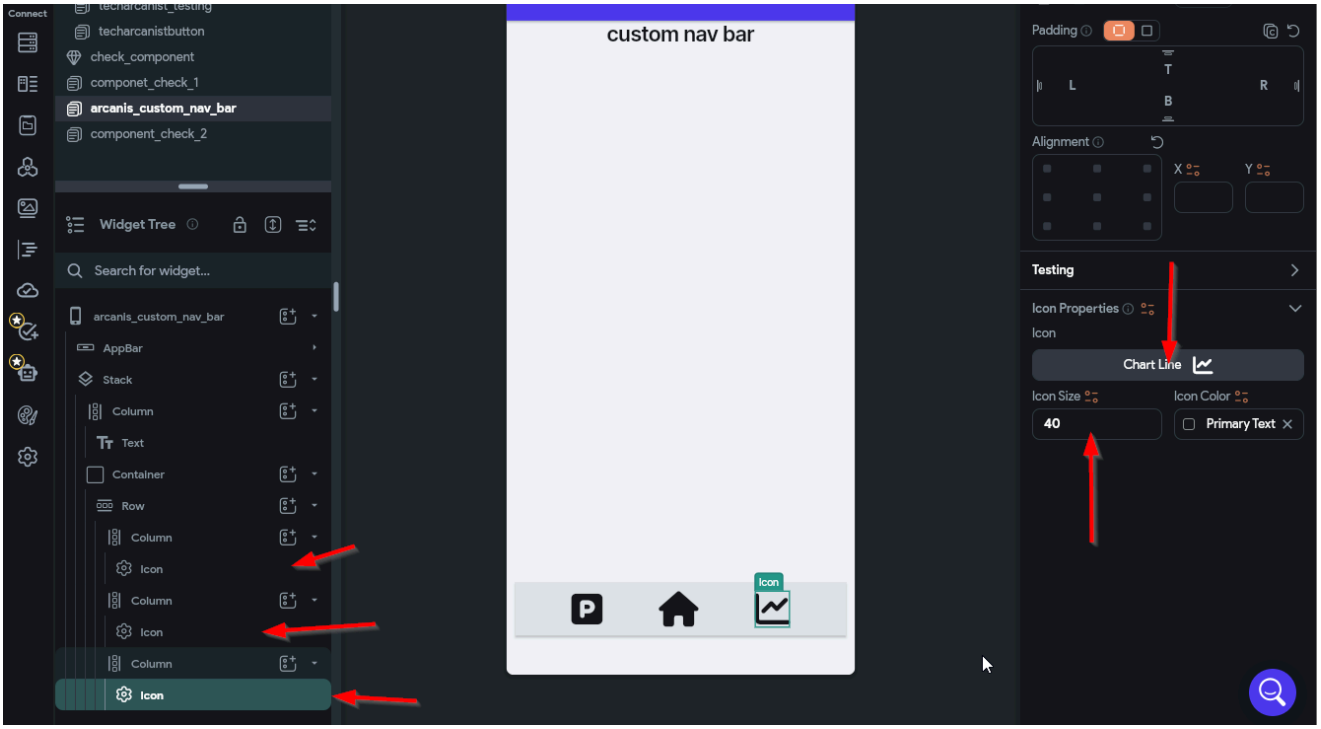


For each column:

- Set **Main Axis Alignment = Center**

This ensures icons and text appear centered.

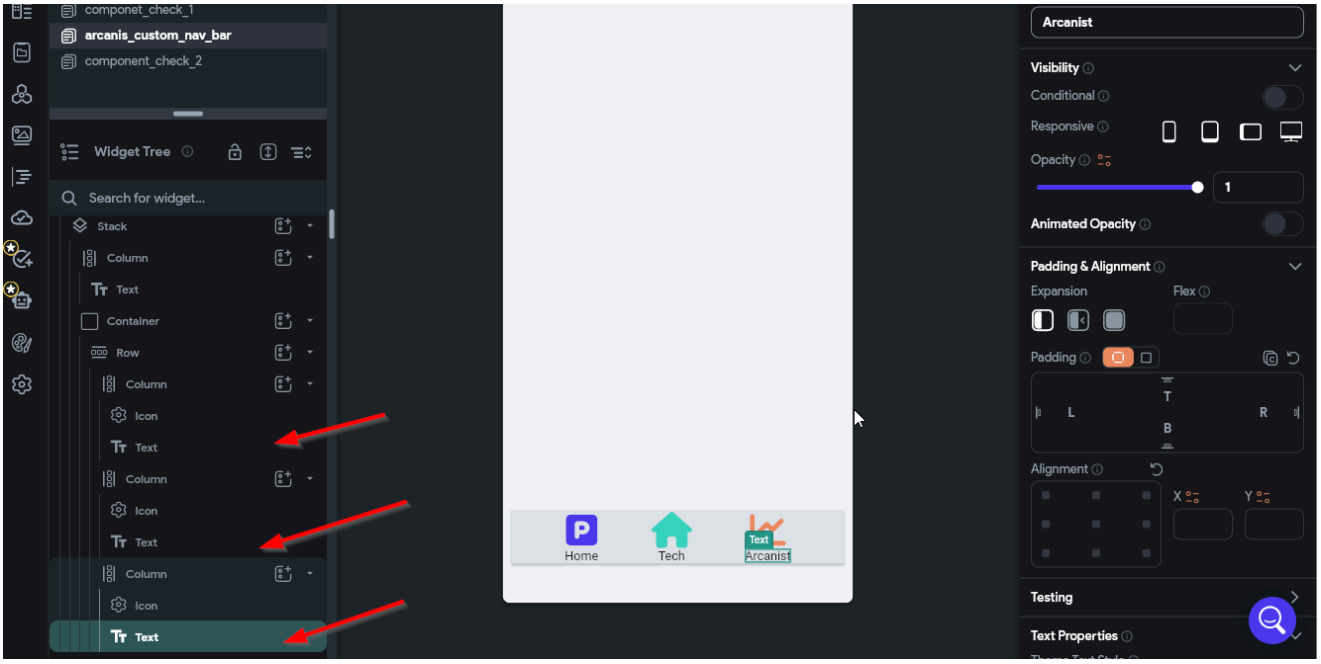
Step 7: Add Icons



Add an **Icon** to each column:

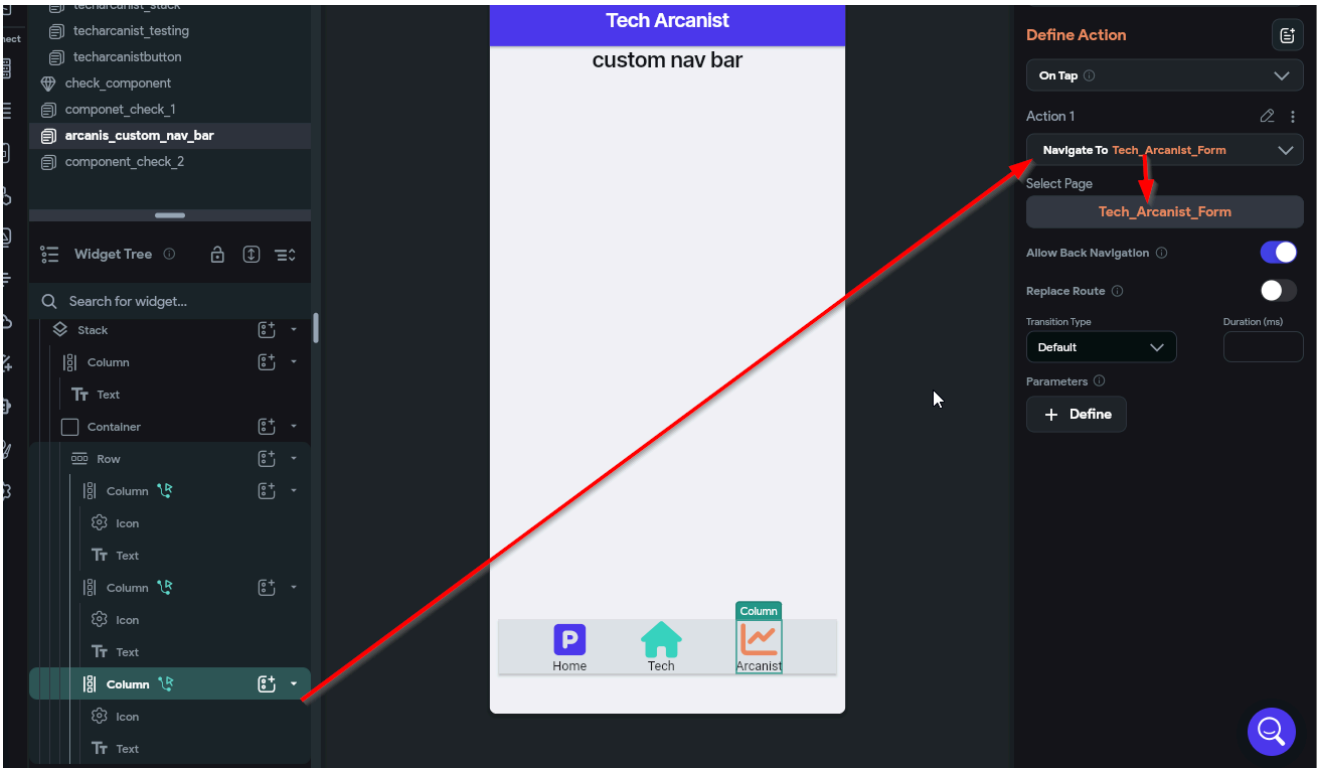
- Set **Icon size = 40**
- Choose required icons
- Change icon colors if needed

Step 8: Add Labels



Add **Text widgets** below each icon to label them
(for example: Home, Tech, Profile).

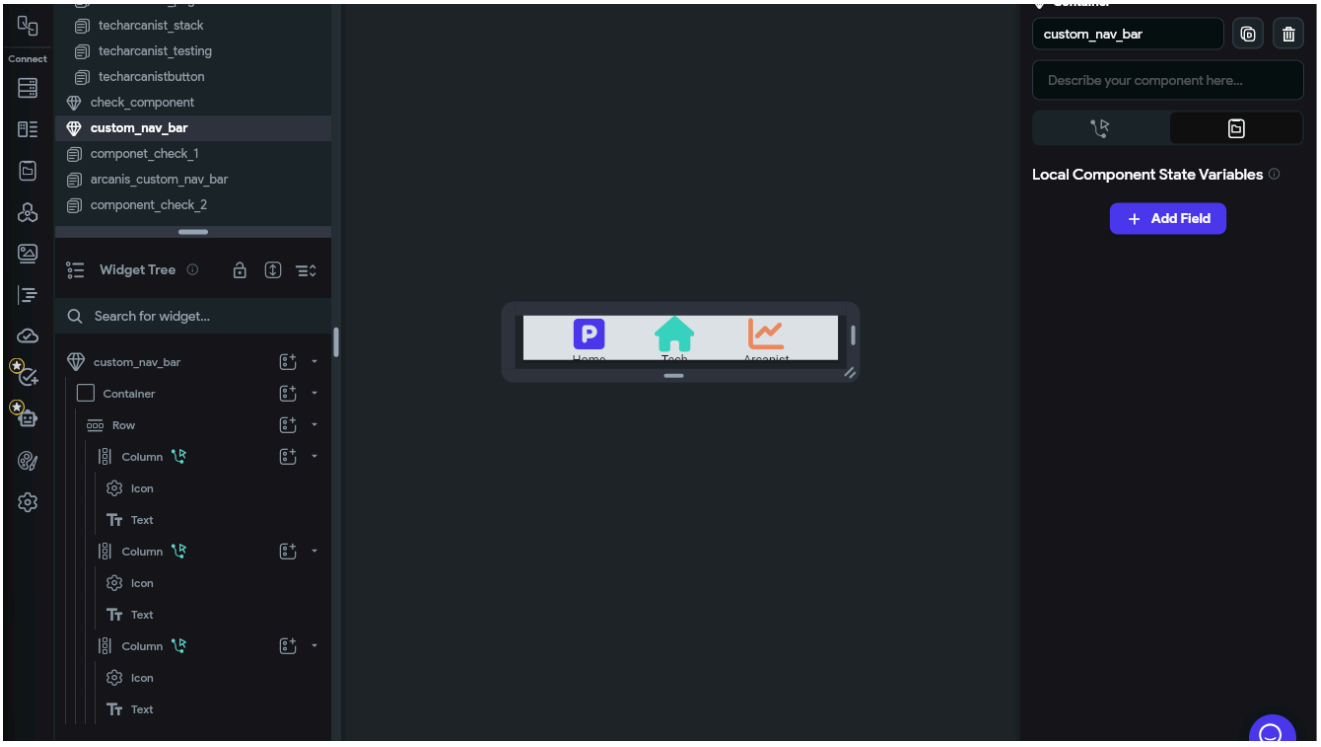
Step 9: Add Navigation Actions



Add **navigation actions** to each column to open different pages.

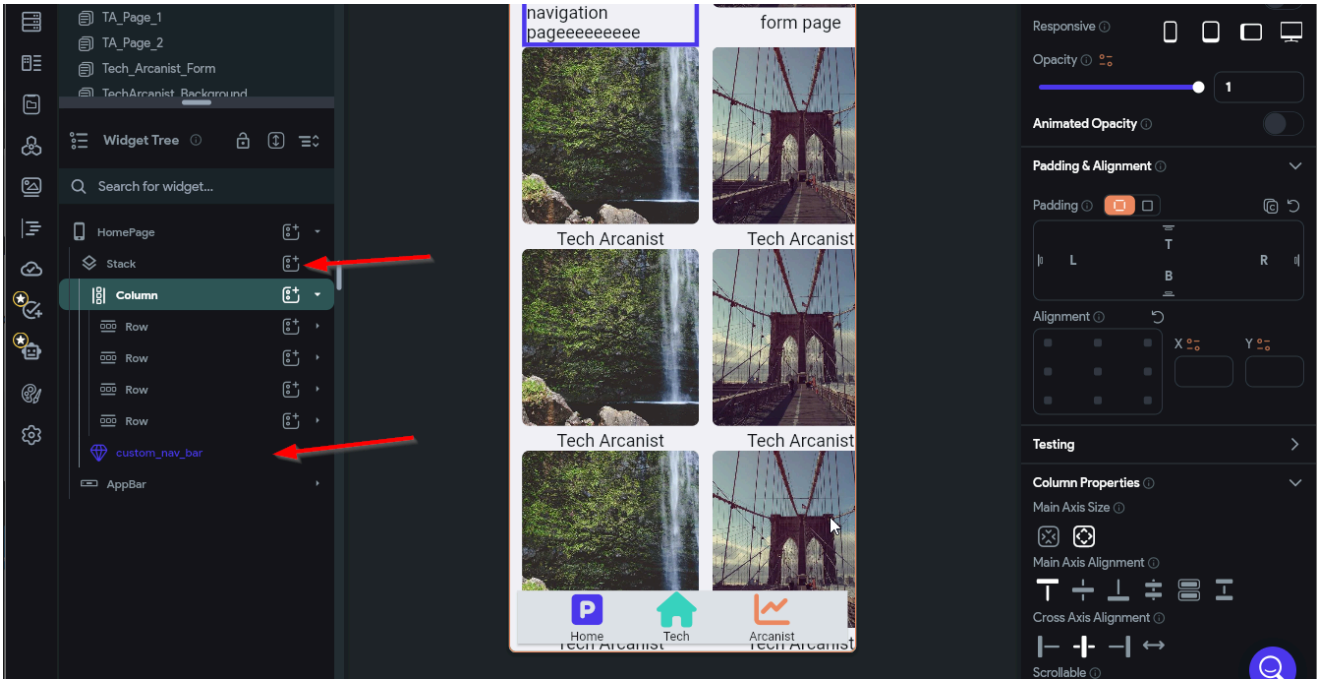
Convert Navbar to Component

Step 10: Convert Container to Component



Right-click the navbar container and select **Convert to Component**.

Step 11: Use Component on Other Pages



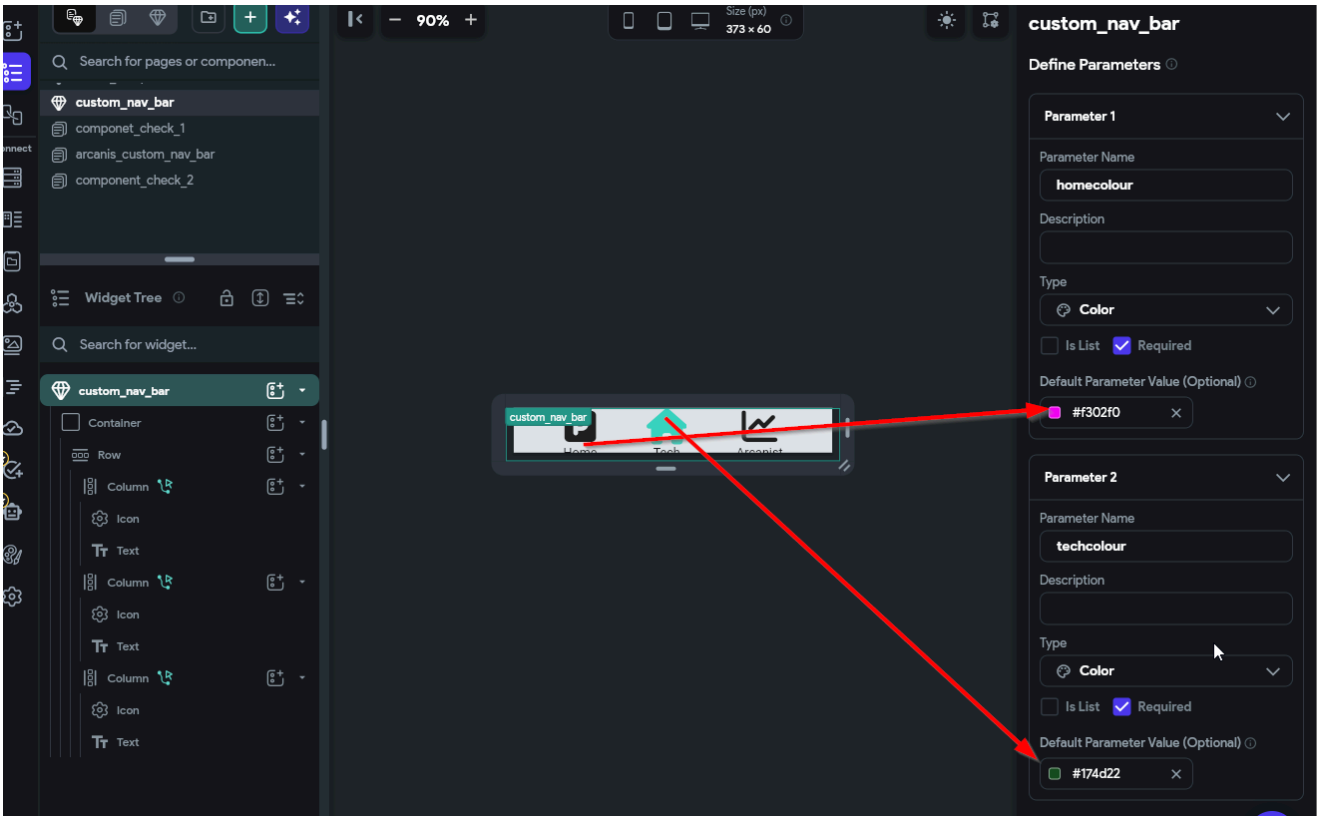
On other pages:

- Wrap the main column inside a **Stack**
- Add this **Navbar Component** inside the Stack

Dynamic Icon Color using Component Parameters

To highlight the active page, we will change icon colors using **Component Parameters**.

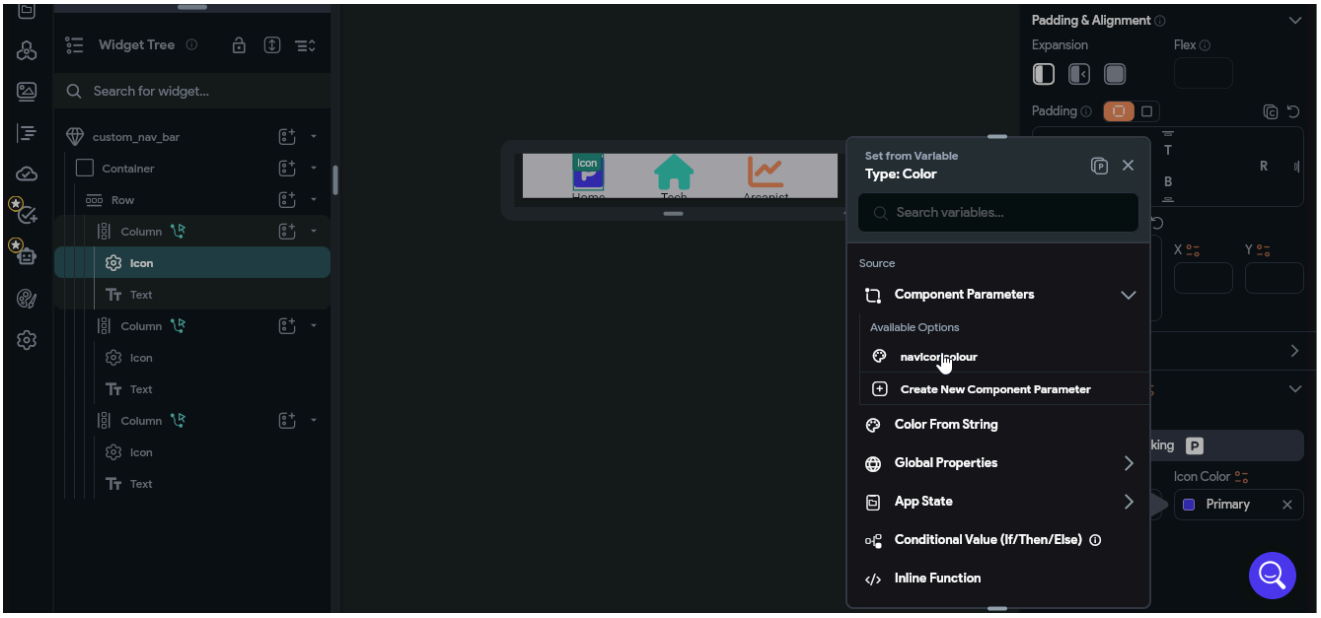
Step 12: Add Component Parameters



Add the following parameters:

- **homeColor** → Type: Color → Default color
- **techColor** → Type: Color → Default color
- **profileColor** → Type: Color → Default color

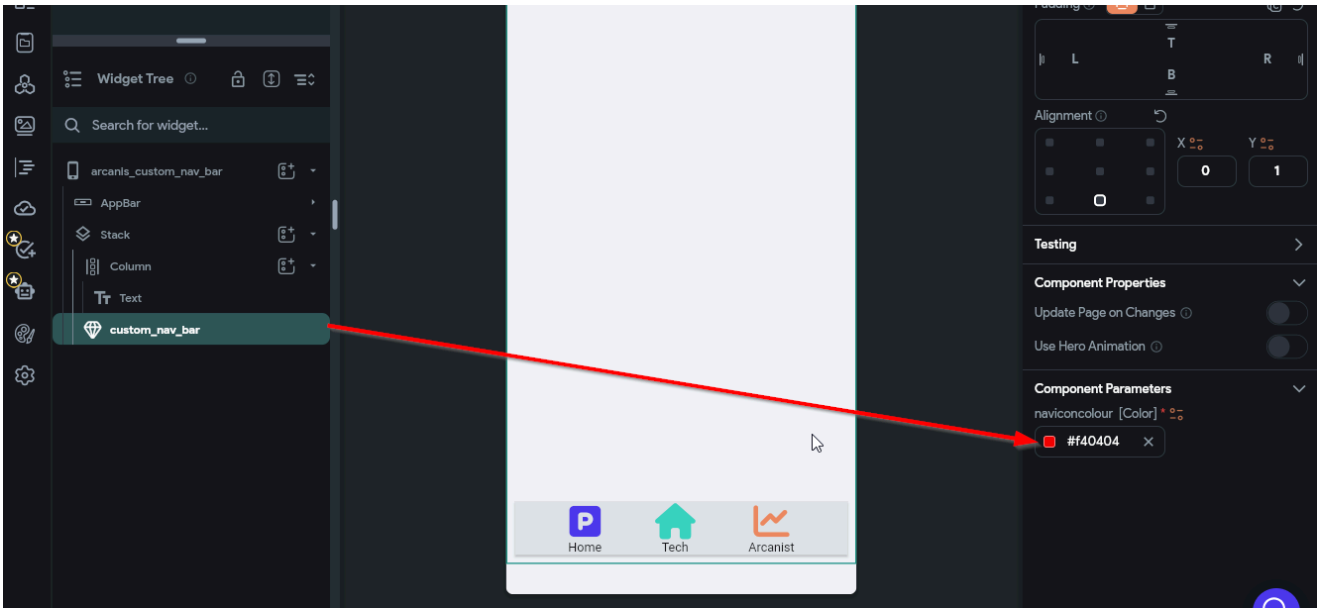
Step 13: Bind Parameters to Icons



Inside the component:

- Bind each icon's **Color** property to its respective parameter (example: Home icon → homeColor)

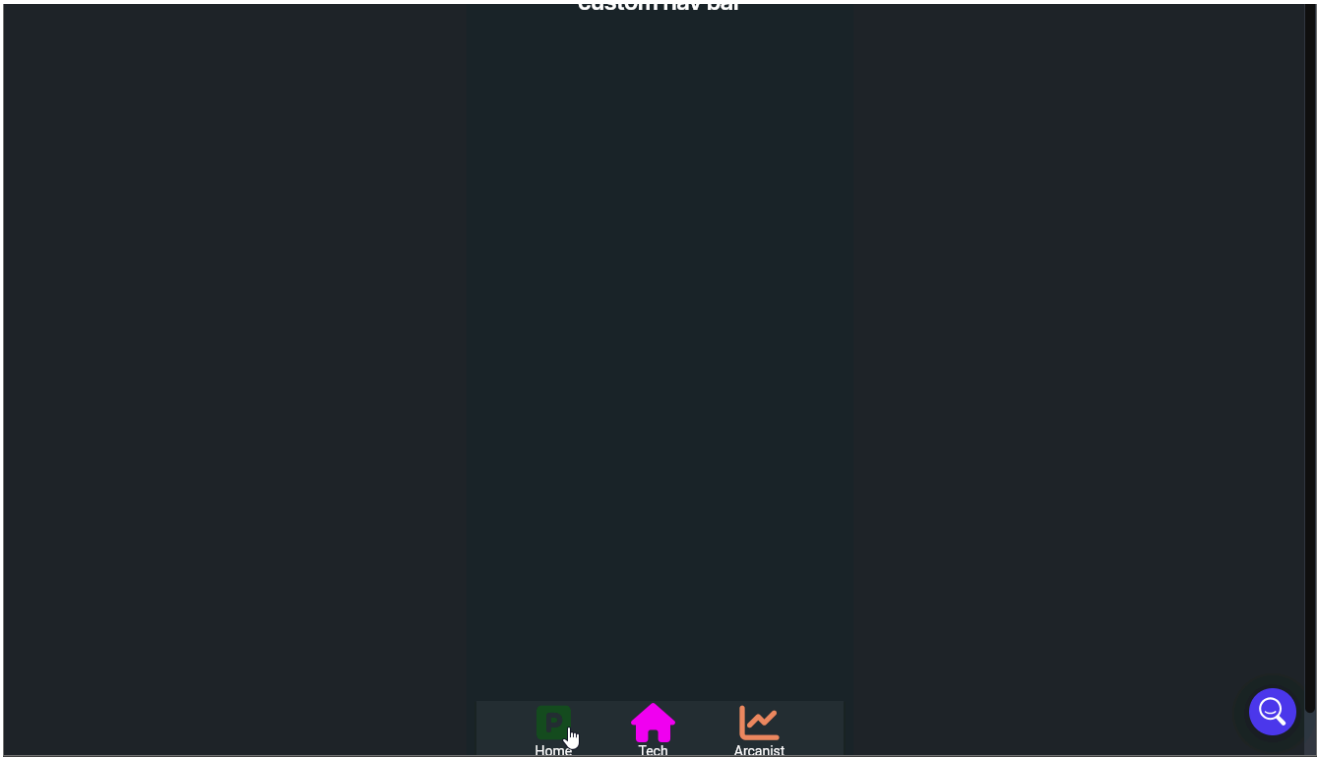
Step 14: Change Colors Per Page



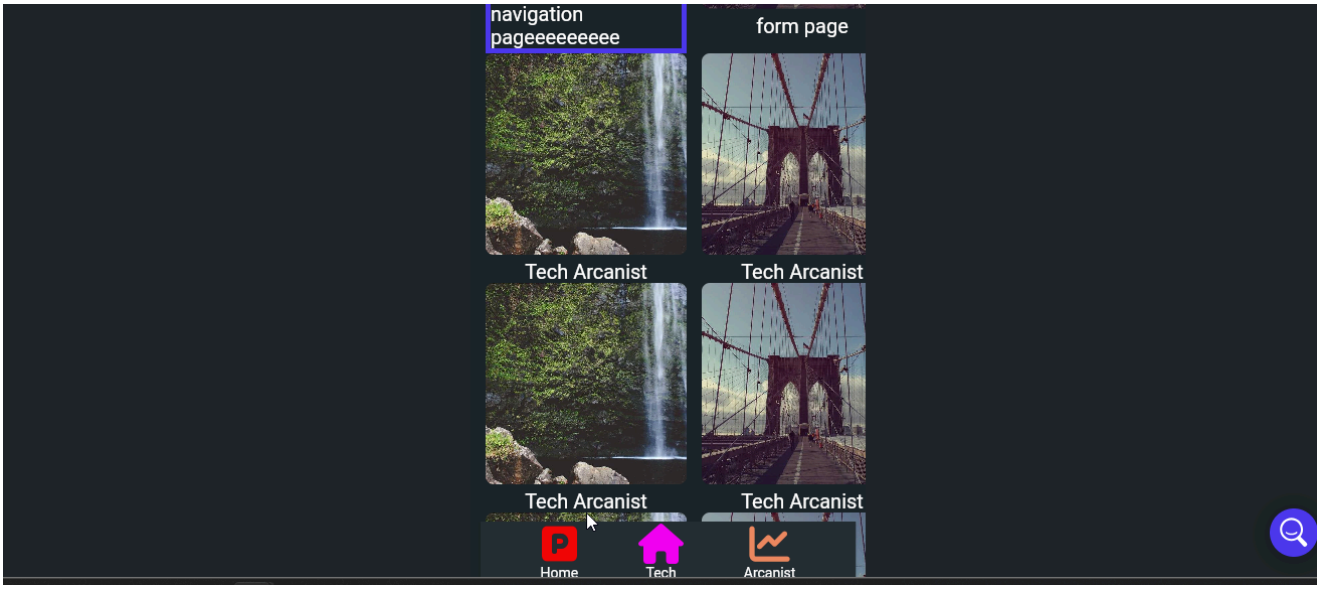
Go to each page, select the navbar component, and change:

- The active page icon color (for example, Red)

Step 15: Result



When navigating to the Home page, the Home icon color changes.



The custom navbar is now working correctly across pages.
